



Частное учреждение высшего образования  
«Высшая школа предпринимательства (институт)»  
(ЧУВО «ВШП»)

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**  
**Б1.О.10 «Современные технологии разработки программного обеспечения»**

Направление подготовки: 09.04.03 Прикладная информатика

Направленность (профиль) программы магистратуры  
«Прикладная информатика и информатизация»

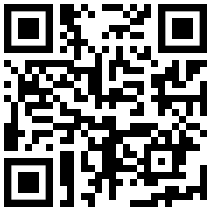
**ОДОБРЕНО**

Ученым советом ЧУВО «ВШП»

Протокол заседания

№01-02/24 от 30 августа 2025 г.



	Документ подписан электронной цифровой подписью VSHR EDS GEN 1, уникальный ключ документа:	
	<b>8F30-29EE-EB2F-GNI5</b>	
	Организация:	ЧУВО «ВШП», ИНН 6903013604
	Дата подписания:	30.08.2025
	Подписал:	Аллабян М.Г.

Рабочая программа учебной дисциплины **Б1.О.10 Современные технологии разработки программного обеспечения**, обязательного компонента основной профессиональной образовательной программы высшего образования - программы магистратуры по направлению подготовки **09.04.03 Прикладная информатика** направленность (профиль) **«Прикладная информатика и информатизация»**, направлена на обеспечение у обучающегося способности осуществлять профессиональную деятельность в соответствующей области и сферах профессиональной деятельности, в том числе на их практическую подготовку с учётом рабочей программы воспитания и календарного плана воспитательной работы Частном учреждении высшего образования **«Высшая школа предпринимательства (институт)»** (далее — **ЧУВО «ВШП»**).

# 1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

В результате освоения программы магистратуры обучающийся должен овладеть следующими результатами обучения по дисциплине:

Код	Результаты освоения ООП (Содержание компетенций)	Индикаторы достижения	Перечень планируемых результатов обучения по дисциплине
УК-4	Способен применять современные коммуникативные технологии, в том числе на иностранном(ых) языке(ах), для академического и профессионального взаимодействия	УК-4.1. Выбирает на государственном и иностранном (-ых) языках коммуникативно приемлемые стили делового общения	<p><b>Знать:</b> особенности корректного коммуникативного поведения; стили делового общения.</p> <p><b>Уметь:</b> использовать различные виды устной и письменной речи в деловом общении; применять коммуникативно приемлемые стили делового общения на государственном и иностранном (-ых) языках</p>
		УК-4.2. Использует информационно-коммуникационные технологии при поиске необходимой информации в процессе решения различных коммуникативных задач на государственном и иностранном (-ых) языках.	<p><b>Знать:</b> функциональные стили, особенности каждого из них, условия и цели их использования в устной и письменной коммуникации</p> <p><b>Уметь:</b> коммуникативно и культурно приемлемо вести деловые разговоры в рамках академического и профессионального взаимодействия на государственном и иностранном (-ых) языках.</p>
		УК-4.3. Демонстрирует способность находить, воспринимать и использовать информацию на иностранном языке, полученную для решения коммуникативных задач в рамках академического и профессионального взаимодействия.	<p><b>Знать:</b> лексические, грамматические нормы современного литературного языка</p> <p><b>Уметь:</b> вести диалог и создавать монолог в соответствии с целями и условиями деловой коммуникации</p>
ОПК-2	Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе	ОПК-2.1 Обладает фундаментальными знаниями по программированию и языкам программирования, организации баз данных, системного программирования и компьютерного моделирования.	<b>Знать:</b> методы и технологии применения объектно-ориентированного подхода для реализации программного кода; модели баз данных; методами

	с использованием современных интеллектуальных технологий, для решения профессиональных задач;		компьютерного моделирования при решении профессиональных задач.
		ОПК-2.2 Демонстрирует умение обосновывать выбор современных интеллектуальных технологий и программной среды при разработке оригинальных программных средств для решения профессиональных задач	<p><b>Знать:</b> современные интеллектуальные технологии для решения профессиональных задач.</p> <p><b>Уметь:</b> обосновывать выбор современных интеллектуальных технологий и программной среды при разработке оригинальных программных средств для решения профессиональных задач.</p>
ОПК-5	Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем;	ОПК-5.1 Демонстрирует навыки разработки программ для создания приложений и баз данных ИС	<p><b>Знать:</b> современный уровень развития алгоритмических и программных решений в области разработки программного обеспечения; структуру и типы баз данных.</p> <p><b>Уметь:</b> строить модели баз данных, проектировать модули информационных систем.</p>
		ОПК-5.2 Демонстрирует навыки внедрения и модернизации программного и аппаратного обеспечения информационных и автоматизированных систем	<p><b>Знать:</b> современное программное и аппаратное обеспечение информационных и автоматизированных систем.</p> <p><b>Уметь:</b> модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем для решения профессиональных задач</p>
ОПК-8	Способен осуществлять эффективное управление разработкой программных средств и проектов.	ОПК-8.1. Демонстрирует знание методов организации сбора и изучения научно-технической информации по теме исследований и разработок при проектировании и реализации информационных систем.	<p><b>Знать:</b> архитектуру информационных систем предприятий и организаций; методологии и технологии реинжиниринга, проектирования и аудита прикладных информационных систем различных классов; инструментальные средства информационных систем и сервисов.</p> <p><b>Уметь:</b> выбирать методологию и технологию</p>

			проектирования информационных систем; обосновывать архитектуру ИС
		ОПК-8.2. Способен решать научные задачи в связи с поставленной целью и в соответствии с выбранной методикой.	<p><b>Знать:</b> методы оценки экономической эффективности и качества, особенности процессного подхода к управлению прикладными ИС; современные ИКТ в процессном управлении</p> <p><b>Уметь:</b> управлять проектами ИС на всех стадиях жизненного цикла, оценивать эффективность и качество проекта; применять современные методы управления проектами и сервисами ИС</p>

## 2. Распределение часов дисциплины по семестрам

### ОФО

Семестр (курс)	2 семестр (1)
Виды деятельности	
лекционные занятия	10
лабораторные занятия	14
практические занятия/ семинарские занятия	-
руководство курсовой работой	-
клинические практические занятия (практическая подготовка)	-
контактная работа на выполнение курсового проекта	-
практическая подготовка	-
консультация перед экзаменом	-
самостоятельная работа	84
промежуточная аттестация	-
общая трудоемкость	108

## 3. Структура, тематический план и содержание учебной дисциплины

	лекционные занятия	лабораторные	самостоятельная	формы текущего контроля
--	--------------------	--------------	-----------------	-------------------------

		занятия	работа	
		О Ф О	О Ф О	
<b>Раздел: Раздел 1. Архитектура современных программных систем</b>	4	6	42	тест по итогам занятия доклад / конференция / реферат устный опрос / собеседование кейс

**Тема раздела: Тема 2. Жизненный цикл программного продукта, модели жизненного цикла, сферы их применения.**

Понятие жизненного цикла программного продукта. Этапы жизненного цикла. Международный стандарт ISO/IEC 12207. Модели жизненного цикла (каскадная, с промежуточным контролем, спиральная). Их преимущества и недостатки, области применения.

**Тема раздела: Тема 1. Обзор архитектуры современных программных систем**

Классы задач, решаемых современным программным обеспечением; характеристики и свойства современного ПО и требования к нему; некоторые основные концепции из области архитектуры программных систем. Акцентируется внимание на разнообразии и значительном усложнении ПО. Определения основных терминов, связанных с видами архитектур программного обеспечения. Формулируются требования к современному программному обеспечению: масштабируемость, стандартизация технологий, безопасность, ориентация на использование Web и др. Необходимость принципиально новых технологий разработки и использования программного обеспечения, к которым и относятся облачные вычисления.

<b>Раздел: Раздел 2. Технологии разработки программного обеспечения</b>	6	8	42	тест по итогам занятия доклад / конференция / реферат устный опрос / собеседование кейс
---	---	---	----	--

**Тема раздела: Тема 3. Язык UML. Диаграммы UML. Применение UML для выполнения этапов анализа и проектирования ПС.**

Концептуальная модель языка UML (основные строительные блоки, правила их сочетания и общие для всего языка механизмы – спецификации, дополнения принятые деления, механизмы расширения). Диаграммы языка UML (вариантов использования, классов, состояний, коопераций, последовательности, компонентов, размещения). Их назначение, структура, правила построения. CASE-средства построения UML-диаграмм. Унифицированный процесс разработки программного обеспечения, его базовые принципы. Последовательность действий при анализе: составление и документирование функциональных и нефункциональных требований. Проектирование программного обеспечения, переход от диаграмм анализа к диаграммам проектирования

**Тема раздела: Тема 4. Создание объектно-ориентированного программного обеспечения. Паттерны проектирования.**

Выполнение этапов реализации и проверки программного обеспечения. Паттерны проектирования. MVC-методологии для разработки динамических информационных систем. Понятия модели, вида и контроллера. Активная и пассивная модель MVC. Объектно-ориентированное программирования и активная модель MVC.				
<b>Итого часов</b>	<b>10</b>	<b>14</b>	<b>84</b>	

#### 4. Формы текущего контроля

- тест по итогам занятия (шкала: значение от 0 до 10, количество: 1)

##### Примерное задание:

1. Буква «U» в аббревиатуре «UML» означает:

- А) United
- Б) Unified
- В) Universal

2. Модель UML состоит из (укажите лишнее):

- А) сущностей
- Б) отношений
- В) множеств

3. Сущности UML подразделяются на (укажите лишнее)

- А) структурные
- Б) поведенческие
- В) графические
- Г) группирующие
- Д) аннотационные

4. Отношения UML подразделяются на (укажите лишнее)

- А) зависимости
- Б) ассоциации
- В) уточнения
- Г) обобщения
- Д) реализации

5. Структурные сущности UML включают в себя (укажите лишнее)

- А) классы
- Б) узлы
- В) пакеты
- Г) варианты использования
- Д) интерфейсы

6. Поведенческие сущности UML включают в себя (укажите лишнее)

- А) состояния
- Б) деятельности
- В) варианты использования
- Г) интерфейсы

7. Сущностями UML являются (укажите лишнее)

- А) классы
- Б) узлы
- В) зависимости
- Г) примечания
- Д) варианты использования

8. Группирующие сущности UML включают в себя

- А) классы
- Б) узлы
- В) пакеты
- г) примечания

10. Отношения зависимости в UML являются

- А) симметричными
- Б) антисимметричными
- В) транзитивными

11. Отношения обобщения в UML являются

- А) симметричными
- Б) антисимметричными
- В) транзитивными
- ?

12. Отношения ассоциации (без дополнений) в UML являются

- А) симметричными
- Б) антисимметричными
- В) транзитивными ?

13. Отношения реализации в UML являются

- А) симметричными
- Б) антисимметричными
- В) транзитивными
- ?

13. Отношения реализации в UML являются

- А) симметричными
- Б) антисимметричными
- В) транзитивными ?

14. Множество канонических диаграмм UML

- А) определяется стандартом языка
- Б) является соглашением пользователей языка
- В) определяется производителями инструментов, поддерживающих UML
- ?

15. Множество канонических структурных диаграмм UML включает в себя (укажите лишнее)

- А) Диаграммы классов
- Б) Диаграммы использования
- В) Диаграммы компонентов
- Г) Диаграммы объектов ?

16. Множество канонических структурных диаграмм UML включает в себя

- А) Диаграммы последовательности
- Б) Диаграммы (кооперации) коммуникации



- В) Диаграммы использования
- Г) Диаграммы размещения ?

17. Множество канонических поведенческих диаграмм UML включает в себя (укажите лишнее)

- А) Диаграммы состояний
- Б) Диаграммы деятельности
- В) Диаграммы последовательности
- Г) Диаграммы потоков данных ?

18. Множество канонических поведенческих диаграмм UML включает в себя

- А) Диаграммы классов
- Б) Диаграммы компонентов
- В) Диаграммы последовательности
- Г) Диаграммы размещения (развертывания)

19. Множество канонических диаграмм UML включает в себя (укажите лишнее)

- А) Диаграммы классов
- Б) Диаграммы состояний
- В) Диаграммы последовательности
- Г) Диаграммы потоков данных ?

20. Канонические диаграммы использования предназначены для описания

- А) поведения
- Б) использования
- В) структуры ?

21. Канонические диаграммы реализации предназначены для описания

- А) поведения
- Б) использования
- В) структуры ?

22. Канонические диаграммы классов предназначены для описания

- А) поведения
- Б) использования
- В) структуры ?

23. Канонические диаграммы взаимодействия предназначены для описания

- А) поведения
- Б) использования
- В) структуры
- ?

24. Канонические диаграммы объектов предназначены для описания

- А) поведения
- Б) использования
- В) структуры ?

25. Канонические диаграммы состояний предназначены для описания

- А) поведения
- Б) использования
- В) структуры ?

26. Канонические диаграммы последовательности предназначены для описания

- А) поведения
- Б) использования
- В) структуры ?

27. Канонические диаграммы кооперации предназначены для описания

- А) поведения
- Б) использования
- В) структуры
- ?

28. Канонические диаграммы размещения предназначены для описания

- А) поведения
- Б) использования
- В) структуры

29. Канонические диаграммы деятельности предназначены для описания

- А) поведения
- Б) использования
- В) структуры ?

30. Канонические диаграммы компонентов предназначены для описания

- А) поведения
- Б) использования
- В) структуры

1. Класс - это:

- любой тип данных, определяемый пользователем
- \* тип данных, определяемый пользователем и сочетающий в себе данные и функции их обработки
- структура, для которой в программе имеются функции работы с нею

2. Членами класса могут быть

- \* как переменные, так и функции, могут быть объявлены как `private` и как `public`
- только переменные, объявленные как `private`
- только функции, объявленные как `private`
- только переменные и функции, объявленные как `private`
- только переменные и функции, объявленные как `public`

3. Что называется конструктором?

- \* метод, имя которого совпадает с именем класса и который вызывается автоматически при создании объекта класса

- метод, имя которого совпадает с именем класса и который вызывается автоматически при объявлении класса (до создания объекта класса)
- метод, имя которого необязательно совпадает с именем класса и который вызывается при создании объекта класса
- метод, имя которого совпадает с именем класса и который необходимо явно вызывать из головной программы при объявлении объекта класса

#### 4. Объект - это

- переменная, содержащая указатель на класс
- \* экземпляр класса
- класс, который содержит в себе данные и методы их обработки

#### 5. Отметьте правильные утверждения

- \* конструкторы класса не наследуются
- конструкторов класса может быть несколько, их синтаксис определяется программистом
- \* конструкторов класса может быть несколько, но их синтаксис должен подчиняться правилам перегрузки функций
- конструктор возвращает указатель на объект
- \* конструктор не возвращает значение

#### 6. Что называется деструктором?

- метод, который уничтожает объект
- метод, который удаляет объект
- \* метод, который освобождает память, занимаемую объектом
- системная функция, которая освобождает память, занимаемую объектом

#### 7. Выберите правильные утверждения

- \* у конструктора могут быть параметры
- конструктор наследуется, но должен быть перегружен
- конструктор должен явно вызываться всегда перед объявлением объекта
- \* конструктор вызывается автоматически при объявлении объекта
- объявление каждого класса должно содержать свой конструктор
- \* если конструктор не создан, компилятор создаст его автоматически

#### 8. Выберите правильные утверждения

- деструктор - это метод класса, применяемый для удаления объекта
- \* деструктор - это метод класса, применяемый для освобождения памяти, занимаемой объектом
- деструктор - это отдельная функция головной программы, применяемая для освобождения памяти, занимаемой объектом
- \* деструктор не наследуется
- деструктор наследуется, но должен быть перегружен

#### 9. Что называется наследованием?

- \* это механизм, посредством которого производный класс получает элементы родительского и может дополнять либо изменять их свойства и методы
- это механизм переопределения методов базового класса
- это механизм, посредством которого производный класс получает все поля базового класса
- это механизм, посредством которого производный класс получает элементы родительского, может их дополнить, но не может переопределить

10. Выберите правильное объявление производного класса

- `class MoreDetails:: Details;`
- `class MoreDetails: public class Details;`
- \* `class MoreDetails: public Details;`
- `class MoreDetails: class(Details);`

11. Выберите правильные утверждения:

- если элементы класса объявлены как `private`, то они доступны только наследникам класса, но не внешним функциям
- \* если элементы класса объявлены как `private`, то они недоступны ни наследникам класса, ни внешним функциям
- если элементы объявлены как `public`, то они доступны наследникам класса, но не внешним функциям
- \* если элементы объявлены как `public`, то они доступны и наследникам класса, и внешним функциям

12. Возможность и способ обращения производного класса к элементам базового определяется

- ключами доступа: `private`, `public`, `protected` в теле производного класса
- только ключом доступа `protected` в заголовке объявления производного класса
- \* ключами доступа: `private`, `public`, `protected` в заголовке объявления производного класса
- ключами доступа: `private`, `public`, `protected` в теле базового класса

13. Выберите правильные соответствия между спецификатором базового класса, ключом доступа в объявлении производного класса и правами доступа производного класса к элементам базового

- ключ доступа - `public`; в базовом классе: `private`; права доступа в производном классе - `protected`
- \* ключ доступа - любой; в базовом классе: `private`; права доступа в производном классе - нет прав
- \* ключ доступа - `protected` или `public`; в базовом классе: `protected`; права доступа в производном классе - `protected`
- ключ доступа - `private`; в базовом классе: `public`; права доступа в производном классе - `public`
- \* ключ доступа – любой; в базовом классе: `public`; права доступа в производном классе – такие же, как ключ доступа

14. Дружественная функция - это

- функция другого класса, среди аргументов которой есть элементы данного класса
- \* функция, объявленная в классе с атрибутом `friend`, но не являющаяся членом класса;
- функция, являющаяся членом класса и объявленная с атрибутом `friend`;
- функция, которая в другом классе объявлена как дружественная данному

15. Выберите правильные утверждения:

- \* одна функция может быть дружественной нескольким классам
- дружественная функция не может быть обычной функцией, а только методом другого класса
- \* дружественная функция объявляется внутри класса, к элементам которого ей нужен доступ
- дружественная функция не может быть методом другого класса

16. Шаблон функции - это...

- \* определение функции, в которой типу обрабатываемых данных присвоено условное обозначение
- прототип функции, в котором вместо имен параметров указан условный тип
- определение функции, в котором указаны возможные варианты типов обрабатываемых параметров
- определение функции, в котором в прототипе указан условный тип, а в определении указаны варианты типов обрабатываемых параметров

17. Выберите правильные утверждения:

- \* по умолчанию члены класса имеют атрибут private
- по умолчанию члены класса имеют атрибут public;
- члены класса имеют доступ только к элементам public;
- \* элементы класса с атрибутом private доступны только членам класса

18. Переопределение операций имеет вид:

- имя\_класса, ключевое слово operation, символ операции
- \* имя\_класса, ключевое слово operator, символ операции, в круглых скобках могут быть указаны аргументы
- имя\_класса, ключевое слово operator, список аргументов
- имя\_класса, два двоеточия, ключевое слово operator, символ операции

19. Для доступа к элементам объекта используются:

- \* при обращении через имя объекта – точка, при обращении через указатель – операция «->»
- при обращении через имя объекта – два двоеточия, при обращении через указатель – операция «точка»
- при обращении через имя объекта – точка, при обращении через указатель – два двоеточия
- при обращении через имя объекта – два двоеточия, при обращении через указатель – операция «->»

20. Полиморфизм – это :

- \* средство, позволяющее использовать одно имя для обозначения действий, общих для родственных классов
- средство, позволяющее в одном классе использовать методы с одинаковыми именами;
- средство, позволяющее в одном классе использовать методы с разными именами для выполнения одинаковых действий
- средство, позволяющее перегружать функции для работы с разными типами или разным количеством аргументов.

21. Полиморфизм реализован через механизмы:

- \* перегрузки функций, виртуальных функций, шаблонов
- перегрузки функций, наследования методов, шаблонов;
- наследования методов, виртуальных функций, шаблонов
- перегрузки функций, наследования, виртуальных функций.

22. Виртуальными называются функции:

- \* функции базового класса, которые могут быть переопределены в производном классе
- функции базового класса, которые не используются в производном классе;
- функции базового класса, которые не могут быть переопределены в базовом классе;
- функции производного класса, переопределенные относительно базового класса

- доклад / конференция / реферат (шкала: значение от 0 до 10, количество: 1)

### **Примерное задание:**

Тематика рефератов:

1. Краткий обзор видов современного ПО.
2. Обзор требований к современному ПО.
3. Обзор современных платформ для разработки ПО.
4. Многоярусные архитектуры ПО.
5. Многоклиентские архитектуры ПО.
6. Обзор сервисно-ориентированных архитектур ПО.
7. Обзор концепции Software as a Service.
8. Облачные вычисления как пример многоярусной, многоклиентской и сервисно-ориентированной архитектуры.

1. Основные процессы жизненного цикла программных систем.
2. Вспомогательные процессы жизненного цикла программных систем.
3. Организационные процессы жизненного цикла программных систем.
4. Взаимосвязь между процессами жизненного цикла программного обеспечения.
5. Модели и стадии жизненного цикла программного обеспечения.
6. Виды моделей жизненного цикла программного обеспечения.

1. Качественные методы описания систем.
2. Модели процессов и систем на основ декомпозиции и агрегирования.
3. Объектно-ориентированное моделирование информационных процессов и систем диаграммами uml.
4. Процессно-ориентированное моделирование информационных процессов и систем.
5. Количественное описание информационных процессов и систем.
6. История языка UML.
7. Необходимость UML. Визуализация. Спецификация. Конструирование. Документирование.
8. Какой должна быть хорошая модель информационной системы с точки зрения UML.
9. Представление системы в UML: архитектура, описание прецедентов, представление проектирования, представление процессов, представление реализации, представление развертывания.
10. Фазы и основные вехи процесса разработки информационных систем в UML. (Начало, развитие, конструирование, передача).
11. Дисциплины в UML: определение требований, анализ, проектирование, реализация, тестирование.
12. Итерации и инкременты в UML.
13. Выявление понятий, относящихся к проблеме реального мира. Объекты и классы в UML.

14. Отношения между классами в UML: ассоциация, агрегация, обобщение, классы ассоциаций.
15. Диаграммы классов. Диаграммы объектов.
16. Примечания. Пакеты.
17. Определение требований к информационным системам. Исполнители и прецеденты.
18. Диаграммы прецедентов.
19. Поток событий. Основной ход развития событий и альтернативные потоки.
20. Организация прецедентов (включение, расширение, обобщение прецедентов). Пакеты и прецеденты.
21. Анализ устойчивости. Пограничный объект. Объект-сущность. Управляющий объект. Выявление объектов анализа.
22. Сообщения и действия (вызов и возврат, создание и уничтожение, отправка).

- устный опрос / собеседование (шкала: значение от 0 до 10, количество: 1)

**Примерное задание:**

1. Перечислите виды современного программного обеспечения (ПО).
2. Перечислите требования к современному ПО
3. Перечислите характерные черты современного ПО
4. Какие Вы знаете современные платформы для разработки ПО?
5. Назовите наиболее важные качества программных продуктов.
6. Каковы три классических элемента любой технологии программирования.
7. Что такое клиент-серверная архитектура и ее компоненты - клиент и сервер?
8. Что такое тонкий клиент?
9. Что такое полнофункциональный клиент?
10. Что такое слой ПО?
11. Что такое уровень абстракции ПО?
12. Что такое вертикальный срез (слой) ПО?
13. Что такое ПО промежуточного уровня (middleware)?
14. Что такое ярус ПО (tier)?
15. В чем основная идея многоярусной архитектуры ПО?
16. Что такое многоклиентская (multi-tenant) архитектура ПО?
17. Что такое сервисно-ориентированная архитектура?
18. В чем суть концепции Software as a Service (SaaS)?

1. Понятие жизненного цикла программного продукта.
2. Этапы жизненного цикла.
3. Международный стандарт ISO/IEC 12207.
4. Каскадная модель жизненного цикла
5. Спиральная модель жизненного цикла
6. Модель жизненного цикла с контролем

1. Опишите процедуру Дельфи-метода.
2. Назовите, какие модели берутся за основания декомпозиции.
3. Дайте определение UML.
4. Объясните, из чего исходит функциональная модель.
5. Перечислите типы шкалы.
6. Правила языка UML. Механизмы и расширения языка UML.
7. Основные типы диаграмм в UML. Диаграммы прецедентов. Сценарии.
8. Основные типы диаграмм в UML. Диаграммы взаимодействия и их виды.

9. Основные типы диаграмм в UML. Диаграммы состояний.
10. Основные типы диаграмм в UML. Диаграммы классов.
11. Особенности жизненного цикла с использованием UML.
12. Диаграммы последовательностей. Указание времени на диаграммах.
13. Диаграммы кооперации. Пакеты анализа и проектирования.
14. Абстрактные классы. Зависимости. Атрибуты и операции. Видимость.
15. Расширение языка UML. Стереотип. Ограничения. Тегированные значения.
16. Интерфейсы и классы. Параметризованные классы. Классы проектирования и пакеты.
17. Описание потоков. Виды деятельности и действия. Ветвление и слияние. Распараллеливание и объединение.
18. Диаграмма видов деятельности. Дорожки. Изменение объектов. Процессы, потоки и активные объекты. Диаграммы видов деятельности и пакеты.
19. События. Состояния, переходы, условия. Машины и диаграммы состояний.
20. Составные состояния. Последовательные подсостояния. Запоминающее состояние. Параллельные подсостояния. Диаграммы состояний и пакеты.
21. Взаимодействие групп объектов. Группы, шаблоны, механизмы и контуры. Системы и подсистемы.
22. Компоненты. Диаграммы компонентов. Узлы. Диаграммы развертывания. Физические диаграммы и пакеты.

1. Объектно-ориентированный подход (ООП).
2. Основные понятия ООП.
3. Определение ООП и его основные концепции.
4. Недостатки ООП.
5. Основы ООП-методологии.
6. Концепция MVC
7. Основные понятия MVC.
8. Определение MVC и его основные концепции.
9. Достоинства и недостатки MVC
10. Назначение MVC.

- кейс (шкала: значение от 0 до 10, количество: 1)

### **Примерное задание:**

Кейс-задание:

Проектирование архитектуры программного обеспечения.

Для формирования структуры программного обеспечения произведите проектирование его архитектуры.

Архитектурный вид проектирования состоит из двух основных компонентов: структурных элементов и отношений между ними. При этом схематически архитектура может быть представлена в логическом или физическом виде.

Если структурные элементы представляют собой концептуальные (логические) компоненты ПО, такие как прецеденты, классы, процессы, состояния и др., то такая модель программной архитектуры будет соответствовать логической. Физическая же архитектура будет являться реализацией логической структуры программной системы, когда ее элементами станут физически существующие компоненты, такие как программы, базы и файлы данных, интерфейсы и т. д.

Основные этапы, которые являются общими для любых процессов проектирования системной архитектуры:



- 1) структурирование системы, на этом этапе система структурируется в виде набора относительно независимых подсистем, при этом определяются виды взаимодействия между ними;
- 2) моделирование управления, при этом создается базовая модель управления отношениями между элементами системы;
- 3) модульная декомпозиция – этап, на котором определенные на первом этапе подсистемы разбиваются на отдельные процессы (модули), а также определяются их типы и взаимные связи.

Как правило, разрабатывается четыре типа моделей архитектуры ПО:

- 1) статическая структурная модель, которая отображает подсистемы или компоненты, разрабатываемые на следующих этапах независимо;
- 2) динамическая модель процессов, представляющая организацию процессов во время функционирования системы;
- 3) интерфейсная модель, устанавливающая сервисы, предоставляемые каждой подсистемой через общесистемный интерфейс;
- 4) модели отношений, показывающие взаимодействия между элементами системы, такие как потоки данных.

Кейс-задание:

В соответствии со стандартом ISO/IEC 12207 все процессы ЖЦ ПО разделены на три группы:

Основные процессы, вспомогательные процессы, организационные процессы.

В группах определено пять основных процессов: приобретение, поставка, разработка, эксплуатация и сопровождение. Восемь вспомогательных процессов обеспечивают выполнение основных процессов, а именно документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместная оценка, аудит, разрешение проблем. Четыре организационных процесса обеспечивают управление, создание инфраструктуры, совершенствование и обучение.

Опишите процесс разрабатываемого программного обеспечения по следующей схеме:

1. подготовительную работу;
2. анализ требований, предъявляемых к системе;
3. проектирование архитектуры системы;
4. анализ требований, предъявляемых к программному обеспечению;
5. проектирование архитектуры программного обеспечения;
6. детальное проектирование программного обеспечения;
7. кодирование и тестирование программного обеспечения;
8. интеграцию программного обеспечения;
9. квалификационное тестирование программного обеспечения;
10. интеграцию системы;
11. квалификационное тестирование системы;
12. установку программного обеспечения;
13. приемку программного обеспечения.

Кейс-задание:

Применение UML для выполнения этапов анализа и проектирования ПС.

1. Построить информационную модель (ИМ) информационной системы, для чего:

- 1.1. Определить основные объекты (сущности), входящие в систему, и их атрибуты (идентификаторы, привилегированные идентификаторы и описательные атрибуты).
- 1.2. Определить или задать связи между объектами, их типы и условность.

- 1.3. Формализовать связи. С этой целью, в зависимости от типа связи,
  - 1.3.1. Ввести вспомогательные атрибуты в соответствующие отношения;
  - 1.3.2. Добавить ассоциативные объекты в ИМ.
2. Построить модели состояний каждого объекта с использованием графического представления диаграммы Мура. Для этого:
  - 2.1. Определить возможные состояния объекта;
  - 2.2. Определить события, переводящие объект из одного состояния в другое;
  - 2.3. Задать действия, выполняемые объектом в каждом из состояний, в виде укрупненного алгоритма.
3. Построить модель взаимодействия объектов и модель доступа к объектам и проконтролировать корректность порождения и приема событий, для чего, при необходимости построить схему канала управления.
4. Построить модели процессов для каждого действия каждого объекта системы в виде диаграммы потоков данных действий, для чего:
  - 4.1. Определить процессы и их типы – аксессоры, генераторы событий, преобразователи, процессы проверки;
  - 4.2. Определить условные и безусловные потоки данных и управления.
5. Построить диаграмму класса для каждого объекта системы, для чего:
  - 5.1. Определить логические компоненты, исходя из атрибутов объектов;
  - 5.2. На основе моделей состояний и моделей процессов определить общедоступные операции, разделив операции, базируемые на классах и базируемые на экземплярах;
  - 5.3. Задать входы и выходы операций, исходя из данных, переносимых соответствующими событиями.
6. Для каждого объекта построить схему структуры класса, основываясь на моделях процессов.
7. Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам.

Варианты заданий:

1. Система поликлиники
2. Система малого предприятия
3. Система организационного управления
4. Система парикмахерской
5. Система библиотеки
6. Система управления учебным процессом
7. Система "Клиент-Банк"
8. Универсальная система "Склад"
9. Система столовой
10. Система СТО
11. Система автозаправки
12. Система управления расписанием школы
13. Система питания детского сада
14. Система закупки товаров аптеки
15. Система швейного ателье
16. Система газетного киоска
17. Система оформления подписки на почте

## **5. Формы промежуточной аттестации**

- зачет - 1 курс, 2 семестр (шкала: значение от 0 до 20)

**Примерное задание:**

## Вопрос №1.

1. Классы задач, решаемых современным программным обеспечением.
2. Характеристики и свойства современного ПО.
3. Определения основных терминов, связанных с видами архитектур программного обеспечения .
4. Требования к современному программному обеспечению.
5. ИКТ и вычислительное оборудование как инструментарий автоматизации и информатизации прикладных задач .
6. Клиент-серверная архитектура и ее компоненты - клиент и сервер .
7. Современные платформы для разработки ПО .
8. Понятие жизненного цикла программного продукта(ПП).
9. Этапы жизненного цикла ПП.
10. Международный стандарт ISO/IEC 12207.
11. Модели жизненного цикла(каскадная, с промежуточным контролем, спиральная).
12. Преимущества и недостатки моделей жизненного цикла, области их применения.
13. Основные подходы к разработке программного обеспечения.
14. Структурный подход к разработке программного обеспечения, основанный на принципе функциональной декомпозиции.
15. Объектно-ориентированный подход к разработке программного обеспечения, основанный на объектной декомпозиции.
16. Принципы объектно-ориентированного представления программных систем(основные-абстрагирование, абстрагирование, инкапсуляция, модульность, иерархия; дополнительные – типизация, параллелизм, устойчивость).
17. Основные понятия объектно-ориентированного подхода(объект, класс, данные, ме-тоды, доступ, наследование свойств, системы объектов и классов)
18. Определение объектно-ориентированного программирования как метода построения программ в виде множества взаимодействующих объектов.
19. Определение объектно-ориентированного языка программирования.
20. Деление объектных языков на языки, использующие объекты, и объектно-ориентированные языки.
21. История развития объектно-ориентированных языков программирования.
22. Назначение и история языка UML.
23. Структура языка UML (сущности, отношения, диаграммы).
24. Основные сущности языка UML.
25. Отношения языка UML.

## Вопрос №2.

1. Диаграммы языка UML. Их назначение, структура, правила построения.
2. Представление архитектуры программных систем(ее видов) диаграммами языка UML.
3. Моделирование программных систем(ее видов) диаграммами UML: а) спецификация разрабатываемого ПО да этапе анализа; б) диаграммы вариантов использования (элементы, актеры, отношения); в) диаграммы классов(уровни использования диаграмм; класс как основное понятие диаграмм; отношение классов; проектирование классов; наследова-ние); г) диаграммы последовательностей(уровни использования диаграмм; диаграммы последовательностей этапов анализа и проектирования); д) диаграмма деятельности(понятие деятельности, вершины диаграмм деятельности).
4. CASE-средства построения UML-диаграмм.
5. Унифицированный процесс разработки программного обеспечения, его базовые принципы.
6. Последовательность действий при анализе: составление и документирование функциональных и нефункциональных требований.

7. Проектирование программного обеспечения, переход от диаграмм анализа к диаграммам проектирования.
8. Выполнение этапов реализации и проверки программного обеспечения.
9. Паттерны проектирования.
10. Понятие тестирования, верификации, валидации.
11. Организация процесса тестирования программного обеспечения.
12. Тестирование объектно-ориентированных программных систем:
  - а) расширение области применения объектно-ориентированных программных средств;
  - б) изменение методик при объектно-ориентированном тестировании;
  - в) проектирование объектно-ориентированных тестовых вариантов.
13. Тенденции развития технологии разработки ПО.
14. Паттерны проектирования и их представление в нотации UML.
15. Архитектурные паттерны. Примеры
16. Паттерны проектирования. Примеры
17. Паттерны анализа. Примеры
18. Паттерны тестирования. Примеры
19. Паттерны реализации. Примеры
20. Основы ООП-методологии.
21. Концепция MVC
22. Основные понятия MVC.
23. Определение MVC и его основные концепции.
24. Достоинства и недостатки MVC
25. Назначение MVC.

**Критерии оценивания:**

18-20 баллов: Обучающийся, достигающий должного уровня:

- даёт полный, глубокий, выстроенный логично по содержанию вопроса ответ, используя различные источники информации, не требующий дополнений
- доказательно иллюстрирует основные теоретические положения практическими примерами;
- способен глубоко анализировать теоретический и практический материал, обобщать его, самостоятельно делать выводы, вести диалог и высказывать свою точку зрения.

14-17 баллов: Обучающийся на должном уровне:

- раскрывает учебный материал: даёт содержательно полный ответ, требующий незначительных дополнений и уточнений, которые он может сделать самостоятельно после наводящих вопросов преподавателя;
- демонстрирует учебные умения и навыки в области решения практико-ориентированных задач;
- владеет способами анализа, сравнения, обобщения и обоснования выбора методов решения практико-ориентированных задач.

11-13 баллов: Достигнутый уровень оценки результатов обучения обучающегося показывает:

- знания имеют фрагментарный характер, отличаются поверхностностью и малой содержательностью; студент раскрывает содержание вопроса, но не глубоко, бессистемно, с некоторыми неточностями;
- слабо, недостаточно аргументированно может обосновать связь теории с практикой;
- способен понимать и интерпретировать основной теоретический материал по дисциплине.

0-10 баллов: Результаты обучения обучающегося свидетельствуют:

- об усвоении им некоторых элементарных знаний, но студент не владеет понятийным аппаратом изучаемой образовательной области (учебной дисциплины);

- не умеет установить связь теории с практикой;
- не владеет способами решения практико-ориентированных задач.

## 6. Балльная система оценивания по дисциплине

ОФО

Семестр (Курс) - 2 (1)			
Форма текущего контроля	Раздел дисциплины	Максимальный балл	Максимальный приведенный балл
доклад / конференция / реферат	Раздел 1. Архитектура современных программных систем	10	
доклад / конференция / реферат	Раздел 2. Технологии разработки программного обеспечения	10	
кейс	Раздел 1. Архитектура современных программных систем	10	
кейс	Раздел 2. Технологии разработки программного обеспечения	10	
тест по итогам занятия	Раздел 1. Архитектура современных программных систем	10	
тест по итогам занятия	Раздел 2. Технологии разработки программного обеспечения	10	
устный опрос / собеседование	Раздел 1. Архитектура современных программных систем	10	
устный опрос / собеседование	Раздел 2. Технологии разработки программного обеспечения	10	
Максимальный текущий балл		80	80
<b>Промежуточная аттестация</b>		зачет	
Максимальный аттестационный балл		20	20
Общий балл по дисциплине		100	100

Общий балл по дисциплине за семестр складывается из результатов, полученных по формам текущего контроля в течение семестра и аттестационного балла.

Оценка успеваемости по дисциплине в семестре пересчитывается по приведенной 100-балльной шкале независимо от шкалы, определенной преподавателем.

Перевод баллов из 100-балльной шкалы в числовой и буквенный эквивалент:

**- для зачета:**

Сумма баллов	Отметка
--------------	---------

51-100	Зачтено
0-50	Не зачтено

## 7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины. Электронно-библиотечные системы

### *Основная литература*

1. Ткаченко, С. Н., Методы и средства проектирования информационных систем и технологий + eПриложение : учебник / С. Н. Ткаченко, Б. Р. Мищук. — Москва : КноРус, 2022. — 222 с. — ISBN 978-5-406-09467-9. — URL: <https://book.ru/book/943815>. — Текст : электронный.
2. Лехмус, М. Ю., Базовые технологии веб-программирования : учебное пособие / М. Ю. Лехмус. — Москва : КноРус, 2025. — 86 с. — ISBN 978-5-406-14103-8. — URL: <https://book.ru/book/956632>. — Текст : электронный.
3. Веретехина, С. В., Информационные технологии. Пакеты программного обеспечения общего блока «ИТ-инструментарий» : учебное пособие / С. В. Веретехина, В. В. Веретехин. — Москва : Русайнс, 2024. — 43 с. — ISBN 978-5-466-04610-6. — URL: <https://book.ru/book/953011>. — Текст : электронный.

### *Дополнительная литература*

1. Назаров, С. В., Программное обеспечение систем реального времени : монография / С. В. Назаров. — Москва : Русайнс, 2022. — 211 с. — ISBN 978-5-4365-9770-6. — URL: <https://book.ru/book/944829>. — Текст : электронный.
2. Горюшкин, А. А., Офисное программное обеспечение : учебное пособие / А. А. Горюшкин. — Москва : Русайнс, 2025. — 118 с. — ISBN 978-5-466-07792-6. — URL: <https://book.ru/book/955486>. — Текст : электронный.

## 8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

Обучающимся (магистрам) обеспечен доступ (удаленный доступ), в том числе в случае применения электронного обучения, дистанционных образовательных технологий, к современным профессиональным базам данных и информационным справочным системам (*подлежащим обновлению при необходимости*), а именно:

1. КонсультантПлюс: справочно-поисковая система [Электронный ресурс]. - <http://www.consultant.ru>
2. Мировая цифровая библиотека: <http://wdl.org/ru>
3. Научная электронная библиотека «Scopus»: <https://www.scopus.com>
4. Научная электронная библиотека ScienceDirect: <http://www.sciencedirect.com>
5. Научная электронная библиотека «eLIBRARY»: <https://elibrary.ru>
6. Портал «Гуманитарное образование» <http://www.humanities.edu.ru>
7. Федеральный портал «Российское образование» <http://www.edu.ru>
8. Федеральное хранилище «Единая коллекция цифровых образовательных ресурсов» <http://school-collection.edu.ru>
9. Поисковые системы Yandex, Rambler и др.
10. Электронная библиотека Российской Государственной Библиотеки (РГБ): <http://elibrary.rsl.ru>
11. Электронно-библиотечная система <http://www.sciteclibrary.ru>

## 9. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Помещения для проведения всех видов работы, предусмотренных учебным планом, укомплектованы необходимым оборудованием и техническими средствами обучения.

<b>Наименование оборудованных учебных кабинетов, объектов для проведения практических занятий, объектов физической культуры и спорта с перечнем основного оборудования</b>	<b>Адрес (местоположение) учебных кабинетов, объектов для проведения практических занятий, объектов физической культуры и спорта (с указанием площади и номера помещения в соответствии с документами бюро технической инвентаризации)</b>	<b>Собственность или оперативное управление, хозяйственное ведение, аренда (субаренда), безвозмездное пользование, практическая подготовка</b>	<b>Полное наименование собственника (арендодателя, ссудодателя) объекта недвижимого имущества</b>	<b>Документ – основание возникновения права (реквизиты и срок действия)</b>
<b>Специализированная многофункциональная учебная аудитория для проведения учебных занятий лекционного и семинарского типов, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, в том числе, для организации практической подготовки обучающихся, с перечнем основного оборудования (аудитория № 3): Столы для обучающихся;</b>	170001, Тверская область, г. Тверь, ул. Спартака, д. 26а (39,2 кв.м., 1 этаж, помещение № 3)	Безвозмездное пользование	Богачев Сергей Александрович	Договор безвозмездного пользования недвижимым имуществом № 01-18/Н от 01.11.2020 с приложениями №№ 1-3; срок действия договора: с 01.11.2020 по 30.09.2025

<p>Стулья для обучающихся;</p> <p>Стол педагогического работника;</p> <p>Стул педагогического работника;</p> <p>Компьютер с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду лицензиата;</p> <p>Интерактивная доска;</p> <p>Проектор</p>				
<p><b>Специализированная многофункциональная учебная аудитория для проведения учебных занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, в том числе, для организации практической подготовки обучающийся, с перечнем основного оборудования (аудитория № 27)</b></p> <p>Компьютерные столы для обучающихся;</p> <p>Стулья для обучающихся;</p> <p>Стол педагогического работника;</p>	<p>170001, Тверская область, г. Тверь, ул. Спартака, д. 26а (31,1 кв.м., 2 этаж, помещение № 27)</p>	<p>Безвозмездное пользование</p>	<p>Богачев Сергей Александрович</p>	<p>Договор безвозмездного пользования недвижимым имуществом № 01-18/Н от 01.11.2020 с приложениями №№ 1-3; срок действия договора: с 01.11.2020 по 30.09.2025</p>



<p>Стул педагогического работника;</p> <p>Компьютеры с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду лицензиата;</p> <p>Интерактивная доска;</p> <p>Проектор</p> <p>Сканер;</p> <p>Принтер</p>				
<p><b>Специализированная многофункциональная учебная аудитория для проведения учебных занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, в том числе, для организации практической подготовки обучающийся, с перечнем основного оборудования (аудитория № 16)</b></p> <p>Компьютерные столы для обучающихся;</p> <p>Стулья для обучающихся;</p> <p>Стол педагогического работника;</p> <p>Стул педагогического работника;</p>	<p>170001, Тверская область, г. Тверь, ул. Спартака, д. 26а (31,4 кв.м., 2 этаж, помещение № 16)</p>	<p>Безвозмездное пользование</p>	<p>Богачев Сергей Александрович</p>	<p>Договор безвозмездного пользования недвижимым имуществом № 01-18/Н от 01.11.2020 с приложениями №№ 1-3; срок действия договора: с 01.11.2020 по 30.09.2025</p>

Компьютеры с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду лицензиата; Интерактивная доска; Проектор Сканер; Принтер				
<b>Помещение для самостоятельной работы обучающихся с перечнем основного оборудования</b> (аудитория № 22): Столы для обучающихся; Стулья для обучающихся; Компьютеры с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду лицензиата; Ноутбуки с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду лицензиата; Принтер; Сканер	170001, Тверская область, г. Тверь, ул. Спартака, д. 26а (19,3 кв.м., 2 этаж, помещение № 22)	Безвозмездное пользование	Богачев Сергей Александрович	Договор безвозмездного пользования недвижимым имуществом № 01-18/Н от 01.11.2020 с приложениями №№ 1-3; срок действия договора: с 01.11.2020 по 30.09.2025
<b>Помещение для самостоятельной работы обучающихся с</b>	170001, Тверская область, г. Тверь,	Безвозмездное пользование	Богачев Сергей Александрович	Договор безвозмездного пользования

<p><b>перечнем основного оборудования</b> (аудитория № 14):  Столы для обучающихся;  Стулья для обучающихся;  Компьютеры с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду лицензиата;  Ноутбуки с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду лицензиата;  Принтер;  Сканер</p>	<p>ул. Спартака,  д. 26а  (22,5 кв.м.,  1 этаж,  помещение № 14)</p>			<p>недвижимым имуществом  № 01-18/Н  от 01.11.2020  с приложениями  №№ 1-3;  срок действия договора:  с 01.11.2020  по 30.09.2025</p>
<p><b>Помещение для самостоятельной работы обучающихся с перечнем основного оборудования</b> (аудитория № 31):  Столы для обучающихся;  Стулья для обучающихся;  Компьютеры с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду лицензиата;</p>	<p>170001,  Тверская область,  г. Тверь,  ул. Спартака,  д. 26а  (20,3 кв.м.,  2 этаж,  помещение № 31)</p>	<p>Безвозмездное пользование</p>	<p>Богачев Сергей Александрович</p>	<p>Договор безвозмездного пользования недвижимым имуществом  № 01-18/Н  от 01.11.2020  с приложениями  №№ 1-3;  срок действия договора:  с 01.11.2020  по 30.09.2025</p>

Ноутбуки с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду лицензиата; Принтер; Сканер				
<b>Помещение для организации практической подготовки обучающихся с перечнем основного оборудования Кабинет заместителя генерального директора № 5:</b> письменный стол - 1 шт.; офисное кресло - 1 шт.; стул – 4 шт.; персональный компьютер - 1 шт.; монитор - 1 шт.; тумба - 2 шт.; стационарный телефон - 1 шт.; настольная лампа - 1 шт.; многофункциональное устройство (принтер, сканер, ксерокс) – 1 шт.; стеллаж - 4 шт.; шкаф - 4 шт.; вешалка - 1 шт.	170100, Тверская область, г. Тверь, ул. Советская, д. 28 (18,6 кв.м., 3 этаж, помещение № 3-4)	Практическая подготовка	ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ «ДСМЛ»	<b>Договор № 1-ПрИн</b> о практической подготовке обучающихся, заключенный между организацией, осуществляющей образовательную деятельность, и организацией, осуществляющей деятельность по профилю образовательной программы от 29.08.2024; срок действия договора – 5 лет
<b>Помещение для организации практической подготовки обучающихся с перечнем основного оборудования</b>	170100, Тверская область, г. Тверь, ул. Советская, д. 28 (15,4 кв.м., 3 этаж,	Практическая подготовка	ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ «ДСМЛ»	<b>Договор № 1-ПрИн</b> о практической подготовке обучающихся, заключенный между организацией,

<p><b>Кабинет специалистов по стратегическому управлению информационными технологиями № 7:</b></p> <p>письменный стол - 2 шт.; офисное кресло - 2 шт.; персональный компьютер - 2 шт.; монитор - 2 шт.; тумба - 4 шт.; стационарный телефон - 2 шт.; настольная лампа - 2 шт.; многофункциональное устройство (принтер, сканер, ксерокс) – 1 шт.; стеллаж - 4 шт.; шкаф - 4 шт.; вешалка - 1 шт.</p>	<p>помещение № 3-6)</p>			<p>осуществляющей образовательную деятельность, и организацией, осуществляющей деятельность по профилю образовательной программы от 29.08.2024; срок действия договора – 5 лет</p>
<p><b>Помещение для организации практической подготовки обучающихся с перечнем основного оборудования Кабинет специалистов по разработке компьютерного программного обеспечения № 8:</b></p> <p>письменный стол - 4 шт.; офисное кресло - 4 шт.; персональный компьютер - 4 шт.; монитор - 4 шт.; тумба - 8 шт.; стационарный телефон - 4 шт.;</p>	<p>170100, Тверская область, г. Тверь, ул. Советская, д. 28 (18,8 кв.м., 3 этаж, помещение № 3-7)</p>	<p>Практическая подготовка</p>	<p>ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ «ДСМЛ»</p>	<p><b>Договор № 1-ПрИн</b> о практической подготовке обучающихся, заключенный между организацией, осуществляющей образовательную деятельность, и организацией, осуществляющей деятельность по профилю образовательной программы от 29.08.2024; срок действия договора – 5 лет</p>

настольная лампа - 4 шт.; многофункциональное устройство (принтер, сканер, ксерокс) – 1 шт.; принтер - 3 шт.; стеллаж - 6 шт.; шкаф - 4 шт.; вешалка - 1 шт.				
<b>Помещение для организации практической подготовки обучающихся с перечнем основного оборудования Консультативный кабинет в области компьютерных технологий № 9:</b> письменный стол - 1 шт.; офисное кресло - 1 шт.; стул – 1 шт.; персональный компьютер - 1 шт.; монитор - 1 шт.; тумба - 2 шт.; стационарный телефон - 1 шт.; настольная лампа - 1 шт.; принтер - 1 шт.; вешалка - 1 шт.	170100, Тверская область, г. Тверь, ул. Советская, д. 28 (12,2 кв.м., 3 этаж, помещение № 3-10)	Практическая подготовка	ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ «ДСМЛ»	<b>Договор № 1-ПрИн</b> о практической подготовке обучающихся, заключенный между организацией, осуществляющей образовательную деятельность, и организацией, осуществляющей деятельность по профилю образовательной программы от 29.08.2024; срок действия договора – 5 лет
<b>Помещение для организации практической подготовки обучающихся с перечнем основного оборудования Кабинет специалистов по обработке данных,</b>	170100, Тверская область, г. Тверь, ул. Советская, д. 28 (16,6 кв.м., 3 этаж, помещение № 3-13)	Практическая подготовка	ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ «ДСМЛ»	<b>Договор № 1-ПрИн</b> о практической подготовке обучающихся, заключенный между организацией, осуществляющей образовательную деятельность, и организацией, осуществляющей деятельность

<p><b>предоставлению услуг по размещению информации, деятельности порталов в информационно-коммуникационной сети Интернет № 11:</b></p> <p>письменный стол - 2 шт.;  офисное кресло - 2 шт.;  персональный компьютер - 2 шт.;  монитор - 2 шт.;  тумба - 4 шт.;  стационарный телефон - 2 шт.;  настольная лампа - 2 шт.;  многофункциональное устройство (принтер, сканер, ксерокс) – 1 шт.;  принтер - 1 шт.;  стеллаж - 3 шт.;  шкаф - 3 шт.;  вешалка - 1 шт.</p>				<p>по профилю образовательной программы от 29.08.2024; срок действия договора – 5 лет</p>
<p><b>Помещение для организации практической подготовки обучающихся с перечнем основного оборудования Кабинет специалистов по созданию и использованию баз данных и информационных ресурсов № 12:</b></p> <p>письменный стол - 2 шт.;  офисное кресло - 2 шт.;  персональный компьютер - 2 шт.;  монитор - 2 шт.;</p>	<p>170100, Тверская область, г. Тверь, ул. Советская, д. 28 (16,4 кв.м., 3 этаж, помещение № 3-14)</p>	<p>Практическая подготовка</p>	<p>ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ «ДСМЛ»</p>	<p><b>Договор № 1-ПрИн</b>  о практической подготовке обучающихся, заключенный между организацией, осуществляющей образовательную деятельность, и организацией, осуществляющей деятельность по профилю образовательной программы от 29.08.2024; срок действия договора – 5 лет</p>

<p>тумба - 4 шт.;  стационарный телефон - 2 шт.;  настольная лампа - 2 шт.;  многофункциональное устройство (принтер, сканер, ксерокс) – 1 шт.;  принтер - 1 шт.;  стеллаж - 3 шт.;  шкаф - 2 шт.;  вешалка - 1 шт.</p>				
<p><b>Помещение для организации практической подготовки обучающихся с перечнем основного оборудования</b>  <b>Кабинет менеджера по цифровой трансформации № 14:</b>  письменный стол - 1 шт.;  офисное кресло - 1 шт.;  стул – 2 шт.;  персональный компьютер - 1 шт.;  монитор - 1 шт.;  тумба - 2 шт.;  стационарный телефон - 1 шт.;  настольная лампа - 1 шт.;  многофункциональное устройство (принтер, сканер, ксерокс) – 1 шт.;  стеллаж - 2 шт.;  шкаф - 2 шт.;  вешалка - 1 шт.</p>	<p>170100,  Тверская область,  г. Тверь,  ул. Советская,  д. 28  (12,8 кв.м.,  3 этаж,  помещение № 3-16)</p>	<p>Практическая подготовка</p>	<p>ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ  «ДСМЛ»</p>	<p><b>Договор № 1-ПрИн</b>  о практической подготовке обучающихся, заключенный между организацией, осуществляющей образовательную деятельность, и организацией, осуществляющей деятельность по профилю образовательной программы от 29.08.2024; срок действия договора –  5 лет</p>
<p><b>Помещение для организации практической подготовки обучающихся с</b></p>	<p>170100,  Тверская область,  г. Тверь,  ул. Советская,</p>	<p>Практическая подготовка</p>	<p>ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ  «ДСМЛ»</p>	<p><b>Договор № 1-ПрИн</b>  о практической подготовке обучающихся,</p>



<p><b>перечнем основного оборудования</b>  <b>Отдел поддержки информационных технологий № 15:</b>  письменный стол - 4 шт.;  офисное кресло - 4 шт.;  персональный компьютер - 4 шт.;  монитор - 4 шт.;  тумба - 8 шт.;  стационарный телефон - 4 шт.;  настольная лампа - 4 шт.;  многофункциональное устройство (принтер, сканер, ксерокс) – 1 шт.;  принтер - 3 шт.;  стеллаж - 4 шт.;  шкаф - 4 шт.;  вешалка - 1 шт.</p>	<p>д. 28  (17,8 кв.м.,  3 этаж,  помещение № 3-18)</p>			<p>заключенный между организацией, осуществляющей образовательную деятельность, и организацией, осуществляющей деятельность по профилю образовательной программы от 29.08.2024; срок действия договора – 5 лет</p>
<p><b>Помещение для организации практической подготовки обучающихся с перечнем основного оборудования</b>  <b>Кабинет заместителя директора № 9:</b>  письменный стол - 1 шт.;  офисное кресло - 1 шт.;  стул – 3 шт.;  персональный компьютер - 1 шт.;  монитор - 1 шт.;  тумба - 2 шт.;  стационарный телефон - 1 шт.;  настольная лампа - 1 шт.;  многофункциональное устройство</p>	<p>170100,  Тверская область,  г. Тверь,  ул. Новоторжская, д.3  (16,8 кв.м.,  2 этаж,  помещение № 11)</p>	<p>Практическая подготовка</p>	<p>ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ  «ТВЕРСКИЕ КРИПТОГРАФИЧЕСКИЕ СИСТЕМЫ»</p>	<p><b>Договор № 2-ПрИн</b>  о практической подготовке обучающихся, заключенный между организацией, осуществляющей образовательную деятельность, и организацией, осуществляющей деятельность по профилю образовательной программы от 29.08.2024; срок действия договора – 5 лет</p>

(принтер, сканер, ксерокс) – 1 шт.; стеллаж - 3 шт.; шкаф - 3 шт.; вешалка - 1 шт.				
<b>Помещение для организации практической подготовки обучающихся с перечнем основного оборудования Кабинет специалистов по разработке компьютерного программного обеспечения № 7:</b> письменный стол - 4 шт.; офисное кресло - 4 шт.; персональный компьютер - 4 шт.; монитор - 4 шт.; тумба - 8 шт.; стационарный телефон - 4 шт.; настольная лампа - 4 шт.; многофункциональное устройство (принтер, сканер, ксерокс) – 1 шт.; принтер - 3 шт.; стеллаж - 4 шт.; шкаф - 4 шт.; вешалка - 1 шт.	170100, Тверская область, г. Тверь, ул. Новоторжская, д.3 (18,2 кв.м., 2 этаж, помещение № 9)	Практическая подготовка	ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ «ТВЕРСКИЕ КРИПТОГРАФИЧЕСКИЕ СИСТЕМЫ»	<b>Договор № 2-ПрИн</b> о практической подготовке обучающихся, заключенный между организацией, осуществляющей образовательную деятельность, и организацией, осуществляющей деятельность по профилю образовательной программы от 29.08.2024; срок действия договора – 5 лет
<b>Помещение для организации практической подготовки обучающихся с перечнем основного оборудования Консультативный кабинет в области компьютерных</b>	170100, Тверская область, г. Тверь, ул. Новоторжская, д.3 (12,8 кв.м., 2 этаж, помещение № 10)	Практическая подготовка	ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ «ТВЕРСКИЕ КРИПТОГРАФИЧЕСКИЕ СИСТЕМЫ»	<b>Договор № 2-ПрИн</b> о практической подготовке обучающихся, заключенный между организацией, осуществляющей образовательную деятельность,

<b>технологий № 8:</b> письменный стол - 1 шт.; офисное кресло - 1 шт.; стул – 1 шт.; персональный компьютер - 1 шт.; монитор - 1 шт.; тумба - 2 шт.; стационарный телефон - 1 шт.; настольная лампа - 1 шт.; принтер - 1 шт.; вешалка - 1 шт.				и организацией, осуществляющей деятельность по профилю образовательной программы от 29.08.2024; срок действия договора – 5 лет
<b>Помещение для организации практической подготовки обучающихся с перечнем основного оборудования</b> <b>Кабинет специалиста по планированию, проектированию компьютерных систем № 6:</b> письменный стол - 1 шт.; офисное кресло - 1 шт.; стул – 1 шт.; персональный компьютер - 1 шт.; монитор - 1 шт.; тумба - 2 шт.; стационарный телефон - 1 шт.; настольная лампа - 1 шт.; многофункциональное устройство (принтер, сканер, ксерокс) – 1 шт.; стеллаж - 2 шт.; шкаф - 2 шт.; вешалка - 1 шт.	170100, Тверская область, г. Тверь, ул. Новоторжская, д.3 (12,6 кв.м., 2 этаж, помещение № 8)	Практическая подготовка	ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ «ТВЕРСКИЕ КРИПТОГРАФИЧЕСКИЕ СИСТЕМЫ»	<b>Договор № 2-ПрИн</b> о практической подготовке обучающихся, заключенный между организацией, осуществляющей образовательную деятельность, и организацией, осуществляющей деятельность по профилю образовательной программы от 29.08.2024; срок действия договора – 5 лет

<p><b>Помещение для организации практической подготовки обучающихся с перечнем основного оборудования</b></p> <p><b>Кабинет специалистов по подготовке компьютерных систем к эксплуатации № 10:</b></p> <p>письменный стол - 3 шт.; офисное кресло - 3 шт.; персональный компьютер - 3 шт.; монитор - 3 шт.; тумба - 6 шт.; стационарный телефон - 3 шт.; настольная лампа - 3 шт.; многофункциональное устройство (принтер, сканер, ксерокс) – 1 шт.; принтер - 2 шт.; стеллаж - 3 шт.; шкаф - 3 шт.; вешалка - 1 шт.</p>	<p>170100, Тверская область, г. Тверь, ул. Новоторжская, д.3 (16,8 кв.м., 2 этаж, помещение № 12)</p>	<p>Практическая подготовка</p>	<p>ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ «ТВЕРСКИЕ КРИПТОГРАФИЧЕСКИЕ СИСТЕМЫ»</p>	<p><b>Договор № 2-ПрИн</b> о практической подготовке обучающихся, заключенный между организацией, осуществляющей образовательную деятельность, и организацией, осуществляющей деятельность по профилю образовательной программы от 29.08.2024; срок действия договора – 5 лет</p>
<p><b>Помещение для организации практической подготовки обучающихся с перечнем основного оборудования</b></p> <p><b>Кабинет специалистов по сопровождению компьютерных систем № 11:</b></p> <p>письменный стол - 2 шт.; офисное кресло - 2 шт.;</p>	<p>170100, Тверская область, г. Тверь, ул. Новоторжская, д.3 (16,2 кв.м., 2 этаж, помещение № 13)</p>	<p>Практическая подготовка</p>	<p>ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ «ТВЕРСКИЕ КРИПТОГРАФИЧЕСКИЕ СИСТЕМЫ»</p>	<p><b>Договор № 2-ПрИн</b> о практической подготовке обучающихся, заключенный между организацией, осуществляющей образовательную деятельность, и организацией, осуществляющей деятельность по профилю образовательной программы от 29.08.2024; срок</p>

персональный компьютер - 2 шт.; монитор - 2 шт.; тумба - 4 шт.; стационарный телефон - 2 шт.; настольная лампа - 2 шт.; многофункциональное устройство (принтер, сканер, ксерокс) – 1 шт.; принтер - 1 шт.; стеллаж - 2 шт.; шкаф - 2 шт.; вешалка - 1 шт.				действия договора – 5 лет
<b>Помещение для организации практической подготовки обучающихся с перечнем основного оборудования</b> <b>Кабинет специалистов по созданию и использованию баз данных и информационных ресурсов № 14:</b> письменный стол - 3 шт.; офисное кресло - 3 шт.; персональный компьютер - 3 шт.; монитор - 3 шт.; тумба - 6 шт.; стационарный телефон - 3 шт.; настольная лампа - 3 шт.; многофункциональное устройство (принтер, сканер, ксерокс) – 1 шт.; принтер - 2 шт.; стеллаж - 3 шт.; шкаф - 3 шт.; вешалка - 1 шт.	170100, Тверская область, г. Тверь, ул. Новоторжская, д.3 (17,8 кв.м., 2 этаж, помещение № 15)	Практическая подготовка	ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ «ТВЕРСКИЕ КРИПТОГРАФИЧЕСКИЕ СИСТЕМЫ»	<b>Договор № 2-ПрИн</b> о практической подготовке обучающихся, заключенный между организацией, осуществляющей образовательную деятельность, и организацией, осуществляющей деятельность по профилю образовательной программы от 29.08.2024; срок действия договора – 5 лет

<b>Помещение для организации практической подготовки обучающихся с перечнем основного оборудования</b> <b>Отдел поддержки компьютерных систем № 3:</b> письменный стол - 3 шт.; офисное кресло - 3 шт.; персональный компьютер - 3 шт.; монитор - 3 шт.; тумба - 6 шт.; стационарный телефон - 3 шт.; настольная лампа - 3 шт.; многофункциональное устройство (принтер, сканер, ксерокс) – 1 шт.; стеллаж - 5 шт.; шкаф - 4 шт.; вешалка - 1 шт.	170100, Тверская область, г. Тверь, ул. Новоторжская, д.3 (17,6 кв.м., 1 этаж, помещение № 4)	Практическая подготовка	<b>ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ «ТВЕРСКИЕ КРИПТОГРАФИЧЕСКИЕ СИСТЕМЫ»</b>	<b>Договор № 2-ПрИн</b> о практической подготовке обучающихся, заключенный между организацией, осуществляющей образовательную деятельность, и организацией, осуществляющей деятельность по профилю образовательной программы от 29.08.2024; срок действия договора – 5 лет
---	--	-------------------------	--	--

## 10. Образовательные технологии

Наименование образовательной технологии	Краткая характеристика
Дифференцированное обучение	Технология обучения, целью которой является создание оптимальных условий для выявления задатков, развития интересов и способностей обучающихся через разделение на группы, подразумевает наличие разных уровней учебных требований к группам в овладении ими содержанием образования.

## 11. Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья

Для осуществления процедур текущего контроля успеваемости и промежуточной аттестации обучающихся созданы фонды оценочных средств, адаптированные для инвалидов и лиц с ограниченными возможностями здоровья и позволяющие оценить достижение ими запланированных в основной образовательной программе результатов обучения и уровень сформированности всех компетенций, заявленных в образовательной программе. Форма проведения текущей аттестации для студентов-инвалидов устанавливается с учетом индивидуальных психофизических особенностей (устно,

письменно на бумаге, письменно на компьютере, в форме тестирования и т.п.). При тестировании для слабовидящих студентов используются фонды оценочных средств с укрупненным шрифтом. На экзамен приглашается сопровождающий, который обеспечивает техническое сопровождение студенту. При необходимости студенту-инвалиду предоставляется дополнительное время для подготовки ответа на экзамене (или зачете). Обучающиеся с ограниченными возможностями здоровья и обучающиеся инвалиды обеспечиваются печатными и электронными образовательными ресурсами (программы, учебники, учебные пособия материалы для самостоятельной работы и т.д.) в формах, адаптированных к ограничениям их здоровья и восприятия информации:

1) для инвалидов и лиц с ограниченными возможностями здоровья **по зрению:**

- **для слепых:** задания для выполнения на семинарах и практических занятиях оформляются рельефно-точечным шрифтом Брайля или в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением для слепых, либо зачитываются ассистентом; письменные задания выполняются на бумаге рельефно-точечным шрифтом Брайля или на компьютере со специализированным программным обеспечением для слепых либо надиктовываются ассистенту; обучающимся для выполнения задания при необходимости предоставляется комплект письменных принадлежностей и бумага для письма рельефно-точечным шрифтом Брайля, компьютер со специализированным программным обеспечением для слепых;
- **для слабовидящих:** обеспечивается индивидуальное равномерное освещение не менее 300 люкс; обучающимся для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств; задания для выполнения заданий оформляются увеличенным шрифтом;

2) для инвалидов и лиц с ограниченными возможностями здоровья **по слуху:**

- **для глухих и слабослышащих:** обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости обучающимся предоставляется звукоусиливающая аппаратура индивидуального пользования; предоставляются услуги сурдопереводчика;
- **для слепоглухих** допускается присутствие ассистента, оказывающего услуги тифлосурдопереводчика (помимо требований, выполняемых соответственно для слепых и глухих);

3) для лиц с тяжелыми нарушениями речи, глухих, слабослышащих лекции и семинары, проводимые в устной форме, проводятся в письменной форме;

4) для инвалидов и лиц с ограниченными возможностями здоровья, **имеющих нарушения опорно-двигательного аппарата:**

- для лиц с нарушениями опорно-двигательного аппарата, нарушениями двигательных функций верхних конечностей или отсутствием верхних конечностей: письменные задания выполняются на компьютере со специализированным программным обеспечением или надиктовываются ассистенту; выполнение заданий (тестов, контрольных работ), проводимые в письменной форме, проводятся в устной форме путем опроса, беседы с обучающимся.



**Частное учреждение высшего образования  
«Высшая школа предпринимательства (институт)»  
(ЧУВО «ВШП»)**

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ  
по дисциплине  
Б1.О.10 «Современные технологии разработки  
программного обеспечения»**

**Направление подготовки: 09.04.03 Прикладная информатика**

**Направленность (профиль) программы магистратуры  
«Прикладная информатика и информатизация»**



**Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы**

В результате освоения программы магистратуры обучающийся должен овладеть следующими результатами обучения по дисциплине (модулю):

Код	Результаты освоения ООП (Содержание компетенций)	Индикаторы достижения	Перечень планируемых результатов обучения по дисциплине
УК-4	Способен применять современные коммуникативные технологии, в том числе на иностранном(ых) языке(ах), для академического и профессионального взаимодействия	УК-4.1. Выбирает на государственном и иностранном (-ых) языках коммуникативно приемлемые стили делового общения	<p><b>Знать:</b> особенности корректного коммуникативного поведения; стили делового общения.</p> <p>П.П1 П.П2 П.П3 П.П4 П.П5 П.П6 П.П7 П.П8 П.П9 П.П10 П.П11 П.П12 П.П13 П.П14 П.П15 П.П16 П.П17 П.П18 П.П19 П.П20 П.П21 П.П22 П.П23 П.П24 П.П25 П.П26 П.П27 П.П28 П.П29 П.П30 П.П31 П.П32 П.ТВ1 П.ТВ2 П.ТВ3</p>

			П.ТВ4
			П.ТВ5
			П.ТВ6
			П.ТВ7
			П.ТВ8
			П.ТВ9
			П.ТВ10
			П.ТВ11
			П.ТВ12
			П.ТВ13
			П.ТВ14
			П.ТВ15
			П.ТВ16
			П.ТВ17
			П.ТВ18
			П.ТВ19
			П.ТВ20
			П.ТВ21
			П.ТВ22
			П.ТВ23
			П.ТВ24
			П.ТВ25
			П.ТВ26
			П.ТВ27
			П.ТВ28
			П.ТВ29
			П.ТВ30
			П.ТВ31
			П.ТВ32
			П.ТВ33
			П.ТВ34
			П.ТВ35
			П.ТВ36
			П.ТВ37
			П.ТВ38
			П.ТВ39
			П.ТВ40
			П.ТВ41
			П.ТВ42
			П.ТВ43
			П.ТВ44
			П.ТВ45
			П.ТВ46
			П.ТВ47
			П.ТВ48

			П.ТВ49 П.ТВ50 Т.Д1_1 Т.Д2_1 Т.Д3_1 Т.Д4_1 Т.Д5_1 Т.Д6_1 Т.Д7_1 Т.Д8_1 Т.Д9_1 Т.Д10_1 Т.Д11_1 Т.Д12_1 Т.Д13_1 Т.Д14_1 Т.КС1_1 Т.КС2_1 Т.КС3_1 Т.КС4_1 Т.КС5_1 Т.КС6_1 Т.КС7_1 Т.КС8_1 Т.КС9_1 Т.КС10_1  Т.КС11_1 Т.КС12_1  Т.КС13_1  Т.КС14_1  Т.КС15_1  Т.КС16_1  Т.КС17_1  Т.КС18_1  Т.КС19_1  Т.КС20_1  Т.КС21_1
--	--	--	--

			T.KC22_1
			T.KC23_1
			T.KC24_1
			T.KC25_1
			T.KC26_1
			T.KC27_1
			T.KC28_1
			T.KC29_1
			T.KC30_1
			T.KC31_1
			T.KC32_1
			T.KC33_1
			T.KC34_1
			T.KC35_1
			T.KC36_1
			T.KC37_1
			T.KC38_1
			T.KC39_1
			T.KC40_1
			T.KC41_1
			T.KC42_1
			T.KC43_1
			T.KC44_1
		<b>Уметь:</b> использовать различные виды устной и	П.П1

			<p>письменной речи в деловом общении; применять коммуникативно приемлемые стили делового общения на государственном и иностранном (-ых) языках</p>	<p>П.П2 П.П3 П.П4 П.П5 П.П6 П.П7 П.П8 П.П9 П.П10 П.П11 П.П12 П.П13 П.П14 П.П15 П.П16 П.П17 П.П18 П.П19 П.П20 П.П21 П.П22 П.П23 П.П24 П.П25 П.П26 П.П27 П.П28 П.П29 П.П30 П.П31 П.П32 П.ТВ1 П.ТВ2 П.ТВ3 П.ТВ4 П.ТВ5 П.ТВ6 П.ТВ7 П.ТВ8 П.ТВ9 П.ТВ10 П.ТВ11 П.ТВ12 П.ТВ13 П.ТВ14</p>
--	--	--	--	---

			П.ТВ15
			П.ТВ16
			П.ТВ17
			П.ТВ18
			П.ТВ19
			П.ТВ20
			П.ТВ21
			П.ТВ22
			П.ТВ23
			П.ТВ24
			П.ТВ25
			П.ТВ26
			П.ТВ27
			П.ТВ28
			П.ТВ29
			П.ТВ30
			П.ТВ31
			П.ТВ32
			П.ТВ33
			П.ТВ34
			П.ТВ35
			П.ТВ36
			П.ТВ37
			П.ТВ38
			П.ТВ39
			П.ТВ40
			П.ТВ41
			П.ТВ42
			П.ТВ43
			П.ТВ44
			П.ТВ45
			П.ТВ46
			П.ТВ47
			П.ТВ48
			П.ТВ49
			П.ТВ50
			Т.Д1_1
			Т.Д2_1
			Т.Д3_1
			Т.Д4_1
			Т.Д5_1
			Т.Д6_1
			Т.Д7_1
			Т.Д8_1
			Т.Д9_1



			<p>T.KC28_1</p> <p>T.KC29_1</p> <p>T.KC30_1</p> <p>T.KC31_1</p> <p>T.KC32_1</p> <p>T.KC33_1</p> <p>T.KC34_1</p> <p>T.KC35_1</p> <p>T.KC36_1</p> <p>T.KC37_1</p> <p>T.KC38_1</p> <p>T.KC39_1</p> <p>T.KC40_1</p> <p>T.KC41_1</p> <p>T.KC42_1</p> <p>T.KC43_1</p> <p>T.KC44_1</p>
		<p>УК-4.2. Использует информационно-коммуникационные технологии при поиске необходимой информации в процессе решения различных коммуникативных задач на государственном и иностранном (-ых) языках.</p>	<p><b>Знать:</b> функциональные стили, особенности каждого из них, условия и цели их использования в устной и письменной коммуникации</p> <p>П.П1</p> <p>П.П2</p> <p>П.П3</p> <p>П.П4</p> <p>П.П5</p> <p>П.П6</p> <p>П.П7</p> <p>П.П8</p> <p>П.П9</p> <p>П.П10</p> <p>П.П11</p>



			П.П12
			П.П13
			П.П14
			П.П15
			П.П16
			П.П17
			П.П18
			П.П19
			П.П20
			П.П21
			П.П22
			П.П23
			П.П24
			П.П25
			П.П26
			П.П27
			П.П28
			П.П29
			П.П30
			П.П31
			П.П32
			П.ТВ1
			П.ТВ2
			П.ТВ3
			П.ТВ4
			П.ТВ5
			П.ТВ6
			П.ТВ7
			П.ТВ8
			П.ТВ9
			П.ТВ10
			П.ТВ11
			П.ТВ12
			П.ТВ13
			П.ТВ14
			П.ТВ15
			П.ТВ16
			П.ТВ17
			П.ТВ18
			П.ТВ19
			П.ТВ20
			П.ТВ21
			П.ТВ22
			П.ТВ23
			П.ТВ24

			П.ТВ25
			П.ТВ26
			П.ТВ27
			П.ТВ28
			П.ТВ29
			П.ТВ30
			П.ТВ31
			П.ТВ32
			П.ТВ33
			П.ТВ34
			П.ТВ35
			П.ТВ36
			П.ТВ37
			П.ТВ38
			П.ТВ39
			П.ТВ40
			П.ТВ41
			П.ТВ42
			П.ТВ43
			П.ТВ44
			П.ТВ45
			П.ТВ46
			П.ТВ47
			П.ТВ48
			П.ТВ49
			П.ТВ50
			Т.Д1_1
			Т.Д2_1
			Т.Д3_1
			Т.Д4_1
			Т.Д5_1
			Т.Д6_1
			Т.Д7_1
			Т.Д8_1
			Т.Д9_1
			Т.Д10_1
			Т.Д11_1
			Т.Д12_1
			Т.Д13_1
			Т.Д14_1
			Т.КС1_1
			Т.КС2_1
			Т.КС3_1
			Т.КС4_1
			Т.КС5_1

				T.KC6_1
				T.KC7_1
				T.KC8_1
				T.KC9_1
				T.KC10_1
				T.KC11_1
				T.KC12_1
				T.KC13_1
				T.KC14_1
				T.KC15_1
				T.KC16_1
				T.KC17_1
				T.KC18_1
				T.KC19_1
				T.KC20_1
				T.KC21_1
				T.KC22_1
				T.KC23_1
				T.KC24_1
				T.KC25_1
				T.KC26_1
				T.KC27_1
				T.KC28_1
				T.KC29_1
				T.KC30_1
				T.KC31_1
				T.KC32_1

			T.KC33_1
			T.KC34_1
			T.KC35_1
			T.KC36_1
			T.KC37_1
			T.KC38_1
			T.KC39_1
			T.KC40_1
			T.KC41_1
			T.KC42_1
			T.KC43_1
			T.KC44_1
		<p><b>Уметь:</b> коммуникативно и культурно приемлемо вести деловые разговоры в рамках академического и профессионального взаимодействия на государственном и иностранном (-ых) языках.</p>	<p>П.П1</p> <p>П.П2</p> <p>П.П3</p> <p>П.П4</p> <p>П.П5</p> <p>П.П6</p> <p>П.П7</p> <p>П.П8</p> <p>П.П9</p> <p>П.П10</p> <p>П.П11</p> <p>П.П12</p> <p>П.П13</p> <p>П.П14</p> <p>П.П15</p> <p>П.П16</p> <p>П.П17</p> <p>П.П18</p> <p>П.П19</p> <p>П.П20</p> <p>П.П21</p> <p>П.П22</p>

			П.П23
			П.П24
			П.П25
			П.П26
			П.П27
			П.П28
			П.П29
			П.П30
			П.П31
			П.П32
			П.ТВ1
			П.ТВ2
			П.ТВ3
			П.ТВ4
			П.ТВ5
			П.ТВ6
			П.ТВ7
			П.ТВ8
			П.ТВ9
			П.ТВ10
			П.ТВ11
			П.ТВ12
			П.ТВ13
			П.ТВ14
			П.ТВ15
			П.ТВ16
			П.ТВ17
			П.ТВ18
			П.ТВ19
			П.ТВ20
			П.ТВ21
			П.ТВ22
			П.ТВ23
			П.ТВ24
			П.ТВ25
			П.ТВ26
			П.ТВ27
			П.ТВ28
			П.ТВ29
			П.ТВ30
			П.ТВ31
			П.ТВ32
			П.ТВ33
			П.ТВ34
			П.ТВ35

			П.ТВ36 П.ТВ37 П.ТВ38 П.ТВ39 П.ТВ40 П.ТВ41 П.ТВ42 П.ТВ43 П.ТВ44 П.ТВ45 П.ТВ46 П.ТВ47 П.ТВ48 П.ТВ49 П.ТВ50 Т.Д1_1 Т.Д2_1 Т.Д3_1 Т.Д4_1 Т.Д5_1
		УК-4.3. Демонстрирует способность находить, воспринимать и использовать информацию на иностранном языке, полученную для решения коммуникативных задач в рамках академического и профессионального взаимодействия.	<b>Знать:</b> лексические, грамматические нормы современного литературного языка П.П1 П.П2 П.П3 П.П4 П.П5 П.П6 П.П7 П.П8 П.П9 П.П10 П.П11 П.П12 П.П13 П.П14 П.П15 П.П16 П.П17 П.П18 П.П19 П.П20 П.П21 П.П22 П.П23

			П.П24
			П.П25
			П.П26
			П.П27
			П.П28
			П.П29
			П.П30
			П.П31
			П.П32
			П.ТВ1
			П.ТВ2
			П.ТВ3
			П.ТВ4
			П.ТВ5
			П.ТВ6
			П.ТВ7
			П.ТВ8
			П.ТВ9
			П.ТВ10
			П.ТВ11
			П.ТВ12
			П.ТВ13
			П.ТВ14
			П.ТВ15
			П.ТВ16
			П.ТВ17
			П.ТВ18
			П.ТВ19
			П.ТВ20
			П.ТВ21
			П.ТВ22
			П.ТВ23
			П.ТВ24
			П.ТВ25
			П.ТВ26
			П.ТВ27
			П.ТВ28
			П.ТВ29
			П.ТВ30
			П.ТВ31
			П.ТВ32
			П.ТВ33
			П.ТВ34
			П.ТВ35
			П.ТВ36

			П.ТВ37
			П.ТВ38
			П.ТВ39
			П.ТВ40
			П.ТВ41
			П.ТВ42
			П.ТВ43
			П.ТВ44
			П.ТВ45
			П.ТВ46
			П.ТВ47
			П.ТВ48
			П.ТВ49
			П.ТВ50
			Т.Д1_1
			Т.Д2_1
			Т.Д3_1
			Т.Д4_1
			Т.Д5_1
			Т.Д6_1
			Т.Д7_1
			Т.Д8_1
			Т.Д9_1
			Т.Д10_1
			Т.Д11_1
			Т.Д12_1
			Т.Д13_1
			Т.Д14_1
			Т.КС1_1
			Т.КС2_1
			Т.КС3_1
			Т.КС4_1
			Т.КС5_1
			Т.КС6_1
			Т.КС7_1
			Т.КС8_1
			Т.КС9_1
			Т.КС10_1
			Т.КС11_1
			Т.КС12_1
			Т.КС13_1
			Т.КС14_1
			Т.КС15_1
			Т.КС16_1
			Т.КС17_1



			T.KC18_1 T.KC19_1 T.KC20_1 T.KC21_1 T.KC22_1 T.KC23_1 T.KC24_1 T.KC25_1 T.KC26_1 T.KC27_1 T.KC28_1 T.KC29_1 T.KC30_1 T.KC31_1 T.KC32_1 T.KC33_1 T.KC34_1 T.KC35_1 T.KC36_1 T.KC37_1 T.KC38_1 T.KC39_1 T.KC40_1 T.KC41_1 T.KC42_1 T.KC43_1 T.KC44_1
			<p><b>Уметь:</b> вести диалог и создавать монолог в соответствии с целями и условиями деловой коммуникации</p> П.П1 П.П2 П.П3 П.П4 П.П5 П.П6 П.П7 П.П8 П.П9 П.П10 П.П11 П.П12 П.П13 П.П14 П.П15 П.П16 П.П17

			П.П18
			П.П19
			П.П20
			П.П21
			П.П22
			П.П23
			П.П24
			П.П25
			П.П26
			П.П27
			П.П28
			П.П29
			П.П30
			П.П31
			П.П32
			П.ТВ1
			П.ТВ2
			П.ТВ3
			П.ТВ4
			П.ТВ5
			П.ТВ6
			П.ТВ7
			П.ТВ8
			П.ТВ9
			П.ТВ10
			П.ТВ11
			П.ТВ12
			П.ТВ13
			П.ТВ14
			П.ТВ15
			П.ТВ16
			П.ТВ17
			П.ТВ18
			П.ТВ19
			П.ТВ20
			П.ТВ21
			П.ТВ22
			П.ТВ23
			П.ТВ24
			П.ТВ25
			П.ТВ26
			П.ТВ27
			П.ТВ28
			П.ТВ29
			П.ТВ30

			П.ТВ31 П.ТВ32 П.ТВ33 П.ТВ34 П.ТВ35 П.ТВ36 П.ТВ37 П.ТВ38 П.ТВ39 П.ТВ40 П.ТВ41 П.ТВ42 П.ТВ43 П.ТВ44 П.ТВ45 П.ТВ46 П.ТВ47 П.ТВ48 П.ТВ49 П.ТВ50 Т.КС22_1
ОПК -2	Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач;	ОПК-2.1 Обладает фундаментальными знаниями по программированию и языкам программирования, организации баз данных, системного программирования и компьютерного моделирования.	<b>Знать:</b> методы и технологии применения объектно-ориентированного подхода для реализации программного кода; модели баз данных; методами компьютерного моделирования при решении профессиональных задач. П.П1 П.П2 П.П3 П.П4 П.П5 П.П6 П.П7 П.П8 П.П9 П.П10 П.П11 П.П12 П.П13 П.П14 П.П15 П.П16 П.П17 П.П18 П.П19 П.П20 П.П21 П.П22

			П.П23
			П.П24
			П.П25
			П.П26
			П.П27
			П.П28
			П.П29
			П.П30
			П.П31
			П.П32
			П.ТВ1
			П.ТВ2
			П.ТВ3
			П.ТВ4
			П.ТВ5
			П.ТВ6
			П.ТВ7
			П.ТВ8
			П.ТВ9
			П.ТВ10
			П.ТВ11
			П.ТВ12
			П.ТВ13
			П.ТВ14
			П.ТВ15
			П.ТВ16
			П.ТВ17
			П.ТВ18
			П.ТВ19
			П.ТВ20
			П.ТВ21
			П.ТВ22
			П.ТВ23
			П.ТВ24
			П.ТВ25
			П.ТВ26
			П.ТВ27
			П.ТВ28
			П.ТВ29
			П.ТВ30
			П.ТВ31
			П.ТВ32
			П.ТВ33
			П.ТВ34
			П.ТВ35

			П.ТВ36 П.ТВ37 П.ТВ38 П.ТВ39 П.ТВ40 П.ТВ41 П.ТВ42 П.ТВ43 П.ТВ44 П.ТВ45 П.ТВ46 П.ТВ47 П.ТВ48 П.ТВ49 П.ТВ50 Т.Д1_1 Т.Д2_1 Т.Д3_1 Т.Д4_1 Т.Д5_1 Т.Д6_1 Т.Д7_1 Т.Д8_1 Т.Д9_1 Т.Д10_1 Т.Д11_1 Т.Д12_1 Т.Д13_1 Т.Д14_1 Т.КС1_1 Т.КС2_1 Т.КС3_1 Т.КС4_1 Т.КС5_1 Т.КС6_1 Т.КС7_1 Т.КС8_1 Т.КС9_1 Т.КС10_1  Т.КС11_1  Т.КС12_1  Т.КС13_1
--	--	--	--

			T.KC14_1
			T.KC15_1
			T.KC16_1
			T.KC17_1
			T.KC18_1
			T.KC19_1
			T.KC20_1
			T.KC21_1
			T.KC22_1
			T.KC23_1
			T.KC24_1
			T.KC25_1
			T.KC26_1
			T.KC27_1
			T.KC28_1
			T.KC29_1
			T.KC30_1
			T.KC31_1
			T.KC32_1
			T.KC33_1
			T.KC34_1
			T.KC35_1
			T.KC36_1
			T.KC37_1
			T.KC38_1

			<p>T.KC39_1</p> <p>T.KC40_1</p> <p>T.KC41_1</p> <p>T.KC42_1</p> <p>T.KC43_1</p> <p>T.KC44_1</p>
		<p>ОПК-2.2 Демонстрирует умение обосновывать выбор современных интеллектуальных технологий и программной среды при разработке оригинальных программных средств для решения профессиональных задач</p>	<p><b>Знать:</b> современные интеллектуальные технологии для решения профессиональных задач.</p> <p>П.П1</p> <p>П.П2</p> <p>П.П3</p> <p>П.П4</p> <p>П.П5</p> <p>П.П6</p> <p>П.П7</p> <p>П.П8</p> <p>П.П9</p> <p>П.П10</p> <p>П.П11</p> <p>П.П12</p> <p>П.П13</p> <p>П.П14</p> <p>П.П15</p> <p>П.П16</p> <p>П.П17</p> <p>П.П18</p> <p>П.П19</p> <p>П.П20</p> <p>П.П21</p> <p>П.П22</p> <p>П.П23</p> <p>П.П24</p> <p>П.П25</p> <p>П.П26</p> <p>П.П27</p> <p>П.П28</p> <p>П.П29</p> <p>П.П30</p> <p>П.П31</p> <p>П.П32</p>

			П.ТВ1
			П.ТВ2
			П.ТВ3
			П.ТВ4
			П.ТВ5
			П.ТВ6
			П.ТВ7
			П.ТВ8
			П.ТВ9
			П.ТВ10
			П.ТВ11
			П.ТВ12
			П.ТВ13
			П.ТВ14
			П.ТВ15
			П.ТВ16
			П.ТВ17
			П.ТВ18
			П.ТВ19
			П.ТВ20
			П.ТВ21
			П.ТВ22
			П.ТВ23
			П.ТВ24
			П.ТВ25
			П.ТВ26
			П.ТВ27
			П.ТВ28
			П.ТВ29
			П.ТВ30
			П.ТВ31
			П.ТВ32
			П.ТВ33
			П.ТВ34
			П.ТВ35
			П.ТВ36
			П.ТВ37
			П.ТВ38
			П.ТВ39
			П.ТВ40
			П.ТВ41
			П.ТВ42
			П.ТВ43
			П.ТВ44
			П.ТВ45



			П.ТВ46
			П.ТВ47
			П.ТВ48
			П.ТВ49
			П.ТВ50
			Т.Д1_1
			Т.Д2_1
			Т.Д3_1
			Т.Д4_1
			Т.Д5_1
			Т.Д6_1
			Т.Д7_1
			Т.Д8_1
			Т.Д9_1
			Т.Д10_1
			Т.Д11_1
			Т.Д12_1
			Т.Д13_1
			Т.Д14_1
			Т.КС1_1
			Т.КС2_1
			Т.КС3_1
			Т.КС4_1
			Т.КС5_1
			Т.КС6_1
			Т.КС7_1
			Т.КС8_1
			Т.КС9_1
			Т.КС10_1
			Т.КС11_1
			Т.КС12_1
			Т.КС13_1
			Т.КС14_1
			Т.КС15_1
			Т.КС16_1
			Т.КС17_1
			Т.КС18_1
			Т.КС19_1

			T.KC20_1
			T.KC21_1
			T.KC22_1
			T.KC23_1
			T.KC24_1
			T.KC25_1
			T.KC26_1
			T.KC27_1
			T.KC28_1
			T.KC29_1
			T.KC30_1
			T.KC31_1
			T.KC32_1
			T.KC33_1
			T.KC34_1
			T.KC35_1
			T.KC36_1
			T.KC37_1
			T.KC38_1
			T.KC39_1
			T.KC40_1
			T.KC41_1
			T.KC42_1
			T.KC43_1
			T.KC44_1

			T.KC45_1
			T.KC46_1
			T.KC47_1
			T.KC48_1
			T.KC49_1
			T.T1_1
			T.T2_1
			T.T3_1
			T.T4_1
			T.T5_1
			T.T6_1
			T.T7_1
			T.T8_1
			T.T9_1
			T.T10_1
			T.T11_1
			T.T12_1
			T.T13_1
			T.T14_1
			T.T15_1
			T.T16_1
			T.T17_1
			T.T18_1
			T.T19_1
			T.T20_1
			T.Y1_1
			T.Y2_1
			T.Y3_1
			T.Y4_1
			T.Y5_1
			T.Y6_1
			T.Y7_1
			T.Y8_1
			T.Y9_1
			T.Y10_1
			T.Y11_1
			T.Y12_1
			T.Y13_1
			T.Y14_1
			T.Y15_1
			T.Y16_1

			T.Y17_1
			T.Y18_1
			T.Y19_1
			T.Y20_1
			T.Y21_1
			T.Y22_1
			T.Y23_1
			T.Y24_1
			T.Y25_1
			T.Y26_1
			T.Y27_1
			T.Y28_1
			T.Y29_1
			T.Y30_1
			T.Y31_1
			T.Y32_1
			T.Y33_1
			T.Y34_1
			T.Y35_1
			T.Y36_1
			T.Y37_1
			T.Y38_1
			T.Y39_1
			T.Y40_1
			T.Y41_1
			T.Y42_1
			T.Y43_1
			T.Y44_1
			T.Д1_2
			T.KC1_2
			T.KC2_2
			T.KC3_2
			T.KC4_2
			T.KC5_2
			T.KC6_2
			T.KC7_2
			T.KC8_2
			T.KC9_2
			T.KC10_2
			T.KC11_2
			T.KC12_2
			T.KC13_2
			T.KC14_2

			T.KC15_2
			T.KC16_2
			T.KC17_2
			T.KC18_2
			T.KC19_2
			T.KC20_2
			T.KC21_2
			T.KC22_2
			T.KC23_2
			T.KC24_2
			T.T1_2
			T.T2_2
			T.T3_2
			T.T4_2
			T.T5_2
			T.T6_2
			T.T7_2
			T.T8_2
			T.T9_2
			T.T10_2
			T.T11_2
			T.T12_2
			T.T13_2
			T.T14_2
			T.T15_2
			T.T16_2
			T.T17_2
			T.T18_2
			T.T19_2
			T.T20_2
			T.T21_2
			T.Y1_2
			T.Y2_2
			T.Y3_2
			T.Y4_2
			<b>Уметь:</b> обосновывать выбор

			современных	П.П1
			интеллектуальных	П.П2
			технологий и программной	П.П3
			среды при разработке	П.П4
			оригинальных программных	П.П5
			средств для решения	П.П6
			профессиональных задач.	П.П7
				П.П8
				П.П9
				П.П10
				П.П11
				П.П12
				П.П13
				П.П14
				П.П15
				П.П16
				П.П17
				П.П18
				П.П19
				П.П20
				П.П21
				П.П22
				П.П23
				П.П24
				П.П25
				П.П26
				П.П27
				П.П28
				П.П29
				П.П30
				П.П31
				П.П32
				П.ТВ1
				П.ТВ2
				П.ТВ3
				П.ТВ4
				П.ТВ5
				П.ТВ6
				П.ТВ7
				П.ТВ8
				П.ТВ9
				П.ТВ10
				П.ТВ11
				П.ТВ12
				П.ТВ13

			П.ТВ14
			П.ТВ15
			П.ТВ16
			П.ТВ17
			П.ТВ18
			П.ТВ19
			П.ТВ20
			П.ТВ21
			П.ТВ22
			П.ТВ23
			П.ТВ24
			П.ТВ25
			П.ТВ26
			П.ТВ27
			П.ТВ28
			П.ТВ29
			П.ТВ30
			П.ТВ31
			П.ТВ32
			П.ТВ33
			П.ТВ34
			П.ТВ36
			П.ТВ37
			П.ТВ38
			П.ТВ39
			П.ТВ40
			П.ТВ41
			П.ТВ42
			П.ТВ43
			П.ТВ44
			П.ТВ45
			П.ТВ46
			П.ТВ47
			П.ТВ48
			П.ТВ49
			П.ТВ50
			Т.Д1_1
			Т.Д2_1
			Т.Д3_1
			Т.Д4_1
			Т.Д5_1
			Т.Д6_1
			Т.Д7_1
			Т.Д8_1
			Т.Д9_1

			T.Д10_1 T.Д11_1 T.Д12_1 T.Д13_1 T.Д14_1 T.KC1_1 T.KC2_1 T.KC3_1 T.KC4_1 T.KC5_1 T.KC6_1 T.KC7_1 T.KC8_1 T.KC9_1 T.KC10_1  T.KC11_1 T.KC12_1  T.KC13_1  T.KC14_1  T.KC15_1  T.KC16_1  T.KC17_1  T.KC18_1  T.KC19_1  T.KC20_1  T.KC21_1  T.KC22_1  T.KC23_1  T.KC24_1  T.KC25_1  T.KC26_1  T.KC27_1
--	--	--	--



			T.KC28_1
			T.KC29_1
			T.KC30_1
			T.KC31_1
			T.KC32_1
			T.KC33_1
			T.KC34_1
			T.KC35_1
			T.KC36_1
			T.KC37_1
			T.KC38_1
			T.KC39_1
			T.KC40_1
			T.KC41_1
			T.KC42_1
			T.KC43_1
			T.KC44_1
			T.KC45_1
			T.KC46_1
			T.KC47_1
			T.KC48_1
			T.KC49_1
			T.T1_1
			T.T2_1
			T.T3_1
			T.T4_1

			T.T5_1
			T.T6_1
			T.T7_1
			T.T8_1
			T.T9_1
			T.T10_1
			T.T11_1
			T.T12_1
			T.T13_1
			T.T14_1
			T.T15_1
			T.T16_1
			T.T17_1
			T.T18_1
			T.T19_1
			T.T20_1
			T.Д1_2
			T.KC1_2
			T.KC2_2
			T.KC3_2
			T.KC4_2
			T.KC5_2
			T.KC6_2
			T.KC7_2
			T.KC8_2
			T.KC9_2
			T.KC10_2
			T.KC11_2
			T.KC12_2
			T.KC13_2
			T.KC14_2
			T.KC15_2
			T.KC16_2
			T.KC17_2
			T.KC18_2
			T.KC19_2
			T.KC20_2

			<p>T.KC21_2</p> <p>T.KC22_2</p> <p>T.KC23_2</p> <p>T.KC24_2</p> <p>T.T1_2</p> <p>T.T2_2</p> <p>T.T3_2</p> <p>T.T4_2</p> <p>T.T5_2</p> <p>T.T6_2</p> <p>T.T7_2</p> <p>T.T8_2</p> <p>T.T9_2</p> <p>T.T10_2</p> <p>T.T11_2</p> <p>T.T12_2</p> <p>T.T13_2</p> <p>T.T14_2</p> <p>T.T15_2</p> <p>T.T16_2</p> <p>T.T17_2</p> <p>T.T18_2</p> <p>T.T19_2</p> <p>T.T20_2</p> <p>T.T21_2</p>
ОПК -5	Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем;	ОПК-5.1 Демонстрирует навыки разработки программ для создания приложений и баз данных ИС	<p><b>Знать:</b> современный уровень развития алгоритмических и программных решений в области разработки программного обеспечения; структуру и типы баз данных.</p> <p>П.П1</p> <p>П.П2</p> <p>П.П3</p> <p>П.П4</p> <p>П.П5</p> <p>П.П6</p> <p>П.П7</p> <p>П.П8</p> <p>П.П9</p> <p>П.П10</p> <p>П.П11</p> <p>П.П12</p> <p>П.П13</p> <p>П.П14</p>

			П.П15
			П.П16
			П.П17
			П.П18
			П.П19
			П.П20
			П.П21
			П.П22
			П.П23
			П.П24
			П.П25
			П.П26
			П.П27
			П.П28
			П.П29
			П.П30
			П.П31
			П.П32
			П.ТВ1
			П.ТВ2
			П.ТВ3
			П.ТВ4
			П.ТВ5
			П.ТВ6
			П.ТВ7
			П.ТВ8
			П.ТВ9
			П.ТВ10
			П.ТВ11
			П.ТВ12
			П.ТВ13
			П.ТВ14
			П.ТВ15
			П.ТВ16
			П.ТВ17
			П.ТВ18
			П.ТВ19
			П.ТВ20
			П.ТВ21
			П.ТВ22
			П.ТВ23
			П.ТВ24
			П.ТВ25
			П.ТВ26
			П.ТВ27

			П.ТВ28
			П.ТВ29
			П.ТВ30
			П.ТВ31
			П.ТВ32
			П.ТВ33
			П.ТВ34
			П.ТВ35
			П.ТВ36
			П.ТВ37
			П.ТВ38
			П.ТВ39
			П.ТВ40
			П.ТВ41
			П.ТВ42
			П.ТВ43
			П.ТВ44
			П.ТВ45
			П.ТВ46
			П.ТВ47
			П.ТВ48
			П.ТВ49
			П.ТВ50
			Т.Д1_1
			Т.Д2_1
			Т.Д3_1
			Т.Д4_1
			Т.Д5_1
			Т.Д6_1
			Т.Д7_1
			Т.Д8_1
			Т.Д9_1
			Т.Д10_1
			Т.Д11_1
			Т.Д12_1
			Т.Д13_1
			Т.Д14_1
			Т.КС1_1
			Т.КС2_1
			Т.КС3_1
			Т.КС4_1
			Т.КС5_1
			Т.КС6_1
			Т.КС7_1
			Т.КС8_1

			T.KC9_1 T.KC10_1
			T.KC11_1 T.KC12_1
			T.KC13_1
			T.KC14_1
			T.KC15_1
			T.KC16_1
			T.KC17_1
			T.KC18_1
			T.KC19_1
			T.KC20_1
			T.KC21_1
			T.KC22_1
			T.KC23_1
			T.KC24_1
			T.KC25_1
			T.KC26_1
			T.KC27_1
			T.KC28_1
			T.KC29_1
			T.KC30_1
			T.KC31_1
			T.KC32_1
			T.KC33_1
			T.KC34_1

			T.KC35_1
			T.KC36_1
			T.KC37_1
			T.KC38_1
			T.KC39_1
			T.KC40_1
			T.KC41_1
			T.KC42_1
			T.KC43_1
			T.KC44_1
		<p><b>Уметь:</b> строить модели баз данных, проектировать модули информационных систем.</p>	<p>П.П1</p> <p>П.П2</p> <p>П.П3</p> <p>П.П4</p> <p>П.П5</p> <p>П.П6</p> <p>П.П7</p> <p>П.П8</p> <p>П.П9</p> <p>П.П10</p> <p>П.П11</p> <p>П.П12</p> <p>П.П13</p> <p>П.П14</p> <p>П.П15</p> <p>П.П16</p> <p>П.П17</p> <p>П.П18</p> <p>П.П19</p> <p>П.П20</p> <p>П.П21</p> <p>П.П22</p> <p>П.П23</p> <p>П.П24</p> <p>П.П25</p>

			П.П26
			П.П27
			П.П28
			П.П29
			П.П30
			П.П31
			П.П32
			П.ТВ1
			П.ТВ2
			П.ТВ3
			П.ТВ4
			П.ТВ5
			П.ТВ6
			П.ТВ7
			П.ТВ8
			П.ТВ9
			П.ТВ10
			П.ТВ11
			П.ТВ12
			П.ТВ13
			П.ТВ14
			П.ТВ15
			П.ТВ16
			П.ТВ17
			П.ТВ18
			П.ТВ19
			П.ТВ20
			П.ТВ21
			П.ТВ22
			П.ТВ23
			П.ТВ24
			П.ТВ25
			П.ТВ26
			П.ТВ27
			П.ТВ28
			П.ТВ29
			П.ТВ30
			П.ТВ31
			П.ТВ32
			П.ТВ33
			П.ТВ34
			П.ТВ36
			П.ТВ37
			П.ТВ38
			П.ТВ39



			П.ТВ40 П.ТВ41 П.ТВ42 П.ТВ43 П.ТВ44 П.ТВ45 П.ТВ46 П.ТВ47 П.ТВ48 П.ТВ49 П.ТВ50
		ОПК-5.2 Демонстрирует навыки внедрения и модернизации программного и аппаратного обеспечения информационных и автоматизированных систем	<b>Знать:</b> современное программное и аппаратное обеспечение информационных и автоматизированных систем. П.П1 П.П2 П.П3 П.П4 П.П5 П.П6 П.П7 П.П8 П.П9 П.П10 П.П11 П.П12 П.П13 П.П14 П.П15 П.П16 П.П17 П.П18 П.П19 П.П20 П.П21 П.П22 П.П23 П.П24 П.П25 П.П26 П.П27 П.П28 П.П29 П.П30 П.П31 П.П32

			П.ТВ1
			П.ТВ2
			П.ТВ3
			П.ТВ4
			П.ТВ5
			П.ТВ6
			П.ТВ7
			П.ТВ8
			П.ТВ9
			П.ТВ10
			П.ТВ11
			П.ТВ12
			П.ТВ13
			П.ТВ14
			П.ТВ15
			П.ТВ16
			П.ТВ17
			П.ТВ18
			П.ТВ19
			П.ТВ20
			П.ТВ21
			П.ТВ22
			П.ТВ23
			П.ТВ24
			П.ТВ25
			П.ТВ26
			П.ТВ27
			П.ТВ28
			П.ТВ29
			П.ТВ30
			П.ТВ31
			П.ТВ32
			П.ТВ33
			П.ТВ34
			П.ТВ35
			П.ТВ36
			П.ТВ37
			П.ТВ38
			П.ТВ39
			П.ТВ40
			П.ТВ41
			П.ТВ42
			П.ТВ43
			П.ТВ44
			П.ТВ45

			П.ТВ46
			П.ТВ47
			П.ТВ48
			П.ТВ49
			П.ТВ50
			Т.Д1_1
			Т.Д2_1
			Т.Д3_1
			Т.Д4_1
			Т.Д5_1
			Т.Д6_1
			Т.Д7_1
			Т.Д8_1
			Т.Д9_1
			Т.Д10_1
			Т.Д11_1
			Т.Д12_1
			Т.Д13_1
			Т.Д14_1
			Т.КС1_1
			Т.КС2_1
			Т.КС3_1
			Т.КС4_1
			Т.КС5_1
			Т.КС6_1
			Т.КС7_1
			Т.КС8_1
			Т.КС9_1
			Т.КС10_1
			Т.КС11_1
			Т.КС12_1
			Т.КС13_1
			Т.КС14_1
			Т.КС15_1
			Т.КС16_1
			Т.КС17_1
			Т.КС18_1
			Т.КС19_1

			T.KC20_1
			T.KC21_1
			T.KC22_1
			T.KC23_1
			T.KC24_1
			T.KC25_1
			T.KC26_1
			T.KC27_1
			T.KC28_1
			T.KC29_1
			T.KC30_1
			T.KC31_1
			T.KC32_1
			T.KC33_1
			T.KC34_1
			T.KC35_1
			T.KC36_1
			T.KC37_1
			T.KC38_1
			T.KC39_1
			T.KC40_1
			T.KC41_1
			T.KC42_1
			T.KC43_1
			T.KC44_1

			T.KC45_1
			T.KC46_1
			T.KC47_1
			T.KC48_1
			T.KC49_1
			T.T1_1
			T.T2_1
			T.T3_1
			T.T4_1
			T.T5_1
			T.T6_1
			T.T7_1
			T.T8_1
			T.T9_1
			T.T10_1
			T.T11_1
			T.T12_1
			T.T13_1
			T.T14_1
			T.T15_1
			T.T16_1
			T.T17_1
			T.T18_1
			T.T19_1
			T.T20_1
			T.Y1_1
			T.Y2_1
			T.Y3_1
			T.Y4_1
			T.Y5_1
			T.Y6_1
			T.Y7_1
			T.Y8_1
			T.Y9_1
			T.Y10_1
			T.Y11_1
			T.Y12_1
			T.Y13_1
			T.Y14_1
			T.Y15_1
			T.Y16_1

			T.Y17_1
			T.Y18_1
			T.Y19_1
			T.Y20_1
			T.Y21_1
			T.Y22_1
			T.Y23_1
			T.Y24_1
			T.Y25_1
			T.Y26_1
			T.Y27_1
			T.Y28_1
			T.Y29_1
			T.Y30_1
			T.Y31_1
			T.Y32_1
			T.Y33_1
			T.Y34_1
			T.Y35_1
			T.Y36_1
			T.Y37_1
			T.Y38_1
			T.Y39_1
			T.Y40_1
			T.Y41_1
			T.Y42_1
			T.Y43_1
			T.Y44_1
			T.Д1_2
			T.KC1_2
			T.KC2_2
			T.KC3_2
			T.KC4_2
			T.KC5_2
			T.KC6_2
			T.KC7_2
			T.KC8_2
			T.KC9_2
			T.KC10_2
			T.KC11_2
			T.KC12_2
			T.KC13_2
			T.KC14_2

			T.KC15_2
			T.KC16_2
			T.KC17_2
			T.KC18_2
			T.KC19_2
			T.KC20_2
			T.KC21_2
			T.KC22_2
			T.KC23_2
			T.KC24_2
			T.T1_2
			T.T2_2
			T.T3_2
			T.T4_2
			T.T5_2
			T.T6_2
			T.T7_2
			T.T8_2
			T.T9_2
			T.T10_2
			T.T11_2
			T.T12_2
			T.T13_2
			T.T14_2
			T.T15_2
			T.T16_2
			T.T17_2
			T.T18_2
			T.T19_2
			T.T20_2
			T.T21_2
			T.Y1_2
			T.Y2_2
			T.Y3_2
			T.Y4_2
			<b>Уметь:</b> модернизировать

			программное и аппаратное обеспечение	П.П1
			информационных и автоматизированных систем	П.П2
			для решения профессиональных задач	П.П3
				П.П4
				П.П5
				П.П6
				П.П7
				П.П8
				П.П9
				П.П10
				П.П11
				П.П12
				П.П13
				П.П14
				П.П15
				П.П16
				П.П17
				П.П18
				П.П19
				П.П20
				П.П21
				П.П22
				П.П23
				П.П24
				П.П25
				П.П26
				П.П27
				П.П28
				П.П29
				П.П30
				П.П31
				П.П32
				П.ТВ2
				П.ТВ3
				П.ТВ5
				П.ТВ6
				П.ТВ7
				П.ТВ8
				П.ТВ9
				П.ТВ11
				П.ТВ12
				П.ТВ13
				П.ТВ14
				П.ТВ15
				П.ТВ16



			П.ТВ17
			П.ТВ18
			П.ТВ19
			П.ТВ20
			П.ТВ21
			П.ТВ22
			П.ТВ23
			П.ТВ24
			П.ТВ25
			П.ТВ26
			П.ТВ27
			П.ТВ28
			П.ТВ29
			П.ТВ30
			П.ТВ31
			П.ТВ32
			П.ТВ33
			П.ТВ34
			П.ТВ36
			П.ТВ37
			П.ТВ38
			П.ТВ39
			П.ТВ40
			П.ТВ41
			П.ТВ42
			П.ТВ43
			П.ТВ44
			П.ТВ46
			П.ТВ47
			П.ТВ48
			П.ТВ49
			П.ТВ50
			Т.КС1_1
			Т.КС2_1
			Т.КС3_1
			Т.КС4_1
			Т.КС5_1
			Т.КС6_1
			Т.КС7_1
			Т.КС8_1
			Т.КС9_1
			Т.КС10_1
			Т.КС11_1
			Т.КС12_1

			T.KC13_1
			T.KC14_1
			T.KC15_1
			T.KC16_1
			T.KC17_1
			T.KC18_1
			T.KC19_1
			T.KC20_1
			T.KC21_1
			T.KC22_1
			T.KC23_1
			T.KC24_1
			T.KC25_1
			T.KC26_1
			T.KC27_1
			T.KC28_1
			T.KC29_1
			T.KC30_1
			T.KC31_1
			T.KC32_1
			T.KC33_1
			T.KC34_1
			T.KC35_1
			T.KC36_1

			T.KC37_1
			T.KC38_1
			T.KC39_1
			T.KC40_1
			T.KC41_1
			T.KC42_1
			T.KC43_1
			T.KC44_1
			T.KC45_1
			T.KC46_1
			T.KC47_1
			T.KC48_1
			T.KC49_1
			T.KC1_2
			T.KC2_2
			T.KC3_2
			T.KC4_2
			T.KC5_2
			T.KC6_2
			T.KC7_2
			T.KC8_2
			T.KC9_2
			T.KC10_2
			T.KC11_2
			T.KC12_2
			T.KC13_2
			T.KC14_2
			T.KC15_2
			T.KC16_2
			T.KC17_2

			<p>T.KC18_2</p> <p>T.KC19_2</p> <p>T.KC20_2</p> <p>T.KC21_2</p> <p>T.KC22_2</p> <p>T.KC23_2</p> <p>T.KC24_2</p>
ОПК -8	Способен осуществлять эффективное управление разработкой программных средств и проектов.	ОПК-8.1. Демонстрирует знание методов организации сбора и изучения научно-технической информации по теме исследований и разработок при проектировании и реализации информационных систем.	<p><b>Знать:</b> архитектуру информационных систем предприятий и организаций; методологии и технологии реинжиниринга, проектирования и аудита прикладных информационных систем различных классов; инструментальные средства информационных систем и сервисов.</p> <p>П.П1</p> <p>П.П2</p> <p>П.П3</p> <p>П.П4</p> <p>П.П5</p> <p>П.П6</p> <p>П.П7</p> <p>П.П8</p> <p>П.П9</p> <p>П.П10</p> <p>П.П11</p> <p>П.П12</p> <p>П.П13</p> <p>П.П14</p> <p>П.П15</p> <p>П.П16</p> <p>П.П17</p> <p>П.П18</p> <p>П.П19</p> <p>П.П20</p> <p>П.П21</p> <p>П.П22</p> <p>П.П23</p> <p>П.П24</p> <p>П.П25</p> <p>П.П26</p> <p>П.П27</p> <p>П.П28</p> <p>П.П29</p>

			П.П30
			П.П31
			П.П32
			П.ТВ1
			П.ТВ2
			П.ТВ3
			П.ТВ4
			П.ТВ5
			П.ТВ6
			П.ТВ7
			П.ТВ8
			П.ТВ9
			П.ТВ10
			П.ТВ11
			П.ТВ12
			П.ТВ13
			П.ТВ14
			П.ТВ15
			П.ТВ16
			П.ТВ17
			П.ТВ18
			П.ТВ19
			П.ТВ20
			П.ТВ21
			П.ТВ22
			П.ТВ23
			П.ТВ24
			П.ТВ25
			П.ТВ26
			П.ТВ27
			П.ТВ28
			П.ТВ29
			П.ТВ30
			П.ТВ31
			П.ТВ32
			П.ТВ33
			П.ТВ34
			П.ТВ35
			П.ТВ36
			П.ТВ37
			П.ТВ38
			П.ТВ39
			П.ТВ40
			П.ТВ41
			П.ТВ42

			П.ТВ43
			П.ТВ44
			П.ТВ45
			П.ТВ46
			П.ТВ47
			П.ТВ48
			П.ТВ49
			П.ТВ50
			Т.Д1_1
			Т.Д2_1
			Т.Д3_1
			Т.Д4_1
			Т.Д5_1
			Т.Д6_1
			Т.Д7_1
			Т.Д8_1
			Т.Д9_1
			Т.Д10_1
			Т.Д11_1
			Т.Д12_1
			Т.Д13_1
			Т.Д14_1
			Т.КС1_1
			Т.КС2_1
			Т.КС3_1
			Т.КС4_1
			Т.КС5_1
			Т.КС6_1
			Т.КС7_1
			Т.КС8_1
			Т.КС9_1
			Т.КС10_1
			Т.КС11_1
			Т.КС12_1
			Т.КС13_1
			Т.КС14_1
			Т.КС15_1
			Т.КС16_1
			Т.КС17_1

			T.KC18_1
			T.KC19_1
			T.KC20_1
			T.KC21_1
			T.KC22_1
			T.KC23_1
			T.KC24_1
			T.KC25_1
			T.KC26_1
			T.KC27_1
			T.KC28_1
			T.KC29_1
			T.KC30_1
			T.KC31_1
			T.KC32_1
			T.KC33_1
			T.KC34_1
			T.KC35_1
			T.KC36_1
			T.KC37_1
			T.KC38_1
			T.KC39_1
			T.KC40_1
			T.KC41_1
			T.KC42_1

			T.KC43_1
			T.KC44_1
			T.KC45_1
			T.KC46_1
			T.KC47_1
			T.KC48_1
			T.KC49_1
			T.T1_1
			T.T2_1
			T.T3_1
			T.T4_1
			T.T5_1
			T.T6_1
			T.T7_1
			T.T8_1
			T.T9_1
			T.T10_1
			T.T11_1
			T.T12_1
			T.T13_1
			T.T14_1
			T.T15_1
			T.T16_1
			T.T17_1
			T.T18_1
			T.T19_1
			T.T20_1
			T.Y1_1
			T.Y2_1
			T.Y3_1
			T.Y4_1
			T.Y5_1
			T.Y6_1
			T.Y7_1
			T.Y8_1
			T.Y9_1
			T.Y10_1
			T.Y11_1
			T.Y12_1



			T.Y13_1
			T.Y14_1
			T.Y15_1
			T.Y16_1
			T.Y17_1
			T.Y18_1
			T.Y19_1
			T.Y20_1
			T.Y21_1
			T.Y22_1
			T.Y23_1
			T.Y24_1
			T.Y25_1
			T.Y26_1
			T.Y27_1
			T.Y28_1
			T.Y29_1
			T.Y30_1
			T.Y31_1
			T.Y32_1
			T.Y33_1
			T.Y34_1
			T.Y35_1
			T.Y36_1
			T.Y37_1
			T.Y38_1
			T.Y39_1
			T.Y40_1
			T.Y41_1
			T.Y42_1
			T.Y43_1
			T.Y44_1
			T.Д1_2
			T.KC1_2
			T.KC2_2
			T.KC3_2
			T.KC4_2
			T.KC5_2
			T.KC6_2
			T.KC7_2
			T.KC8_2
			T.KC9_2
			T.KC10_2
			T.KC11_2

			T.KC12_2
			T.KC13_2
			T.KC14_2
			T.KC15_2
			T.KC16_2
			T.KC17_2
			T.KC18_2
			T.KC19_2
			T.KC20_2
			T.KC21_2
			T.KC22_2
			T.KC23_2
			T.KC24_2
			T.T1_2
			T.T2_2
			T.T3_2
			T.T4_2
			T.T5_2
			T.T6_2
			T.T7_2
			T.T8_2
			T.T9_2
			T.T10_2
			T.T11_2
			T.T12_2
			T.T13_2
			T.T14_2
			T.T15_2
			T.T16_2
			T.T17_2
			T.T18_2
			T.T19_2
			T.T20_2
			T.T21_2

			<p>T.Y1_2 T.Y2_2 T.Y3_2 T.Y4_2</p> <p><b>Уметь:</b> выбирать методологию и технологию проектирования информационных систем; обосновывать архитектуру ИС</p> <p>П.П1 П.П2 П.П3 П.П4 П.П5 П.П6 П.П7 П.П8 П.П9 П.П10 П.П11 П.П12 П.П13 П.П14 П.П15 П.П16 П.П17 П.П18 П.П19 П.П20 П.П21 П.П22 П.П23 П.П24 П.П25 П.П26 П.П27 П.П28 П.П29 П.П30 П.П31 П.П32 П.ТВ2 П.ТВ3 П.ТВ4 П.ТВ5 П.ТВ6 П.ТВ7 П.ТВ8 П.ТВ9</p>
--	--	--	---

			П.ТВ10
			П.ТВ11
			П.ТВ12
			П.ТВ13
			П.ТВ14
			П.ТВ15
			П.ТВ16
			П.ТВ17
			П.ТВ18
			П.ТВ19
			П.ТВ20
			П.ТВ21
			П.ТВ22
			П.ТВ23
			П.ТВ24
			П.ТВ25
			П.ТВ26
			П.ТВ27
			П.ТВ28
			П.ТВ29
			П.ТВ30
			П.ТВ31
			П.ТВ32
			П.ТВ33
			П.ТВ34
			П.ТВ36
			П.ТВ37
			П.ТВ38
			П.ТВ39
			П.ТВ40
			П.ТВ41
			П.ТВ42
			П.ТВ43
			П.ТВ44
			П.ТВ45
			П.ТВ46
			П.ТВ47
			П.ТВ48
			П.ТВ49
			П.ТВ50
			Т.Д1_1
			Т.Д2_1
			Т.Д3_1
			Т.Д4_1
			Т.Д5_1



			T.KC26_1
			T.KC27_1
			T.KC28_1
			T.KC29_1
			T.KC30_1
			T.KC31_1
			T.KC32_1
			T.KC33_1
			T.KC34_1
			T.KC35_1
			T.KC36_1
			T.KC37_1
			T.KC38_1
			T.KC39_1
			T.KC40_1
			T.KC41_1
			T.KC42_1
			T.KC43_1
			T.KC44_1
			T.KC45_1
			T.KC46_1
			T.KC47_1
			T.KC48_1
			T.KC49_1

			T.T1_1
			T.T2_1
			T.T3_1
			T.T4_1
			T.T5_1
			T.T6_1
			T.T7_1
			T.T8_1
			T.T9_1
			T.T10_1
			T.T11_1
			T.T12_1
			T.T13_1
			T.T14_1
			T.T15_1
			T.T16_1
			T.T17_1
			T.T18_1
			T.T19_1
			T.KC1_2
			T.KC2_2
			T.KC3_2
			T.KC4_2
			T.KC5_2
			T.KC6_2
			T.KC7_2
			T.KC8_2
			T.KC9_2
			T.KC10_2
			T.KC11_2
			T.KC12_2
			T.KC13_2
			T.KC14_2
			T.KC15_2
			T.KC16_2
			T.KC17_2
			T.KC18_2
			T.KC19_2

			T.KC20_2
			T.KC21_2
			T.KC22_2
			T.KC23_2
			T.KC24_2
		ОПК-8.2. Способен решать научные задачи в связи с поставленной целью и в соответствии с выбранной методикой.	<b>Знать:</b> методы оценки экономической эффективности и качества, особенности процессного подхода к управлению прикладными ИС; современные ИКТ в процессном управлении П.П1 П.П2 П.П3 П.П4 П.П5 П.П6 П.П7 П.П8 П.П9 П.П10 П.П11 П.П12 П.П13 П.П14 П.П15 П.П16 П.П17 П.П18 П.П19 П.П20 П.П21 П.П22 П.П23 П.П24 П.П25 П.П26 П.П27 П.П28 П.П29 П.П30 П.П31 П.П32 П.ТВ1



			П.ТБ2
			П.ТБ3
			П.ТБ4
			П.ТБ5
			П.ТБ6
			П.ТБ7
			П.ТБ8
			П.ТБ9
			П.ТБ10
			П.ТБ11
			П.ТБ12
			П.ТБ13
			П.ТБ14
			П.ТБ15
			П.ТБ16
			П.ТБ17
			П.ТБ18
			П.ТБ19
			П.ТБ20
			П.ТБ21
			П.ТБ22
			П.ТБ23
			П.ТБ24
			П.ТБ25
			П.ТБ26
			П.ТБ27
			П.ТБ28
			П.ТБ29
			П.ТБ30
			П.ТБ31
			П.ТБ32
			П.ТБ33
			П.ТБ34
			П.ТБ35
			П.ТБ36
			П.ТБ37
			П.ТБ38
			П.ТБ39
			П.ТБ40
			П.ТБ41
			П.ТБ42
			П.ТБ43
			П.ТБ44
			П.ТБ45
			П.ТБ46

			П.ТВ47
			П.ТВ48
			П.ТВ49
			П.ТВ50
			Т.Д1_1
			Т.Д2_1
			Т.Д3_1
			Т.Д4_1
			Т.Д5_1
			Т.Д6_1
			Т.Д7_1
			Т.Д8_1
			Т.Д9_1
			Т.Д10_1
			Т.Д11_1
			Т.Д12_1
			Т.Д13_1
			Т.Д14_1
			Т.КС1_1
			Т.КС2_1
			Т.КС3_1
			Т.КС4_1
			Т.КС5_1
			Т.КС6_1
			Т.КС7_1
			Т.КС8_1
			Т.КС9_1
			Т.КС10_1
			Т.КС11_1
			Т.КС12_1
			Т.КС13_1
			Т.КС14_1
			Т.КС15_1
			Т.КС16_1
			Т.КС17_1
			Т.КС18_1
			Т.КС19_1
			Т.КС20_1

			T.KC21_1
			T.KC22_1
			T.KC23_1
			T.KC24_1
			T.KC25_1
			T.KC26_1
			T.KC27_1
			T.KC28_1
			T.KC29_1
			T.KC30_1
			T.KC31_1
			T.KC32_1
			T.KC33_1
			T.KC34_1
			T.KC35_1
			T.KC36_1
			T.KC37_1
			T.KC38_1
			T.KC39_1
			T.KC40_1
			T.KC41_1
			T.KC42_1
			T.KC43_1
			T.KC44_1

			T.KC45_1
			T.KC46_1
			T.KC47_1
			T.KC48_1
			T.KC49_1
			T.T1_1
			T.T2_1
			T.T3_1
			T.T4_1
			T.T5_1
			T.T6_1
			T.T7_1
			T.T8_1
			T.T9_1
			T.T10_1
			T.T11_1
			T.T12_1
			T.T13_1
			T.T14_1
			T.T15_1
			T.T16_1
			T.T17_1
			T.T18_1
			T.T19_1
			T.T20_1
			T.Y1_1
			T.Y2_1
			T.Y3_1
			T.Y4_1
			T.Y5_1
			T.Y6_1
			T.Y7_1
			T.Y8_1
			T.Y9_1
			T.Y10_1
			T.Y11_1
			T.Y12_1
			T.Y13_1
			T.Y14_1
			T.Y15_1
			T.Y16_1

			T.Y17_1
			T.Y18_1
			T.Y19_1
			T.Y20_1
			T.Y21_1
			T.Y22_1
			T.Y23_1
			T.Y24_1
			T.Y25_1
			T.Y26_1
			T.Y27_1
			T.Y28_1
			T.Y29_1
			T.Y30_1
			T.Y31_1
			T.Y32_1
			T.Y33_1
			T.Y34_1
			T.Y35_1
			T.Y36_1
			T.Y37_1
			T.Y38_1
			T.Y39_1
			T.Y40_1
			T.Y41_1
			T.Y42_1
			T.Y43_1
			T.Y44_1
			T.Д1_2
			T.KC1_2
			T.KC2_2
			T.KC3_2
			T.KC4_2
			T.KC5_2
			T.KC6_2
			T.KC7_2
			T.KC8_2
			T.KC9_2
			T.KC10_2
			T.KC11_2
			T.KC12_2
			T.KC13_2
			T.KC14_2

			T.KC15_2
			T.KC16_2
			T.KC17_2
			T.KC18_2
			T.KC19_2
			T.KC20_2
			T.KC21_2
			T.KC22_2
			T.KC23_2
			T.KC24_2
			T.T1_2
			T.T2_2
			T.T3_2
			T.T4_2
			T.T5_2
			T.T6_2
			T.T7_2
			T.T8_2
			T.T9_2
			T.T10_2
			T.T11_2
			T.T12_2
			T.T13_2
			T.T14_2
			T.T15_2
			T.T16_2
			T.T17_2
			T.T18_2
			T.T19_2
			T.T20_2
			T.T21_2
			T.Y1_2
			T.Y2_2
			T.Y3_2
			T.Y4_2
			<b>Уметь:</b> управлять

			<p>проектами ИС на всех стадиях жизненного цикла, оценивать эффективность и качество проекта; применять современные методы управления проектами и сервисами ИС</p>	<p>П.П1 П.П2 П.П3 П.П4 П.П5 П.П6 П.П7 П.П8 П.П9 П.П10 П.П11 П.П12 П.П13 П.П14 П.П15 П.П16 П.П17 П.П18 П.П19 П.П20 П.П21 П.П22 П.П23 П.П24 П.П25 П.П26 П.П27 П.П28 П.П29 П.П30 П.П31 П.П32 П.ТВ2 П.ТВ3 П.ТВ4 П.ТВ5 П.ТВ6 П.ТВ7 П.ТВ8 П.ТВ9 П.ТВ10 П.ТВ11 П.ТВ12 П.ТВ13 П.ТВ14</p>
--	--	--	--	--

			П.ТB15
			П.ТB16
			П.ТB17
			П.ТB18
			П.ТB19
			П.ТB20
			П.ТB21
			П.ТB22
			П.ТB23
			П.ТB24
			П.ТB25
			П.ТB26
			П.ТB27
			П.ТB28
			П.ТB29
			П.ТB30
			П.ТB31
			П.ТB32
			П.ТB33
			П.ТB34
			П.ТB36
			П.ТB37
			П.ТB38
			П.ТB39
			П.ТB40
			П.ТB41
			П.ТB42
			П.ТB43
			П.ТB44
			П.ТB45
			П.ТB46
			П.ТB47
			П.ТB48
			П.ТB49
			П.ТB50
			Т.КC1_1
			Т.КC2_1
			Т.КC3_1
			Т.КC4_1
			Т.КC5_1
			Т.КC6_1
			Т.КC7_1
			Т.КC8_1
			Т.КC9_1
			Т.КC10_1



			T.KC11_1
			T.KC12_1
			T.KC13_1
			T.KC14_1
			T.KC15_1
			T.KC16_1
			T.KC17_1
			T.KC18_1
			T.KC19_1
			T.KC20_1
			T.KC21_1
			T.KC22_1
			T.KC23_1
			T.KC24_1
			T.KC25_1
			T.KC26_1
			T.KC27_1
			T.KC28_1
			T.KC29_1
			T.KC30_1
			T.KC31_1
			T.KC32_1
			T.KC33_1
			T.KC34_1
			T.KC35_1

			T.KC36_1
			T.KC37_1
			T.KC38_1
			T.KC39_1
			T.KC40_1
			T.KC41_1
			T.KC42_1
			T.KC43_1
			T.KC44_1
			T.KC45_1
			T.KC46_1
			T.KC47_1
			T.KC48_1
			T.KC49_1
			T.KC1_2
			T.KC2_2
			T.KC3_2
			T.KC4_2
			T.KC5_2
			T.KC6_2
			T.KC7_2
			T.KC8_2
			T.KC9_2
			T.KC10_2
			T.KC11_2
			T.KC12_2
			T.KC13_2
			T.KC14_2
			T.KC15_2

			T.KC16_2
			T.KC17_2
			T.KC18_2
			T.KC19_2
			T.KC20_2
			T.KC21_2
			T.KC22_2
			T.KC23_2
			T.KC24_2

### Контрольные задания. Текущая аттестация

<b>доклад / конференция / реферат - Раздел 1. Архитектура современных программных систем</b>	<b>Номер задания</b>
Краткий обзор видов современного ПО.	T.Д1_1
Обзор требований к современному ПО.	T.Д2_1
Обзор современных платформ для разработки ПО.	T.Д3_1
Многоярусные архитектуры ПО.	T.Д4_1
Многоклиентские архитектуры ПО.	T.Д5_1
Обзор сервисно-ориентированных архитектур ПО.	T.Д6_1
Обзор концепции Software as a Service.	T.Д7_1
Облачные вычисления как пример многоярусной, многоклиентской и сервисно-ориентированной архитектуры.	T.Д8_1
Основные процессы жизненного цикла программных систем.	T.Д9_1
Вспомогательные процессы жизненного цикла программных систем	T.Д10_1
Организационные процессы жизненного цикла программных систем	T.Д11_1
Взаимосвязь между процессами жизненного цикла программного обеспечения.	T.Д12_1
Модели и стадии жизненного цикла программного обеспечения.	T.Д13_1
Виды моделей жизненного цикла программного обеспечения.	T.Д14_1

<b>кейс - Раздел 1. Архитектура современных программных систем</b>	<b>Номер задания</b>
--	----------------------

<p>Кейс-задание - "система поликлиники":</p> <p>Проектирование архитектуры программного обеспечения.</p> <p>Для формирования структуры программного обеспечения произведите проектирование его архитектуры.</p> <p>Архитектурный вид проектирования состоит из двух основных компонентов: структурных элементов и отношений между ними. При этом схематически архитектура может быть представлена в логическом или физическом виде.</p> <p>Если структурные элементы представляют собой концептуальные (логические) компоненты ПО, такие как прецеденты, классы, процессы, состояния и др., то такая модель программной архитектуры будет соответствовать логической. Физическая же архитектура будет являться реализацией логической структуры программной системы, когда ее элементами станут физически существующие компоненты, такие как программы, базы и файлы данных, интерфейсы и т. д.</p> <p>Основные этапы, которые являются общими для любых процессов проектирования системной архитектуры:</p> <ol style="list-style-type: none"> <li>1) структурирование системы, на этом этапе система структурируется в виде набора относительно независимых подсистем, при этом определяются виды взаимодействия между ними;</li> <li>2) моделирование управления, при этом создается базовая модель управления отношениями между элементами системы;</li> <li>3) модульная декомпозиция – этап, на котором определенные на первом этапе подсистемы разбиваются на отдельные процессы (модули), а также определяются их типы и взаимные связи.</li> </ol> <p>Как правило, разрабатывается четыре типа моделей архитектуры ПО:</p> <ol style="list-style-type: none"> <li>1) статическая структурная модель, которая отображает подсистемы или компоненты, разрабатываемые на следующих этапах независимо;</li> <li>2) динамическая модель процессов, представляющая организацию процессов во время функционирования системы;</li> <li>3) интерфейсная модель, устанавливающая сервисы, предоставляемые каждой подсистемой через общесистемный интерфейс;</li> <li>4) модели отношений, показывающие взаимодействия между элементами системы, такие как потоки данных.</li> </ol>	<p>T.KC1_1</p>
<p>Кейс-задание - "Система малого предприятия ":</p> <p>Проектирование архитектуры программного обеспечения.</p> <p>Для формирования структуры программного обеспечения произведите проектирование его архитектуры.</p> <p>Архитектурный вид проектирования состоит из двух основных компонентов: структурных элементов и отношений между ними. При этом схематически архитектура может быть представлена в логическом или физическом виде.</p> <p>Если структурные элементы представляют собой концептуальные (логические) компоненты ПО, такие как прецеденты, классы, процессы, состояния и др., то такая модель программной архитектуры будет соответствовать логической. Физическая же архитектура будет являться реализацией логической структуры программной системы, когда ее элементами станут физически существующие компоненты, такие как программы, базы и файлы данных, интерфейсы и т. д.</p> <p>Основные этапы, которые являются общими для любых процессов проектирования системной архитектуры:</p> <ol style="list-style-type: none"> <li>1) структурирование системы, на этом этапе система структурируется в виде набора относительно независимых подсистем, при этом определяются виды взаимодействия между ними;</li> <li>2) моделирование управления, при этом создается базовая модель управления отношениями между элементами системы;</li> <li>3) модульная декомпозиция – этап, на котором определенные на первом этапе подсистемы разбиваются на отдельные процессы (модули), а также определяются их типы и взаимные связи.</li> </ol> <p>Как правило, разрабатывается четыре типа моделей архитектуры ПО:</p> <ol style="list-style-type: none"> <li>1) статическая структурная модель, которая отображает подсистемы или компоненты,</li> </ol>	<p>T.KC2_1</p>

<p>разрабатываемые на следующих этапах независимо;</p> <p>2) динамическая модель процессов, представляющая организацию процессов во время функционирования системы;</p> <p>3) интерфейсная модель, устанавливающая сервисы, предоставляемые каждой подсистемой через общесистемный интерфейс;</p> <p>4) модели отношений, показывающие взаимодействия между элементами системы, такие как потоки данных.</p>	
<p>Кейс-задание - "Система организационного управления ":</p> <p>Проектирование архитектуры программного обеспечения.</p> <p>Для формирования структуры программного обеспечения произведите проектирование его архитектуры.</p> <p>Архитектурный вид проектирования состоит из двух основных компонентов: структурных элементов и отношений между ними. При этом схематически архитектура может быть представлена в логическом или физическом виде.</p> <p>Если структурные элементы представляют собой концептуальные (логические) компоненты ПО, такие как прецеденты, классы, процессы, состояния и др., то такая модель программной архитектуры будет соответствовать логической. Физическая же архитектура будет являться реализацией логической структуры программной системы, когда ее элементами станут физически существующие компоненты, такие как программы, базы и файлы данных, интерфейсы и т. д.</p> <p>Основные этапы, которые являются общими для любых процессов проектирования системной архитектуры:</p> <p>1) структурирование системы, на этом этапе система структурируется в виде набора относительно независимых подсистем, при этом определяются виды взаимодействия между ними;</p> <p>2) моделирование управления, при этом создается базовая модель управления отношениями между элементами системы;</p> <p>3) модульная декомпозиция – этап, на котором определенные на первом этапе подсистемы разбиваются на отдельные процессы (модули), а также определяются их типы и взаимные связи.</p> <p>Как правило, разрабатывается четыре типа моделей архитектуры ПО:</p> <p>1) статическая структурная модель, которая отображает подсистемы или компоненты, разрабатываемые на следующих этапах независимо;</p> <p>2) динамическая модель процессов, представляющая организацию процессов во время функционирования системы;</p> <p>3) интерфейсная модель, устанавливающая сервисы, предоставляемые каждой подсистемой через общесистемный интерфейс;</p> <p>4) модели отношений, показывающие взаимодействия между элементами системы, такие как потоки данных.</p>	Т.КС3_1
<p>Кейс-задание - "Система парикмахерской ":</p> <p>Проектирование архитектуры программного обеспечения.</p> <p>Для формирования структуры программного обеспечения произведите проектирование его архитектуры.</p> <p>Архитектурный вид проектирования состоит из двух основных компонентов: структурных элементов и отношений между ними. При этом схематически архитектура может быть представлена в логическом или физическом виде.</p> <p>Если структурные элементы представляют собой концептуальные (логические) компоненты ПО, такие как прецеденты, классы, процессы, состояния и др., то такая модель программной архитектуры будет соответствовать логической. Физическая же архитектура будет являться реализацией логической структуры программной системы, когда ее элементами станут физически существующие компоненты, такие как программы, базы и файлы данных, интерфейсы и т. д.</p> <p>Основные этапы, которые являются общими для любых процессов проектирования системной архитектуры:</p> <p>1) структурирование системы, на этом этапе система структурируется в виде набора относительно независимых подсистем, при этом определяются виды взаимодействия между</p>	Т.КС4_1

<p>ними;</p> <p>2) моделирование управления, при этом создается базовая модель управления отношениями между элементами системы;</p> <p>3) модульная декомпозиция – этап, на котором определенные на первом этапе подсистемы разбиваются на отдельные процессы (модули), а также определяются их типы и взаимные связи. Как правило, разрабатывается четыре типа моделей архитектуры ПО:</p> <p>1) статическая структурная модель, которая отображает подсистемы или компоненты, разрабатываемые на следующих этапах независимо;</p> <p>2) динамическая модель процессов, представляющая организацию процессов во время функционирования системы;</p> <p>3) интерфейсная модель, устанавливающая сервисы, предоставляемые каждой подсистемой через общесистемный интерфейс;</p> <p>4) модели отношений, показывающие взаимодействия между элементами системы, такие как потоки данных.</p>	
<p>Кейс-задание - "Система библиотеки ":</p> <p>Проектирование архитектуры программного обеспечения.</p> <p>Для формирования структуры программного обеспечения произведите проектирование его архитектуры.</p> <p>Архитектурный вид проектирования состоит из двух основных компонентов: структурных элементов и отношений между ними. При этом схематически архитектура может быть представлена в логическом или физическом виде.</p> <p>Если структурные элементы представляют собой концептуальные (логические) компоненты ПО, такие как прецеденты, классы, процессы, состояния и др., то такая модель программной архитектуры будет соответствовать логической. Физическая же архитектура будет являться реализацией логической структуры программной системы, когда ее элементами станут физически существующие компоненты, такие как программы, базы и файлы данных, интерфейсы и т. д.</p> <p>Основные этапы, которые являются общими для любых процессов проектирования системной архитектуры:</p> <p>1) структурирование системы, на этом этапе система структурируется в виде набора относительно независимых подсистем, при этом определяются виды взаимодействия между ними;</p> <p>2) моделирование управления, при этом создается базовая модель управления отношениями между элементами системы;</p> <p>3) модульная декомпозиция – этап, на котором определенные на первом этапе подсистемы разбиваются на отдельные процессы (модули), а также определяются их типы и взаимные связи. Как правило, разрабатывается четыре типа моделей архитектуры ПО:</p> <p>1) статическая структурная модель, которая отображает подсистемы или компоненты, разрабатываемые на следующих этапах независимо;</p> <p>2) динамическая модель процессов, представляющая организацию процессов во время функционирования системы;</p> <p>3) интерфейсная модель, устанавливающая сервисы, предоставляемые каждой подсистемой через общесистемный интерфейс;</p> <p>4) модели отношений, показывающие взаимодействия между элементами системы, такие как потоки данных.</p>	<p>T.KC5_1</p>
<p>Кейс-задание - "Система управления учебным процессом ":</p> <p>Проектирование архитектуры программного обеспечения.</p> <p>Для формирования структуры программного обеспечения произведите проектирование его архитектуры.</p> <p>Архитектурный вид проектирования состоит из двух основных компонентов: структурных элементов и отношений между ними. При этом схематически архитектура может быть представлена в логическом или физическом виде.</p> <p>Если структурные элементы представляют собой концептуальные (логические) компоненты ПО, такие как прецеденты, классы, процессы, состояния и др., то такая модель программной архитектуры будет соответствовать логической. Физическая же архитектура будет являться</p>	<p>T.KC6_1</p>

<p>реализацией логической структуры программной системы, когда ее элементами станут физически существующие компоненты, такие как программы, базы и файлы данных, интерфейсы и т. д.</p> <p>Основные этапы, которые являются общими для любых процессов проектирования системной архитектуры:</p> <ol style="list-style-type: none"> <li>1) структурирование системы, на этом этапе система структурируется в виде набора относительно независимых подсистем, при этом определяются виды взаимодействия между ними;</li> <li>2) моделирование управления, при этом создается базовая модель управления отношениями между элементами системы;</li> <li>3) модульная декомпозиция – этап, на котором определенные на первом этапе подсистемы разбиваются на отдельные процессы (модули), а также определяются их типы и взаимные связи.</li> </ol> <p>Как правило, разрабатывается четыре типа моделей архитектуры ПО:</p> <ol style="list-style-type: none"> <li>1) статическая структурная модель, которая отображает подсистемы или компоненты, разрабатываемые на следующих этапах независимо;</li> <li>2) динамическая модель процессов, представляющая организацию процессов во время функционирования системы;</li> <li>3) интерфейсная модель, устанавливающая сервисы, предоставляемые каждой подсистемой через общесистемный интерфейс;</li> <li>4) модели отношений, показывающие взаимодействия между элементами системы, такие как потоки данных.</li> </ol>	
<p>Кейс-задание - "Система "Клиент-Банк" ": Проектирование архитектуры программного обеспечения. Для формирования структуры программного обеспечения произведите проектирование его архитектуры. Архитектурный вид проектирования состоит из двух основных компонентов: структурных элементов и отношений между ними. При этом схематически архитектура может быть представлена в логическом или физическом виде. Если структурные элементы представляют собой концептуальные (логические) компоненты ПО, такие как прецеденты, классы, процессы, состояния и др., то такая модель программной архитектуры будет соответствовать логической. Физическая же архитектура будет являться реализацией логической структуры программной системы, когда ее элементами станут физически существующие компоненты, такие как программы, базы и файлы данных, интерфейсы и т. д.</p> <p>Основные этапы, которые являются общими для любых процессов проектирования системной архитектуры:</p> <ol style="list-style-type: none"> <li>1) структурирование системы, на этом этапе система структурируется в виде набора относительно независимых подсистем, при этом определяются виды взаимодействия между ними;</li> <li>2) моделирование управления, при этом создается базовая модель управления отношениями между элементами системы;</li> <li>3) модульная декомпозиция – этап, на котором определенные на первом этапе подсистемы разбиваются на отдельные процессы (модули), а также определяются их типы и взаимные связи.</li> </ol> <p>Как правило, разрабатывается четыре типа моделей архитектуры ПО:</p> <ol style="list-style-type: none"> <li>1) статическая структурная модель, которая отображает подсистемы или компоненты, разрабатываемые на следующих этапах независимо;</li> <li>2) динамическая модель процессов, представляющая организацию процессов во время функционирования системы;</li> <li>3) интерфейсная модель, устанавливающая сервисы, предоставляемые каждой подсистемой через общесистемный интерфейс;</li> <li>4) модели отношений, показывающие взаимодействия между элементами системы, такие как потоки данных.</li> </ol>	Т.КС7_1
<p>Кейс-задание - "Универсальная система "Склад" ": Проектирование архитектуры программного обеспечения. Для формирования структуры программного обеспечения произведите проектирование его</p>	Т.КС8_1

<p>архитектуры.</p> <p>Архитектурный вид проектирования состоит из двух основных компонентов: структурных элементов и отношений между ними. При этом схематически архитектура может быть представлена в логическом или физическом виде.</p> <p>Если структурные элементы представляют собой концептуальные (логические) компоненты ПО, такие как прецеденты, классы, процессы, состояния и др., то такая модель программной архитектуры будет соответствовать логической. Физическая же архитектура будет являться реализацией логической структуры программной системы, когда ее элементами станут физически существующие компоненты, такие как программы, базы и файлы данных, интерфейсы и т. д.</p> <p>Основные этапы, которые являются общими для любых процессов проектирования системной архитектуры:</p> <ol style="list-style-type: none"> <li>1) структурирование системы, на этом этапе система структурируется в виде набора относительно независимых подсистем, при этом определяются виды взаимодействия между ними;</li> <li>2) моделирование управления, при этом создается базовая модель управления отношениями между элементами системы;</li> <li>3) модульная декомпозиция – этап, на котором определенные на первом этапе подсистемы разбиваются на отдельные процессы (модули), а также определяются их типы и взаимные связи.</li> </ol> <p>Как правило, разрабатывается четыре типа моделей архитектуры ПО:</p> <ol style="list-style-type: none"> <li>1) статическая структурная модель, которая отображает подсистемы или компоненты, разрабатываемые на следующих этапах независимо;</li> <li>2) динамическая модель процессов, представляющая организацию процессов во время функционирования системы;</li> <li>3) интерфейсная модель, устанавливающая сервисы, предоставляемые каждой подсистемой через общесистемный интерфейс;</li> <li>4) модели отношений, показывающие взаимодействия между элементами системы, такие как потоки данных.</li> </ol>	
<p>Кейс-задание - "Система столовой ":</p> <p>Проектирование архитектуры программного обеспечения.</p> <p>Для формирования структуры программного обеспечения произведите проектирование его архитектуры.</p> <p>Архитектурный вид проектирования состоит из двух основных компонентов: структурных элементов и отношений между ними. При этом схематически архитектура может быть представлена в логическом или физическом виде.</p> <p>Если структурные элементы представляют собой концептуальные (логические) компоненты ПО, такие как прецеденты, классы, процессы, состояния и др., то такая модель программной архитектуры будет соответствовать логической. Физическая же архитектура будет являться реализацией логической структуры программной системы, когда ее элементами станут физически существующие компоненты, такие как программы, базы и файлы данных, интерфейсы и т. д.</p> <p>Основные этапы, которые являются общими для любых процессов проектирования системной архитектуры:</p> <ol style="list-style-type: none"> <li>1) структурирование системы, на этом этапе система структурируется в виде набора относительно независимых подсистем, при этом определяются виды взаимодействия между ними;</li> <li>2) моделирование управления, при этом создается базовая модель управления отношениями между элементами системы;</li> <li>3) модульная декомпозиция – этап, на котором определенные на первом этапе подсистемы разбиваются на отдельные процессы (модули), а также определяются их типы и взаимные связи.</li> </ol> <p>Как правило, разрабатывается четыре типа моделей архитектуры ПО:</p> <ol style="list-style-type: none"> <li>1) статическая структурная модель, которая отображает подсистемы или компоненты, разрабатываемые на следующих этапах независимо;</li> <li>2) динамическая модель процессов, представляющая организацию процессов во время функционирования системы;</li> </ol>	<p>Т.КС9_1</p>



<p>3) интерфейсная модель, устанавливающая сервисы, предоставляемые каждой подсистемой через общесистемный интерфейс;</p> <p>4) модели отношений, показывающие взаимодействия между элементами системы, такие как потоки данных.</p>	
<p>Кейс-задание - "Система СТО ":</p> <p>Проектирование архитектуры программного обеспечения.</p> <p>Для формирования структуры программного обеспечения произведите проектирование его архитектуры.</p> <p>Архитектурный вид проектирования состоит из двух основных компонентов: структурных элементов и отношений между ними. При этом схематически архитектура может быть представлена в логическом или физическом виде.</p> <p>Если структурные элементы представляют собой концептуальные (логические) компоненты ПО, такие как прецеденты, классы, процессы, состояния и др., то такая модель программной архитектуры будет соответствовать логической. Физическая же архитектура будет являться реализацией логической структуры программной системы, когда ее элементами станут физически существующие компоненты, такие как программы, базы и файлы данных, интерфейсы и т. д.</p> <p>Основные этапы, которые являются общими для любых процессов проектирования системной архитектуры:</p> <p>1) структурирование системы, на этом этапе система структурируется в виде набора относительно независимых подсистем, при этом определяются виды взаимодействия между ними;</p> <p>2) моделирование управления, при этом создается базовая модель управления отношениями между элементами системы;</p> <p>3) модульная декомпозиция – этап, на котором определенные на первом этапе подсистемы разбиваются на отдельные процессы (модули), а также определяются их типы и взаимные связи.</p> <p>Как правило, разрабатывается четыре типа моделей архитектуры ПО:</p> <p>1) статическая структурная модель, которая отображает подсистемы или компоненты, разрабатываемые на следующих этапах независимо;</p> <p>2) динамическая модель процессов, представляющая организацию процессов во время функционирования системы;</p> <p>3) интерфейсная модель, устанавливающая сервисы, предоставляемые каждой подсистемой через общесистемный интерфейс;</p> <p>4) модели отношений, показывающие взаимодействия между элементами системы, такие как потоки данных.</p>	<p>Т.КС10_1</p>
<p>Кейс-задание - "Система автозаправки ":</p> <p>Проектирование архитектуры программного обеспечения.</p> <p>Для формирования структуры программного обеспечения произведите проектирование его архитектуры.</p> <p>Архитектурный вид проектирования состоит из двух основных компонентов: структурных элементов и отношений между ними. При этом схематически архитектура может быть представлена в логическом или физическом виде.</p> <p>Если структурные элементы представляют собой концептуальные (логические) компоненты ПО, такие как прецеденты, классы, процессы, состояния и др., то такая модель программной архитектуры будет соответствовать логической. Физическая же архитектура будет являться реализацией логической структуры программной системы, когда ее элементами станут физически существующие компоненты, такие как программы, базы и файлы данных, интерфейсы и т. д.</p> <p>Основные этапы, которые являются общими для любых процессов проектирования системной архитектуры:</p> <p>1) структурирование системы, на этом этапе система структурируется в виде набора относительно независимых подсистем, при этом определяются виды взаимодействия между ними;</p> <p>2) моделирование управления, при этом создается базовая модель управления отношениями между элементами системы;</p>	<p>Т.КС11_1</p>

<p>3) модульная декомпозиция – этап, на котором определенные на первом этапе подсистемы разбиваются на отдельные процессы (модули), а также определяются их типы и взаимные связи. Как правило, разрабатывается четыре типа моделей архитектуры ПО:</p> <p>1) статическая структурная модель, которая отображает подсистемы или компоненты, разрабатываемые на следующих этапах независимо;</p> <p>2) динамическая модель процессов, представляющая организацию процессов во время функционирования системы;</p> <p>3) интерфейсная модель, устанавливающая сервисы, предоставляемые каждой подсистемой через общесистемный интерфейс;</p> <p>4) модели отношений, показывающие взаимодействия между элементами системы, такие как потоки данных.</p>	
<p>Кейс-задание - "Система управления расписанием школы ":</p> <p>Проектирование архитектуры программного обеспечения.</p> <p>Для формирования структуры программного обеспечения произведите проектирование его архитектуры.</p> <p>Архитектурный вид проектирования состоит из двух основных компонентов: структурных элементов и отношений между ними. При этом схематически архитектура может быть представлена в логическом или физическом виде.</p> <p>Если структурные элементы представляют собой концептуальные (логические) компоненты ПО, такие как прецеденты, классы, процессы, состояния и др., то такая модель программной архитектуры будет соответствовать логической. Физическая же архитектура будет являться реализацией логической структуры программной системы, когда ее элементами станут физически существующие компоненты, такие как программы, базы и файлы данных, интерфейсы и т. д.</p> <p>Основные этапы, которые являются общими для любых процессов проектирования системной архитектуры:</p> <p>1) структурирование системы, на этом этапе система структурируется в виде набора относительно независимых подсистем, при этом определяются виды взаимодействия между ними;</p> <p>2) моделирование управления, при этом создается базовая модель управления отношениями между элементами системы;</p> <p>3) модульная декомпозиция – этап, на котором определенные на первом этапе подсистемы разбиваются на отдельные процессы (модули), а также определяются их типы и взаимные связи. Как правило, разрабатывается четыре типа моделей архитектуры ПО:</p> <p>1) статическая структурная модель, которая отображает подсистемы или компоненты, разрабатываемые на следующих этапах независимо;</p> <p>2) динамическая модель процессов, представляющая организацию процессов во время функционирования системы;</p> <p>3) интерфейсная модель, устанавливающая сервисы, предоставляемые каждой подсистемой через общесистемный интерфейс;</p> <p>4) модели отношений, показывающие взаимодействия между элементами системы, такие как потоки данных.</p>	<p>Т.КС12_1</p>
<p>Кейс-задание - "Система питания детского сада ":</p> <p>Проектирование архитектуры программного обеспечения.</p> <p>Для формирования структуры программного обеспечения произведите проектирование его архитектуры.</p> <p>Архитектурный вид проектирования состоит из двух основных компонентов: структурных элементов и отношений между ними. При этом схематически архитектура может быть представлена в логическом или физическом виде.</p> <p>Если структурные элементы представляют собой концептуальные (логические) компоненты ПО, такие как прецеденты, классы, процессы, состояния и др., то такая модель программной архитектуры будет соответствовать логической. Физическая же архитектура будет являться реализацией логической структуры программной системы, когда ее элементами станут физически существующие компоненты, такие как программы, базы и файлы данных, интерфейсы и т. д.</p>	<p>Т.КС13_1</p>

<p>Основные этапы, которые являются общими для любых процессов проектирования системной архитектуры:</p> <ol style="list-style-type: none"> <li>1) структурирование системы, на этом этапе система структурируется в виде набора относительно независимых подсистем, при этом определяются виды взаимодействия между ними;</li> <li>2) моделирование управления, при этом создается базовая модель управления отношениями между элементами системы;</li> <li>3) модульная декомпозиция – этап, на котором определенные на первом этапе подсистемы разбиваются на отдельные процессы (модули), а также определяются их типы и взаимные связи. Как правило, разрабатывается четыре типа моделей архитектуры ПО:</li> </ol> <ol style="list-style-type: none"> <li>1) статическая структурная модель, которая отображает подсистемы или компоненты, разрабатываемые на следующих этапах независимо;</li> <li>2) динамическая модель процессов, представляющая организацию процессов во время функционирования системы;</li> <li>3) интерфейсная модель, устанавливающая сервисы, предоставляемые каждой подсистемой через общесистемный интерфейс;</li> <li>4) модели отношений, показывающие взаимодействия между элементами системы, такие как потоки данных.</li> </ol>	
<p>Кейс-задание - "Система закупки товаров аптеки ":</p> <p>Проектирование архитектуры программного обеспечения.</p> <p>Для формирования структуры программного обеспечения произведите проектирование его архитектуры.</p> <p>Архитектурный вид проектирования состоит из двух основных компонентов: структурных элементов и отношений между ними. При этом схематически архитектура может быть представлена в логическом или физическом виде.</p> <p>Если структурные элементы представляют собой концептуальные (логические) компоненты ПО, такие как прецеденты, классы, процессы, состояния и др., то такая модель программной архитектуры будет соответствовать логической. Физическая же архитектура будет являться реализацией логической структуры программной системы, когда ее элементами станут физически существующие компоненты, такие как программы, базы и файлы данных, интерфейсы и т. д.</p> <p>Основные этапы, которые являются общими для любых процессов проектирования системной архитектуры:</p> <ol style="list-style-type: none"> <li>1) структурирование системы, на этом этапе система структурируется в виде набора относительно независимых подсистем, при этом определяются виды взаимодействия между ними;</li> <li>2) моделирование управления, при этом создается базовая модель управления отношениями между элементами системы;</li> <li>3) модульная декомпозиция – этап, на котором определенные на первом этапе подсистемы разбиваются на отдельные процессы (модули), а также определяются их типы и взаимные связи. Как правило, разрабатывается четыре типа моделей архитектуры ПО:</li> </ol> <ol style="list-style-type: none"> <li>1) статическая структурная модель, которая отображает подсистемы или компоненты, разрабатываемые на следующих этапах независимо;</li> <li>2) динамическая модель процессов, представляющая организацию процессов во время функционирования системы;</li> <li>3) интерфейсная модель, устанавливающая сервисы, предоставляемые каждой подсистемой через общесистемный интерфейс;</li> <li>4) модели отношений, показывающие взаимодействия между элементами системы, такие как потоки данных.</li> </ol>	<p>Т.КС14_1</p>
<p>Кейс-задание - "Система швейного ателье ":</p> <p>Проектирование архитектуры программного обеспечения.</p> <p>Для формирования структуры программного обеспечения произведите проектирование его архитектуры.</p> <p>Архитектурный вид проектирования состоит из двух основных компонентов: структурных элементов и отношений между ними. При этом схематически архитектура может быть</p>	<p>Т.КС15_1</p>

<p>представлена в логическом или физическом виде.</p> <p>Если структурные элементы представляют собой концептуальные (логические) компоненты ПО, такие как прецеденты, классы, процессы, состояния и др., то такая модель программной архитектуры будет соответствовать логической. Физическая же архитектура будет являться реализацией логической структуры программной системы, когда ее элементами станут физически существующие компоненты, такие как программы, базы и файлы данных, интерфейсы и т. д.</p> <p>Основные этапы, которые являются общими для любых процессов проектирования системной архитектуры:</p> <ol style="list-style-type: none"> <li>1) структурирование системы, на этом этапе система структурируется в виде набора относительно независимых подсистем, при этом определяются виды взаимодействия между ними;</li> <li>2) моделирование управления, при этом создается базовая модель управления отношениями между элементами системы;</li> <li>3) модульная декомпозиция – этап, на котором определенные на первом этапе подсистемы разбиваются на отдельные процессы (модули), а также определяются их типы и взаимные связи. Как правило, разрабатывается четыре типа моделей архитектуры ПО:</li> </ol> <ol style="list-style-type: none"> <li>1) статическая структурная модель, которая отображает подсистемы или компоненты, разрабатываемые на следующих этапах независимо;</li> <li>2) динамическая модель процессов, представляющая организацию процессов во время функционирования системы;</li> <li>3) интерфейсная модель, устанавливающая сервисы, предоставляемые каждой подсистемой через общесистемный интерфейс;</li> <li>4) модели отношений, показывающие взаимодействия между элементами системы, такие как потоки данных.</li> </ol>	
<p>Кейс-задание - "Система газетного киоска":</p> <p>Проектирование архитектуры программного обеспечения.</p> <p>Для формирования структуры программного обеспечения произведите проектирование его архитектуры.</p> <p>Архитектурный вид проектирования состоит из двух основных компонентов: структурных элементов и отношений между ними. При этом схематически архитектура может быть представлена в логическом или физическом виде.</p> <p>Если структурные элементы представляют собой концептуальные (логические) компоненты ПО, такие как прецеденты, классы, процессы, состояния и др., то такая модель программной архитектуры будет соответствовать логической. Физическая же архитектура будет являться реализацией логической структуры программной системы, когда ее элементами станут физически существующие компоненты, такие как программы, базы и файлы данных, интерфейсы и т. д.</p> <p>Основные этапы, которые являются общими для любых процессов проектирования системной архитектуры:</p> <ol style="list-style-type: none"> <li>1) структурирование системы, на этом этапе система структурируется в виде набора относительно независимых подсистем, при этом определяются виды взаимодействия между ними;</li> <li>2) моделирование управления, при этом создается базовая модель управления отношениями между элементами системы;</li> <li>3) модульная декомпозиция – этап, на котором определенные на первом этапе подсистемы разбиваются на отдельные процессы (модули), а также определяются их типы и взаимные связи. Как правило, разрабатывается четыре типа моделей архитектуры ПО:</li> </ol> <ol style="list-style-type: none"> <li>1) статическая структурная модель, которая отображает подсистемы или компоненты, разрабатываемые на следующих этапах независимо;</li> <li>2) динамическая модель процессов, представляющая организацию процессов во время функционирования системы;</li> <li>3) интерфейсная модель, устанавливающая сервисы, предоставляемые каждой подсистемой через общесистемный интерфейс;</li> <li>4) модели отношений, показывающие взаимодействия между элементами системы, такие как</li> </ol>	<p>Т.КС16_1</p>

<p>потоки данных.</p> <p>Кейс-задание - "Система оформления подписки на почте":          Проектирование архитектуры программного обеспечения.          Для формирования структуры программного обеспечения произведите проектирование его архитектуры.          Архитектурный вид проектирования состоит из двух основных компонентов: структурных элементов и отношений между ними. При этом схематически архитектура может быть представлена в логическом или физическом виде.          Если структурные элементы представляют собой концептуальные (логические) компоненты ПО, такие как прецеденты, классы, процессы, состояния и др., то такая модель программной архитектуры будет соответствовать логической. Физическая же архитектура будет являться реализацией логической структуры программной системы, когда ее элементами станут физически существующие компоненты, такие как программы, базы и файлы данных, интерфейсы и т. д.          Основные этапы, которые являются общими для любых процессов проектирования системной архитектуры:</p> <ol style="list-style-type: none"> <li>1) структурирование системы, на этом этапе система структурируется в виде набора относительно независимых подсистем, при этом определяются виды взаимодействия между ними;</li> <li>2) моделирование управления, при этом создается базовая модель управления отношениями между элементами системы;</li> <li>3) модульная декомпозиция – этап, на котором определенные на первом этапе подсистемы разбиваются на отдельные процессы (модули), а также определяются их типы и взаимные связи.</li> </ol> <p>Как правило, разрабатывается четыре типа моделей архитектуры ПО:</p> <ol style="list-style-type: none"> <li>1) статическая структурная модель, которая отображает подсистемы или компоненты, разрабатываемые на следующих этапах независимо;</li> <li>2) динамическая модель процессов, представляющая организацию процессов во время функционирования системы;</li> <li>3) интерфейсная модель, устанавливающая сервисы, предоставляемые каждой подсистемой через общесистемный интерфейс;</li> <li>4) модели отношений, показывающие взаимодействия между элементами системы, такие как потоки данных.</li> </ol>	<p>Т.КС17_1</p>
<p>Кейс-задание - "Система поликлиники ":          Жизненный цикл программного продукта          В соответствии со стандартом ISO/IEC 12207 все процессы ЖЦ ПО разделены на три группы: Основные процессы, вспомогательные процессы, организационные процессы.          В группах определено пять основных процессов: приобретение, поставка, разработка, эксплуатация и сопровождение. Восемь вспомогательных процессов обеспечивают выполнение основных процессов, а именно документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместная оценка, аудит, разрешение проблем. Четыре организационных процесса обеспечивают управление, создание инфраструктуры, усовершенствование и обучение. Опишите процесс разрабатываемого программного обеспечения по следующей схеме:</p> <ol style="list-style-type: none"> <li>1. подготовительную работу;</li> <li>2. анализ требований, предъявляемых к системе;</li> <li>3. проектирование архитектуры системы;</li> <li>4. анализ требований, предъявляемых к программному обеспечению;</li> <li>5. проектирование архитектуры программного обеспечения;</li> <li>6. детальное проектирование программного обеспечения;</li> <li>7. кодирование и тестирование программного обеспечения;</li> <li>8. интеграцию программного обеспечения;</li> <li>9. квалификационное тестирование программного обеспечения;</li> <li>10. интеграцию системы;</li> <li>11. квалификационное тестирование системы;</li> <li>12. установку программного обеспечения;</li> </ol>	<p>Т.КС18_1</p>

<p>13. приемку программного обеспечения.</p> <p>Кейс-задание - "Система малого предприятия":  Жизненный цикл программного продукта  В соответствии со стандартом ISO/IEC 12207 все процессы ЖЦ ПО разделены на три группы: Основные процессы, вспомогательные процессы, организационные процессы.  В группах определено пять основных процессов: приобретение, поставка, разработка, эксплуатация и сопровождение. Восемь вспомогательных процессов обеспечивают выполнение основных процессов, а именно документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместная оценка, аудит, разрешение проблем. Четыре организационных процесса обеспечивают управление, создание инфраструктуры, усовершенствование и обучение. Опишите процесс разрабатываемого программного обеспечения по следующей схеме:</p> <ol style="list-style-type: none"> <li>1. подготовительную работу;</li> <li>2. анализ требований, предъявляемых к системе;</li> <li>3. проектирование архитектуры системы;</li> <li>4. анализ требований, предъявляемых к программному обеспечению;</li> <li>5. проектирование архитектуры программного обеспечения;</li> <li>6. детальное проектирование программного обеспечения;</li> <li>7. кодирование и тестирование программного обеспечения;</li> <li>8. интеграцию программного обеспечения;</li> <li>9. квалификационное тестирование программного обеспечения;</li> <li>10. интеграцию системы;</li> <li>11. квалификационное тестирование системы;</li> <li>12. установку программного обеспечения;</li> <li>13. приемку программного обеспечения.</li> </ol>	<p>Т.КС19_1</p>
<p>Кейс-задание - "Система организационного управления ":  Жизненный цикл программного продукта  В соответствии со стандартом ISO/IEC 12207 все процессы ЖЦ ПО разделены на три группы: Основные процессы, вспомогательные процессы, организационные процессы.  В группах определено пять основных процессов: приобретение, поставка, разработка, эксплуатация и сопровождение. Восемь вспомогательных процессов обеспечивают выполнение основных процессов, а именно документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместная оценка, аудит, разрешение проблем. Четыре организационных процесса обеспечивают управление, создание инфраструктуры, усовершенствование и обучение. Опишите процесс разрабатываемого программного обеспечения по следующей схеме:</p> <ol style="list-style-type: none"> <li>1. подготовительную работу;</li> <li>2. анализ требований, предъявляемых к системе;</li> <li>3. проектирование архитектуры системы;</li> <li>4. анализ требований, предъявляемых к программному обеспечению;</li> <li>5. проектирование архитектуры программного обеспечения;</li> <li>6. детальное проектирование программного обеспечения;</li> <li>7. кодирование и тестирование программного обеспечения;</li> <li>8. интеграцию программного обеспечения;</li> <li>9. квалификационное тестирование программного обеспечения;</li> <li>10. интеграцию системы;</li> <li>11. квалификационное тестирование системы;</li> <li>12. установку программного обеспечения;</li> <li>13. приемку программного обеспечения.</li> </ol>	<p>Т.КС20_1</p>
<p>Кейс-задание - "Система парикмахерской ":  Жизненный цикл программного продукта  В соответствии со стандартом ISO/IEC 12207 все процессы ЖЦ ПО разделены на три группы: Основные процессы, вспомогательные процессы, организационные процессы.  В группах определено пять основных процессов: приобретение, поставка, разработка,</p>	<p>Т.КС21_1</p>

<p>эксплуатация и сопровождение. Восемь вспомогательных процессов обеспечивают выполнение основных процессов, а именно документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместная оценка, аудит, разрешение проблем. Четыре организационных процесса обеспечивают управление, создание инфраструктуры, усовершенствование и обучение. Опишите процесс разрабатываемого программного обеспечения по следующей схеме:</p> <ol style="list-style-type: none"> <li>1. подготовительную работу;</li> <li>2. анализ требований, предъявляемых к системе;</li> <li>3. проектирование архитектуры системы;</li> <li>4. анализ требований, предъявляемых к программному обеспечению;</li> <li>5. проектирование архитектуры программного обеспечения;</li> <li>6. детальное проектирование программного обеспечения;</li> <li>7. кодирование и тестирование программного обеспечения;</li> <li>8. интеграцию программного обеспечения;</li> <li>9. квалификационное тестирование программного обеспечения;</li> <li>10. интеграцию системы;</li> <li>11. квалификационное тестирование системы;</li> <li>12. установку программного обеспечения;</li> <li>13. приемку программного обеспечения.</li> </ol>	
<p>Кейс-задание - "Система библиотеки":</p> <p>Жизненный цикл программного продукта</p> <p>В соответствии со стандартом ISO/IEC 12207 все процессы ЖЦ ПО разделены на три группы: Основные процессы, вспомогательные процессы, организационные процессы.</p> <p>В группах определено пять основных процессов: приобретение, поставка, разработка, эксплуатация и сопровождение. Восемь вспомогательных процессов обеспечивают выполнение основных процессов, а именно документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместная оценка, аудит, разрешение проблем. Четыре организационных процесса обеспечивают управление, создание инфраструктуры, усовершенствование и обучение. Опишите процесс разрабатываемого программного обеспечения по следующей схеме:</p> <ol style="list-style-type: none"> <li>1. подготовительную работу;</li> <li>2. анализ требований, предъявляемых к системе;</li> <li>3. проектирование архитектуры системы;</li> <li>4. анализ требований, предъявляемых к программному обеспечению;</li> <li>5. проектирование архитектуры программного обеспечения;</li> <li>6. детальное проектирование программного обеспечения;</li> <li>7. кодирование и тестирование программного обеспечения;</li> <li>8. интеграцию программного обеспечения;</li> <li>9. квалификационное тестирование программного обеспечения;</li> <li>10. интеграцию системы;</li> <li>11. квалификационное тестирование системы;</li> <li>12. установку программного обеспечения;</li> <li>13. приемку программного обеспечения.</li> </ol>	<p>T.KC22_ 1</p>
<p>Кейс-задание - "Система управления учебным процессом":</p> <p>Жизненный цикл программного продукта</p> <p>В соответствии со стандартом ISO/IEC 12207 все процессы ЖЦ ПО разделены на три группы: Основные процессы, вспомогательные процессы, организационные процессы.</p> <p>В группах определено пять основных процессов: приобретение, поставка, разработка, эксплуатация и сопровождение. Восемь вспомогательных процессов обеспечивают выполнение основных процессов, а именно документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместная оценка, аудит, разрешение проблем. Четыре организационных процесса обеспечивают управление, создание инфраструктуры, усовершенствование и обучение. Опишите процесс разрабатываемого программного обеспечения по следующей схеме:</p> <ol style="list-style-type: none"> <li>1. подготовительную работу;</li> </ol>	<p>T.KC23_ 1</p>

<ol style="list-style-type: none"> <li>2. анализ требований, предъявляемых к системе;</li> <li>3. проектирование архитектуры системы;</li> <li>4. анализ требований, предъявляемых к программному обеспечению;</li> <li>5. проектирование архитектуры программного обеспечения;</li> <li>6. детальное проектирование программного обеспечения;</li> <li>7. кодирование и тестирование программного обеспечения;</li> <li>8. интеграцию программного обеспечения;</li> <li>9. квалификационное тестирование программного обеспечения;</li> <li>10. интеграцию системы;</li> <li>11. квалификационное тестирование системы;</li> <li>12. установку программного обеспечения;</li> <li>13. приемку программного обеспечения.</li> </ol>	
<p>Кейс-задание - "Универсальная система "Склад"": Жизненный цикл программного продукта В соответствии со стандартом ISO/IEC 12207 все процессы ЖЦ ПО разделены на три группы: Основные процессы, вспомогательные процессы, организационные процессы. В группах определено пять основных процессов: приобретение, поставка, разработка, эксплуатация и сопровождение. Восемь вспомогательных процессов обеспечивают выполнение основных процессов, а именно документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместная оценка, аудит, разрешение проблем. Четыре организационных процесса обеспечивают управление, создание инфраструктуры, усовершенствование и обучение. Опишите процесс разрабатываемого программного обеспечения по следующей схеме:</p> <ol style="list-style-type: none"> <li>1. подготовительную работу;</li> <li>2. анализ требований, предъявляемых к системе;</li> <li>3. проектирование архитектуры системы;</li> <li>4. анализ требований, предъявляемых к программному обеспечению;</li> <li>5. проектирование архитектуры программного обеспечения;</li> <li>6. детальное проектирование программного обеспечения;</li> <li>7. кодирование и тестирование программного обеспечения;</li> <li>8. интеграцию программного обеспечения;</li> <li>9. квалификационное тестирование программного обеспечения;</li> <li>10. интеграцию системы;</li> <li>11. квалификационное тестирование системы;</li> <li>12. установку программного обеспечения;</li> <li>13. приемку программного обеспечения.</li> </ol>	<p>Т.КС24_1</p>
<p>Кейс-задание - "Система столовой ": Жизненный цикл программного продукта В соответствии со стандартом ISO/IEC 12207 все процессы ЖЦ ПО разделены на три группы: Основные процессы, вспомогательные процессы, организационные процессы. В группах определено пять основных процессов: приобретение, поставка, разработка, эксплуатация и сопровождение. Восемь вспомогательных процессов обеспечивают выполнение основных процессов, а именно документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместная оценка, аудит, разрешение проблем. Четыре организационных процесса обеспечивают управление, создание инфраструктуры, усовершенствование и обучение. Опишите процесс разрабатываемого программного обеспечения по следующей схеме:</p> <ol style="list-style-type: none"> <li>1. подготовительную работу;</li> <li>2. анализ требований, предъявляемых к системе;</li> <li>3. проектирование архитектуры системы;</li> <li>4. анализ требований, предъявляемых к программному обеспечению;</li> <li>5. проектирование архитектуры программного обеспечения;</li> <li>6. детальное проектирование программного обеспечения;</li> <li>7. кодирование и тестирование программного обеспечения;</li> <li>8. интеграцию программного обеспечения;</li> </ol>	<p>Т.КС25_1</p>



<p>9. квалификационное тестирование программного обеспечения;</p> <p>10. интеграцию системы;</p> <p>11. квалификационное тестирование системы;</p> <p>12. установку программного обеспечения;</p> <p>13. приемку программного обеспечения.</p>	
<p>Кейс-задание - "Система СТО":</p> <p>Жизненный цикл программного продукта</p> <p>В соответствии со стандартом ISO/IEC 12207 все процессы ЖЦ ПО разделены на три группы: Основные процессы, вспомогательные процессы, организационные процессы.</p> <p>В группах определено пять основных процессов: приобретение, поставка, разработка, эксплуатация и сопровождение. Восемь вспомогательных процессов обеспечивают выполнение основных процессов, а именно документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместная оценка, аудит, разрешение проблем. Четыре организационных процесса обеспечивают управление, создание инфраструктуры, усовершенствование и обучение. Опишите процесс разрабатываемого программного обеспечения по следующей схеме:</p> <ol style="list-style-type: none"> <li>1. подготовительную работу;</li> <li>2. анализ требований, предъявляемых к системе;</li> <li>3. проектирование архитектуры системы;</li> <li>4. анализ требований, предъявляемых к программному обеспечению;</li> <li>5. проектирование архитектуры программного обеспечения;</li> <li>6. детальное проектирование программного обеспечения;</li> <li>7. кодирование и тестирование программного обеспечения;</li> <li>8. интеграцию программного обеспечения;</li> <li>9. квалификационное тестирование программного обеспечения;</li> <li>10. интеграцию системы;</li> <li>11. квалификационное тестирование системы;</li> <li>12. установку программного обеспечения;</li> <li>13. приемку программного обеспечения.</li> </ol>	<p>Т.КC26_1</p>
<p>Кейс-задание - "Система автозаправки":</p> <p>Жизненный цикл программного продукта</p> <p>В соответствии со стандартом ISO/IEC 12207 все процессы ЖЦ ПО разделены на три группы: Основные процессы, вспомогательные процессы, организационные процессы.</p> <p>В группах определено пять основных процессов: приобретение, поставка, разработка, эксплуатация и сопровождение. Восемь вспомогательных процессов обеспечивают выполнение основных процессов, а именно документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместная оценка, аудит, разрешение проблем. Четыре организационных процесса обеспечивают управление, создание инфраструктуры, усовершенствование и обучение. Опишите процесс разрабатываемого программного обеспечения по следующей схеме:</p> <ol style="list-style-type: none"> <li>1. подготовительную работу;</li> <li>2. анализ требований, предъявляемых к системе;</li> <li>3. проектирование архитектуры системы;</li> <li>4. анализ требований, предъявляемых к программному обеспечению;</li> <li>5. проектирование архитектуры программного обеспечения;</li> <li>6. детальное проектирование программного обеспечения;</li> <li>7. кодирование и тестирование программного обеспечения;</li> <li>8. интеграцию программного обеспечения;</li> <li>9. квалификационное тестирование программного обеспечения;</li> <li>10. интеграцию системы;</li> <li>11. квалификационное тестирование системы;</li> <li>12. установку программного обеспечения;</li> <li>13. приемку программного обеспечения.</li> </ol>	<p>Т.КC27_1</p>
<p>Кейс-задание - "Система управления расписанием школы ":</p>	<p>Т.КC28_</p>

<p>Жизненный цикл программного продукта</p> <p>В соответствии со стандартом ISO/IEC 12207 все процессы ЖЦ ПО разделены на три группы: Основные процессы, вспомогательные процессы, организационные процессы.</p> <p>В группах определено пять основных процессов: приобретение, поставка, разработка, эксплуатация и сопровождение. Восемь вспомогательных процессов обеспечивают выполнение основных процессов, а именно документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместная оценка, аудит, разрешение проблем. Четыре организационных процесса обеспечивают управление, создание инфраструктуры, усовершенствование и обучение. Опишите процесс разрабатываемого программного обеспечения по следующей схеме:</p> <ol style="list-style-type: none"> <li>1. подготовительную работу;</li> <li>2. анализ требований, предъявляемых к системе;</li> <li>3. проектирование архитектуры системы;</li> <li>4. анализ требований, предъявляемых к программному обеспечению;</li> <li>5. проектирование архитектуры программного обеспечения;</li> <li>6. детальное проектирование программного обеспечения;</li> <li>7. кодирование и тестирование программного обеспечения;</li> <li>8. интеграцию программного обеспечения;</li> <li>9. квалификационное тестирование программного обеспечения;</li> <li>10. интеграцию системы;</li> <li>11. квалификационное тестирование системы;</li> <li>12. установку программного обеспечения;</li> <li>13. приемку программного обеспечения.</li> </ol>	1
<p>Кейс-задание - "Система питания детского сада ":</p> <p>Жизненный цикл программного продукта</p> <p>В соответствии со стандартом ISO/IEC 12207 все процессы ЖЦ ПО разделены на три группы: Основные процессы, вспомогательные процессы, организационные процессы.</p> <p>В группах определено пять основных процессов: приобретение, поставка, разработка, эксплуатация и сопровождение. Восемь вспомогательных процессов обеспечивают выполнение основных процессов, а именно документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместная оценка, аудит, разрешение проблем. Четыре организационных процесса обеспечивают управление, создание инфраструктуры, усовершенствование и обучение. Опишите процесс разрабатываемого программного обеспечения по следующей схеме:</p> <ol style="list-style-type: none"> <li>1. подготовительную работу;</li> <li>2. анализ требований, предъявляемых к системе;</li> <li>3. проектирование архитектуры системы;</li> <li>4. анализ требований, предъявляемых к программному обеспечению;</li> <li>5. проектирование архитектуры программного обеспечения;</li> <li>6. детальное проектирование программного обеспечения;</li> <li>7. кодирование и тестирование программного обеспечения;</li> <li>8. интеграцию программного обеспечения;</li> <li>9. квалификационное тестирование программного обеспечения;</li> <li>10. интеграцию системы;</li> <li>11. квалификационное тестирование системы;</li> <li>12. установку программного обеспечения;</li> <li>13. приемку программного обеспечения.</li> </ol>	Т.КС29_1
<p>Кейс-задание - "Система закупки товаров аптеки ":</p> <p>Жизненный цикл программного продукта</p> <p>В соответствии со стандартом ISO/IEC 12207 все процессы ЖЦ ПО разделены на три группы: Основные процессы, вспомогательные процессы, организационные процессы.</p> <p>В группах определено пять основных процессов: приобретение, поставка, разработка, эксплуатация и сопровождение. Восемь вспомогательных процессов обеспечивают выполнение основных процессов, а именно документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместная оценка, аудит, разрешение проблем. Четыре</p>	Т.КС30_1

<p>организационных процесса обеспечивают управление, создание инфраструктуры, усовершенствование и обучение. Опишите процесс разрабатываемого программного обеспечения по следующей схеме:</p> <ol style="list-style-type: none"> <li>1. подготовительную работу;</li> <li>2. анализ требований, предъявляемых к системе;</li> <li>3. проектирование архитектуры системы;</li> <li>4. анализ требований, предъявляемых к программному обеспечению;</li> <li>5. проектирование архитектуры программного обеспечения;</li> <li>6. детальное проектирование программного обеспечения;</li> <li>7. кодирование и тестирование программного обеспечения;</li> <li>8. интеграцию программного обеспечения;</li> <li>9. квалификационное тестирование программного обеспечения;</li> <li>10. интеграцию системы;</li> <li>11. квалификационное тестирование системы;</li> <li>12. установку программного обеспечения;</li> <li>13. приемку программного обеспечения.</li> </ol>	
<p>Кейс-задание - "Система швейного ателье": Жизненный цикл программного продукта В соответствии со стандартом ISO/IEC 12207 все процессы ЖЦ ПО разделены на три группы: Основные процессы, вспомогательные процессы, организационные процессы. В группах определено пять основных процессов: приобретение, поставка, разработка, эксплуатация и сопровождение. Восемь вспомогательных процессов обеспечивают выполнение основных процессов, а именно документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместная оценка, аудит, разрешение проблем. Четыре организационных процесса обеспечивают управление, создание инфраструктуры, усовершенствование и обучение. Опишите процесс разрабатываемого программного обеспечения по следующей схеме:</p> <ol style="list-style-type: none"> <li>1. подготовительную работу;</li> <li>2. анализ требований, предъявляемых к системе;</li> <li>3. проектирование архитектуры системы;</li> <li>4. анализ требований, предъявляемых к программному обеспечению;</li> <li>5. проектирование архитектуры программного обеспечения;</li> <li>6. детальное проектирование программного обеспечения;</li> <li>7. кодирование и тестирование программного обеспечения;</li> <li>8. интеграцию программного обеспечения;</li> <li>9. квалификационное тестирование программного обеспечения;</li> <li>10. интеграцию системы;</li> <li>11. квалификационное тестирование системы;</li> <li>12. установку программного обеспечения;</li> <li>13. приемку программного обеспечения.</li> </ol>	<p>Т.КС31_1</p>
<p>Кейс-задание - "Система газетного киоска": Жизненный цикл программного продукта В соответствии со стандартом ISO/IEC 12207 все процессы ЖЦ ПО разделены на три группы: Основные процессы, вспомогательные процессы, организационные процессы. В группах определено пять основных процессов: приобретение, поставка, разработка, эксплуатация и сопровождение. Восемь вспомогательных процессов обеспечивают выполнение основных процессов, а именно документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместная оценка, аудит, разрешение проблем. Четыре организационных процесса обеспечивают управление, создание инфраструктуры, усовершенствование и обучение. Опишите процесс разрабатываемого программного обеспечения по следующей схеме:</p> <ol style="list-style-type: none"> <li>1. подготовительную работу;</li> <li>2. анализ требований, предъявляемых к системе;</li> <li>3. проектирование архитектуры системы;</li> <li>4. анализ требований, предъявляемых к программному обеспечению;</li> </ol>	<p>Т.КС32_1</p>

<p>5. проектирование архитектуры программного обеспечения;  6. детальное проектирование программного обеспечения;  7. кодирование и тестирование программного обеспечения;  8. интеграцию программного обеспечения;  9. квалификационное тестирование программного обеспечения;  10. интеграцию системы;  11. квалификационное тестирование системы;  12. установку программного обеспечения;  13. приемку программного обеспечения.</p>	
<p>Кейс-задание - "Система оформления подписки на почте":  Жизненный цикл программного продукта  В соответствии со стандартом ISO/IEC 12207 все процессы ЖЦ ПО разделены на три группы: Основные процессы, вспомогательные процессы, организационные процессы.  В группах определено пять основных процессов: приобретение, поставка, разработка, эксплуатация и сопровождение. Восемь вспомогательных процессов обеспечивают выполнение основных процессов, а именно документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместная оценка, аудит, разрешение проблем. Четыре организационных процесса обеспечивают управление, создание инфраструктуры, усовершенствование и обучение. Опишите процесс разрабатываемого программного обеспечения по следующей схеме:</p> <ol style="list-style-type: none"> <li>1. подготовительную работу;</li> <li>2. анализ требований, предъявляемых к системе;</li> <li>3. проектирование архитектуры системы;</li> <li>4. анализ требований, предъявляемых к программному обеспечению;</li> <li>5. проектирование архитектуры программного обеспечения;</li> <li>6. детальное проектирование программного обеспечения;</li> <li>7. кодирование и тестирование программного обеспечения;</li> <li>8. интеграцию программного обеспечения;</li> <li>9. квалификационное тестирование программного обеспечения;</li> <li>10. интеграцию системы;</li> <li>11. квалификационное тестирование системы;</li> <li>12. установку программного обеспечения;</li> <li>13. приемку программного обеспечения.</li> </ol>	<p>Т.КС33_ 1</p>
<p>Кейс-задание - "Система поликлиники":  Применение UML для выполнения этапов анализа и проектирования ПС.</p> <ol style="list-style-type: none"> <li>1. Построить информационную модель (ИМ) информационной системы, для чего: <ol style="list-style-type: none"> <li>1.1. Определить основные объекты (сущности), входящие в систему, и их атрибуты (идентификаторы, привилегированные идентификаторы и описательные атрибуты).</li> <li>1.2. Определить или задать связи между объектами, их типы и условность.</li> <li>1.3. Формализовать связи. С этой целью, в зависимости от типа связи, <ol style="list-style-type: none"> <li>1.3.1. Ввести вспомогательные атрибуты в соответствующие отношения;</li> <li>1.3.2. Добавить ассоциативные объекты в ИМ.</li> </ol> </li> </ol> </li> <li>2. Построить модели состояний каждого объекта с использованием графического представления диаграммы Мура. Для этого: <ol style="list-style-type: none"> <li>2.1. Определить возможные состояния объекта;</li> <li>2.2. Определить события, переводящие объект из одного состояния в другое;</li> <li>2.3. Задать действия, выполняемые объектом в каждом из состояний, в виде укрупненного алгоритма.</li> </ol> </li> <li>3. Построить модель взаимодействия объектов и модель доступа к объектам и проконтролировать корректность порождения и приема событий, для чего, при необходимости построить схему канала управления.</li> <li>4. Построить модели процессов для каждого действия каждого объекта системы в виде диаграммы потоков данных действий, для чего: <ol style="list-style-type: none"> <li>4.1. Определить процессы и их типы –аксессуары, генераторы событий, преобразователи, процессы проверки;</li> </ol> </li> </ol>	<p>Т.КС34_ 1</p>

<p>4.2. Определить условные и безусловные потоки данных и управления.</p> <p>5. Построить диаграмму класса для каждого объекта системы, для чего:</p> <p>5.1. Определить логические компоненты, исходя из атрибутов объектов;</p> <p>5.2. На основе моделей состояний и моделей процессов определить общедоступные операции, разделив операции, базируемые на классах и базируемые на экземплярах;</p> <p>5.3. Задать входы и выходы операций, исходя из данных, переносимых соответствующими событиями.</p> <p>6. Для каждого объекта построить схему структуры класса, основываясь на моделях процессов.</p> <p>7. Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам.</p>	
<p>Кейс-задание - "Система малого предприятия":</p> <p>Применение UML для выполнения этапов анализа и проектирования ПС.</p> <p>1. Построить информационную модель (ИМ) информационной системы, для чего:</p> <p>1.1. Определить основные объекты (сущности), входящие в систему, и их атрибуты (идентификаторы, привилегированные идентификаторы и описательные атрибуты).</p> <p>1.2. Определить или задать связи между объектами, их типы и условность.</p> <p>1.3. Формализовать связи. С этой целью, в зависимости от типа связи,</p> <p>1.3.1. Ввести вспомогательные атрибуты в соответствующие отношения;</p> <p>1.3.2. Добавить ассоциативные объекты в ИМ.</p> <p>2. Построить модели состояний каждого объекта с использованием графического представления диаграммы Мура. Для этого:</p> <p>2.1. Определить возможные состояния объекта;</p> <p>2.2. Определить события, переводящие объект из одного состояния в другое;</p> <p>2.3. Задать действия, выполняемые объектом в каждом из состояний, в виде укрупненного алгоритма.</p> <p>3. Построить модель взаимодействия объектов и модель доступа к объектам и проконтролировать корректность порождения и приема событий, для чего, при необходимости построить схему канала управления.</p> <p>4. Построить модели процессов для каждого действия каждого объекта системы в виде диаграммы потоков данных действий, для чего:</p> <p>4.1. Определить процессы и их типы –аксессуары, генераторы событий, преобразователи, процессы проверки;</p> <p>4.2. Определить условные и безусловные потоки данных и управления.</p> <p>5. Построить диаграмму класса для каждого объекта системы, для чего:</p> <p>5.1. Определить логические компоненты, исходя из атрибутов объектов;</p> <p>5.2. На основе моделей состояний и моделей процессов определить общедоступные операции, разделив операции, базируемые на классах и базируемые на экземплярах;</p> <p>5.3. Задать входы и выходы операций, исходя из данных, переносимых соответствующими событиями.</p> <p>6. Для каждого объекта построить схему структуры класса, основываясь на моделях процессов.</p> <p>7. Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам.</p>	<p>Т.КС35_1</p>
<p>Кейс-задание - "Система организационного управления ":</p> <p>Применение UML для выполнения этапов анализа и проектирования ПС.</p> <p>1. Построить информационную модель (ИМ) информационной системы, для чего:</p> <p>1.1. Определить основные объекты (сущности), входящие в систему, и их атрибуты (идентификаторы, привилегированные идентификаторы и описательные атрибуты).</p> <p>1.2. Определить или задать связи между объектами, их типы и условность.</p> <p>1.3. Формализовать связи. С этой целью, в зависимости от типа связи,</p> <p>1.3.1. Ввести вспомогательные атрибуты в соответствующие отношения;</p> <p>1.3.2. Добавить ассоциативные объекты в ИМ.</p> <p>2. Построить модели состояний каждого объекта с использованием графического представления диаграммы Мура. Для этого:</p>	<p>Т.КС36_1</p>

<p>2.1. Определить возможные состояния объекта;</p> <p>2.2. Определить события, переводящие объект из одного состояния в другое;</p> <p>2.3. Задать действия, выполняемые объектом в каждом из состояний, в виде укрупненного алгоритма.</p> <p>3. Построить модель взаимодействия объектов и модель доступа к объектам и проконтролировать корректность порождения и приема событий, для чего, при необходимости построить схему канала управления.</p> <p>4. Построить модели процессов для каждого действия каждого объекта системы в виде диаграммы потоков данных действий, для чего:</p> <p>4.1. Определить процессы и их типы –аксессуары, генераторы событий, преобразователи, процессы проверки;</p> <p>4.2. Определить условные и безусловные потоки данных и управления.</p> <p>5. Построить диаграмму класса для каждого объекта системы, для чего:</p> <p>5.1. Определить логические компоненты, исходя из атрибутов объектов;</p> <p>5.2. На основе моделей состояний и моделей процессов определить общедоступные операции, разделив операции, базируемые на классах и базируемые на экземплярах;</p> <p>5.3. Задать входы и выходы операций, исходя из данных, переносимых соответствующими событиями.</p> <p>6. Для каждого объекта построить схему структуры класса, основываясь на моделях процессов.</p> <p>7. Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам.</p>	
<p>Кейс-задание - "Система парикмахерской ":</p> <p>Применение UML для выполнения этапов анализа и проектирования ПС.</p> <p>1. Построить информационную модель (ИМ) информационной системы, для чего:</p> <p>1.1. Определить основные объекты (сущности), входящие в систему, и их атрибуты (идентификаторы, привилегированные идентификаторы и описательные атрибуты).</p> <p>1.2. Определить или задать связи между объектами, их типы и условность.</p> <p>1.3. Формализовать связи. С этой целью, в зависимости от типа связи,</p> <p>1.3.1. Ввести вспомогательные атрибуты в соответствующие отношения;</p> <p>1.3.2. Добавить ассоциативные объекты в ИМ.</p> <p>2. Построить модели состояний каждого объекта с использованием графического представления диаграммы Мура. Для этого:</p> <p>2.1. Определить возможные состояния объекта;</p> <p>2.2. Определить события, переводящие объект из одного состояния в другое;</p> <p>2.3. Задать действия, выполняемые объектом в каждом из состояний, в виде укрупненного алгоритма.</p> <p>3. Построить модель взаимодействия объектов и модель доступа к объектам и проконтролировать корректность порождения и приема событий, для чего, при необходимости построить схему канала управления.</p> <p>4. Построить модели процессов для каждого действия каждого объекта системы в виде диаграммы потоков данных действий, для чего:</p> <p>4.1. Определить процессы и их типы –аксессуары, генераторы событий, преобразователи, процессы проверки;</p> <p>4.2. Определить условные и безусловные потоки данных и управления.</p> <p>5. Построить диаграмму класса для каждого объекта системы, для чего:</p> <p>5.1. Определить логические компоненты, исходя из атрибутов объектов;</p> <p>5.2. На основе моделей состояний и моделей процессов определить общедоступные операции, разделив операции, базируемые на классах и базируемые на экземплярах;</p> <p>5.3. Задать входы и выходы операций, исходя из данных, переносимых соответствующими событиями.</p> <p>6. Для каждого объекта построить схему структуры класса, основываясь на моделях процессов.</p> <p>7. Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам.</p>	<p>Т.КС37_1</p>

<p>Кейс-задание - "Система библиотеки ":</p> <p>Применение UML для выполнения этапов анализа и проектирования ПС.</p> <ol style="list-style-type: none"> <li>1. Построить информационную модель (ИМ) информационной системы, для чего: <ol style="list-style-type: none"> <li>1.1. Определить основные объекты (сущности), входящие в систему, и их атрибуты (идентификаторы, привилегированные идентификаторы и описательные атрибуты).</li> <li>1.2. Определить или задать связи между объектами, их типы и условность.</li> <li>1.3. Формализовать связи. С этой целью, в зависимости от типа связи, <ol style="list-style-type: none"> <li>1.3.1. Ввести вспомогательные атрибуты в соответствующие отношения;</li> <li>1.3.2. Добавить ассоциативные объекты в ИМ.</li> </ol> </li> </ol> </li> <li>2. Построить модели состояний каждого объекта с использованием графического представления диаграммы Мура. Для этого: <ol style="list-style-type: none"> <li>2.1. Определить возможные состояния объекта;</li> <li>2.2. Определить события, переводящие объект из одного состояния в другое;</li> <li>2.3. Задать действия, выполняемые объектом в каждом из состояний, в виде укрупненного алгоритма.</li> </ol> </li> <li>3. Построить модель взаимодействия объектов и модель доступа к объектам и проконтролировать корректность порождения и приема событий, для чего, при необходимости построить схему канала управления.</li> <li>4. Построить модели процессов для каждого действия каждого объекта системы в виде диаграммы потоков данных действий, для чего: <ol style="list-style-type: none"> <li>4.1. Определить процессы и их типы –аксессуары, генераторы событий, преобразователи, процессы проверки;</li> <li>4.2. Определить условные и безусловные потоки данных и управления.</li> </ol> </li> <li>5. Построить диаграмму класса для каждого объекта системы, для чего: <ol style="list-style-type: none"> <li>5.1. Определить логические компоненты, исходя из атрибутов объектов;</li> <li>5.2. На основе моделей состояний и моделей процессов определить общедоступные операции, разделив операции, базируемые на классах и базируемые на экземплярах;</li> <li>5.3. Задать входы и выходы операций, исходя из данных, переносимых соответствующими событиями.</li> </ol> </li> <li>6. Для каждого объекта построить схему структуры класса, основываясь на моделях процессов.</li> <li>7. Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам.</li> </ol>	<p>Т.КС38_1</p>
<p>Кейс-задание - "Система управления учебным процессом ":</p> <p>Применение UML для выполнения этапов анализа и проектирования ПС.</p> <ol style="list-style-type: none"> <li>1. Построить информационную модель (ИМ) информационной системы, для чего: <ol style="list-style-type: none"> <li>1.1. Определить основные объекты (сущности), входящие в систему, и их атрибуты (идентификаторы, привилегированные идентификаторы и описательные атрибуты).</li> <li>1.2. Определить или задать связи между объектами, их типы и условность.</li> <li>1.3. Формализовать связи. С этой целью, в зависимости от типа связи, <ol style="list-style-type: none"> <li>1.3.1. Ввести вспомогательные атрибуты в соответствующие отношения;</li> <li>1.3.2. Добавить ассоциативные объекты в ИМ.</li> </ol> </li> </ol> </li> <li>2. Построить модели состояний каждого объекта с использованием графического представления диаграммы Мура. Для этого: <ol style="list-style-type: none"> <li>2.1. Определить возможные состояния объекта;</li> <li>2.2. Определить события, переводящие объект из одного состояния в другое;</li> <li>2.3. Задать действия, выполняемые объектом в каждом из состояний, в виде укрупненного алгоритма.</li> </ol> </li> <li>3. Построить модель взаимодействия объектов и модель доступа к объектам и проконтролировать корректность порождения и приема событий, для чего, при необходимости построить схему канала управления.</li> <li>4. Построить модели процессов для каждого действия каждого объекта системы в виде диаграммы потоков данных действий, для чего: <ol style="list-style-type: none"> <li>4.1. Определить процессы и их типы –аксессуары, генераторы событий, преобразователи, процессы проверки;</li> </ol> </li> </ol>	<p>Т.КС39_1</p>

<p>4.2. Определить условные и безусловные потоки данных и управления.</p> <p>5. Построить диаграмму класса для каждого объекта системы, для чего:</p> <p>5.1. Определить логические компоненты, исходя из атрибутов объектов;</p> <p>5.2. На основе моделей состояний и моделей процессов определить общедоступные операции, разделив операции, базируемые на классах и базируемые на экземплярах;</p> <p>5.3. Задать входы и выходы операций, исходя из данных, переносимых соответствующими событиями.</p> <p>6. Для каждого объекта построить схему структуры класса, основываясь на моделях процессов.</p> <p>7. Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам.</p>	
<p>Кейс-задание - "Система "Клиент-Банк" ": Применение UML для выполнения этапов анализа и проектирования ПС.</p> <p>1. Построить информационную модель (ИМ) информационной системы, для чего:</p> <p>1.1. Определить основные объекты (сущности), входящие в систему, и их атрибуты (идентификаторы, привилегированные идентификаторы и описательные атрибуты).</p> <p>1.2. Определить или задать связи между объектами, их типы и условность.</p> <p>1.3. Формализовать связи. С этой целью, в зависимости от типа связи,</p> <p>1.3.1. Ввести вспомогательные атрибуты в соответствующие отношения;</p> <p>1.3.2. Добавить ассоциативные объекты в ИМ.</p> <p>2. Построить модели состояний каждого объекта с использованием графического представления диаграммы Мура. Для этого:</p> <p>2.1. Определить возможные состояния объекта;</p> <p>2.2. Определить события, переводящие объект из одного состояния в другое;</p> <p>2.3. Задать действия, выполняемые объектом в каждом из состояний, в виде укрупненного алгоритма.</p> <p>3. Построить модель взаимодействия объектов и модель доступа к объектам и проконтролировать корректность порождения и приема событий, для чего, при необходимости построить схему канала управления.</p> <p>4. Построить модели процессов для каждого действия каждого объекта системы в виде диаграммы потоков данных действий, для чего:</p> <p>4.1. Определить процессы и их типы –аксессуары, генераторы событий, преобразователи, процессы проверки;</p> <p>4.2. Определить условные и безусловные потоки данных и управления.</p> <p>5. Построить диаграмму класса для каждого объекта системы, для чего:</p> <p>5.1. Определить логические компоненты, исходя из атрибутов объектов;</p> <p>5.2. На основе моделей состояний и моделей процессов определить общедоступные операции, разделив операции, базируемые на классах и базируемые на экземплярах;</p> <p>5.3. Задать входы и выходы операций, исходя из данных, переносимых соответствующими событиями.</p> <p>6. Для каждого объекта построить схему структуры класса, основываясь на моделях процессов.</p> <p>7. Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам.</p>	<p>Т.КС40_1</p>
<p>Кейс-задание - "Универсальная система "Склад" ": Применение UML для выполнения этапов анализа и проектирования ПС.</p> <p>1. Построить информационную модель (ИМ) информационной системы, для чего:</p> <p>1.1. Определить основные объекты (сущности), входящие в систему, и их атрибуты (идентификаторы, привилегированные идентификаторы и описательные атрибуты).</p> <p>1.2. Определить или задать связи между объектами, их типы и условность.</p> <p>1.3. Формализовать связи. С этой целью, в зависимости от типа связи,</p> <p>1.3.1. Ввести вспомогательные атрибуты в соответствующие отношения;</p> <p>1.3.2. Добавить ассоциативные объекты в ИМ.</p> <p>2. Построить модели состояний каждого объекта с использованием графического представления диаграммы Мура. Для этого:</p>	<p>Т.КС41_1</p>



<p>2.1. Определить возможные состояния объекта;</p> <p>2.2. Определить события, переводящие объект из одного состояния в другое;</p> <p>2.3. Задать действия, выполняемые объектом в каждом из состояний, в виде укрупненного алгоритма.</p> <p>3. Построить модель взаимодействия объектов и модель доступа к объектам и проконтролировать корректность порождения и приема событий, для чего, при необходимости построить схему канала управления.</p> <p>4. Построить модели процессов для каждого действия каждого объекта системы в виде диаграммы потоков данных действий, для чего:</p> <p>4.1. Определить процессы и их типы –аксессуары, генераторы событий, преобразователи, процессы проверки;</p> <p>4.2. Определить условные и безусловные потоки данных и управления.</p> <p>5. Построить диаграмму класса для каждого объекта системы, для чего:</p> <p>5.1. Определить логические компоненты, исходя из атрибутов объектов;</p> <p>5.2. На основе моделей состояний и моделей процессов определить общедоступные операции, разделив операции, базируемые на классах и базируемые на экземплярах;</p> <p>5.3. Задать входы и выходы операций, исходя из данных, переносимых соответствующими событиями.</p> <p>6. Для каждого объекта построить схему структуры класса, основываясь на моделях процессов.</p> <p>7. Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам.</p>	
<p>Кейс-задание - "Система столовой":</p> <p>Применение UML для выполнения этапов анализа и проектирования ПС.</p> <p>1. Построить информационную модель (ИМ) информационной системы, для чего:</p> <p>1.1. Определить основные объекты (сущности), входящие в систему, и их атрибуты (идентификаторы, привилегированные идентификаторы и описательные атрибуты).</p> <p>1.2. Определить или задать связи между объектами, их типы и условность.</p> <p>1.3. Формализовать связи. С этой целью, в зависимости от типа связи,</p> <p>1.3.1. Ввести вспомогательные атрибуты в соответствующие отношения;</p> <p>1.3.2. Добавить ассоциативные объекты в ИМ.</p> <p>2. Построить модели состояний каждого объекта с использованием графического представления диаграммы Мура. Для этого:</p> <p>2.1. Определить возможные состояния объекта;</p> <p>2.2. Определить события, переводящие объект из одного состояния в другое;</p> <p>2.3. Задать действия, выполняемые объектом в каждом из состояний, в виде укрупненного алгоритма.</p> <p>3. Построить модель взаимодействия объектов и модель доступа к объектам и проконтролировать корректность порождения и приема событий, для чего, при необходимости построить схему канала управления.</p> <p>4. Построить модели процессов для каждого действия каждого объекта системы в виде диаграммы потоков данных действий, для чего:</p> <p>4.1. Определить процессы и их типы –аксессуары, генераторы событий, преобразователи, процессы проверки;</p> <p>4.2. Определить условные и безусловные потоки данных и управления.</p> <p>5. Построить диаграмму класса для каждого объекта системы, для чего:</p> <p>5.1. Определить логические компоненты, исходя из атрибутов объектов;</p> <p>5.2. На основе моделей состояний и моделей процессов определить общедоступные операции, разделив операции, базируемые на классах и базируемые на экземплярах;</p> <p>5.3. Задать входы и выходы операций, исходя из данных, переносимых соответствующими событиями.</p> <p>6. Для каждого объекта построить схему структуры класса, основываясь на моделях процессов.</p> <p>7. Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам.</p>	<p>Т.КС42_1</p>

<p>Кейс-задание - "Система СТО ":</p> <p>Применение UML для выполнения этапов анализа и проектирования ПС.</p> <ol style="list-style-type: none"> <li>1. Построить информационную модель (ИМ) информационной системы, для чего: <ol style="list-style-type: none"> <li>1.1. Определить основные объекты (сущности), входящие в систему, и их атрибуты (идентификаторы, привилегированные идентификаторы и описательные атрибуты).</li> <li>1.2. Определить или задать связи между объектами, их типы и условность.</li> <li>1.3. Формализовать связи. С этой целью, в зависимости от типа связи, <ol style="list-style-type: none"> <li>1.3.1. Ввести вспомогательные атрибуты в соответствующие отношения;</li> <li>1.3.2. Добавить ассоциативные объекты в ИМ.</li> </ol> </li> </ol> </li> <li>2. Построить модели состояний каждого объекта с использованием графического представления диаграммы Мура. Для этого: <ol style="list-style-type: none"> <li>2.1. Определить возможные состояния объекта;</li> <li>2.2. Определить события, переводящие объект из одного состояния в другое;</li> <li>2.3. Задать действия, выполняемые объектом в каждом из состояний, в виде укрупненного алгоритма.</li> </ol> </li> <li>3. Построить модель взаимодействия объектов и модель доступа к объектам и проконтролировать корректность порождения и приема событий, для чего, при необходимости построить схему канала управления.</li> <li>4. Построить модели процессов для каждого действия каждого объекта системы в виде диаграммы потоков данных действий, для чего: <ol style="list-style-type: none"> <li>4.1. Определить процессы и их типы –аксессуары, генераторы событий, преобразователи, процессы проверки;</li> <li>4.2. Определить условные и безусловные потоки данных и управления.</li> </ol> </li> <li>5. Построить диаграмму класса для каждого объекта системы, для чего: <ol style="list-style-type: none"> <li>5.1. Определить логические компоненты, исходя из атрибутов объектов;</li> <li>5.2. На основе моделей состояний и моделей процессов определить общедоступные операции, разделив операции, базируемые на классах и базируемые на экземплярах;</li> <li>5.3. Задать входы и выходы операций, исходя из данных, переносимых соответствующими событиями.</li> </ol> </li> <li>6. Для каждого объекта построить схему структуры класса, основываясь на моделях процессов.</li> <li>7. Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам.</li> </ol>	<p>Т.КС43_1</p>
<p>Кейс-задание - "Система автозаправки ":</p> <p>Применение UML для выполнения этапов анализа и проектирования ПС.</p> <ol style="list-style-type: none"> <li>1. Построить информационную модель (ИМ) информационной системы, для чего: <ol style="list-style-type: none"> <li>1.1. Определить основные объекты (сущности), входящие в систему, и их атрибуты (идентификаторы, привилегированные идентификаторы и описательные атрибуты).</li> <li>1.2. Определить или задать связи между объектами, их типы и условность.</li> <li>1.3. Формализовать связи. С этой целью, в зависимости от типа связи, <ol style="list-style-type: none"> <li>1.3.1. Ввести вспомогательные атрибуты в соответствующие отношения;</li> <li>1.3.2. Добавить ассоциативные объекты в ИМ.</li> </ol> </li> </ol> </li> <li>2. Построить модели состояний каждого объекта с использованием графического представления диаграммы Мура. Для этого: <ol style="list-style-type: none"> <li>2.1. Определить возможные состояния объекта;</li> <li>2.2. Определить события, переводящие объект из одного состояния в другое;</li> <li>2.3. Задать действия, выполняемые объектом в каждом из состояний, в виде укрупненного алгоритма.</li> </ol> </li> <li>3. Построить модель взаимодействия объектов и модель доступа к объектам и проконтролировать корректность порождения и приема событий, для чего, при необходимости построить схему канала управления.</li> <li>4. Построить модели процессов для каждого действия каждого объекта системы в виде диаграммы потоков данных действий, для чего: <ol style="list-style-type: none"> <li>4.1. Определить процессы и их типы –аксессуары, генераторы событий, преобразователи, процессы проверки;</li> </ol> </li> </ol>	<p>Т.КС44_1</p>

<p>4.2. Определить условные и безусловные потоки данных и управления.</p> <p>5. Построить диаграмму класса для каждого объекта системы, для чего:</p> <p>5.1. Определить логические компоненты, исходя из атрибутов объектов;</p> <p>5.2. На основе моделей состояний и моделей процессов определить общедоступные операции, разделив операции, базируемые на классах и базируемые на экземплярах;</p> <p>5.3. Задать входы и выходы операций, исходя из данных, переносимых соответствующими событиями.</p> <p>6. Для каждого объекта построить схему структуры класса, основываясь на моделях процессов.</p> <p>7. Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам.</p>	
<p>Кейс-задание - "Система управления расписанием школы ":</p> <p>Применение UML для выполнения этапов анализа и проектирования ПС.</p> <p>1. Построить информационную модель (ИМ) информационной системы, для чего:</p> <p>1.1. Определить основные объекты (сущности), входящие в систему, и их атрибуты (идентификаторы, привилегированные идентификаторы и описательные атрибуты).</p> <p>1.2. Определить или задать связи между объектами, их типы и условность.</p> <p>1.3. Формализовать связи. С этой целью, в зависимости от типа связи,</p> <p>1.3.1. Ввести вспомогательные атрибуты в соответствующие отношения;</p> <p>1.3.2. Добавить ассоциативные объекты в ИМ.</p> <p>2. Построить модели состояний каждого объекта с использованием графического представления диаграммы Мура. Для этого:</p> <p>2.1. Определить возможные состояния объекта;</p> <p>2.2. Определить события, переводящие объект из одного состояния в другое;</p> <p>2.3. Задать действия, выполняемые объектом в каждом из состояний, в виде укрупненного алгоритма.</p> <p>3. Построить модель взаимодействия объектов и модель доступа к объектам и проконтролировать корректность порождения и приема событий, для чего, при необходимости построить схему канала управления.</p> <p>4. Построить модели процессов для каждого действия каждого объекта системы в виде диаграммы потоков данных действий, для чего:</p> <p>4.1. Определить процессы и их типы –аксессуары, генераторы событий, преобразователи, процессы проверки;</p> <p>4.2. Определить условные и безусловные потоки данных и управления.</p> <p>5. Построить диаграмму класса для каждого объекта системы, для чего:</p> <p>5.1. Определить логические компоненты, исходя из атрибутов объектов;</p> <p>5.2. На основе моделей состояний и моделей процессов определить общедоступные операции, разделив операции, базируемые на классах и базируемые на экземплярах;</p> <p>5.3. Задать входы и выходы операций, исходя из данных, переносимых соответствующими событиями.</p> <p>6. Для каждого объекта построить схему структуры класса, основываясь на моделях процессов.</p> <p>7. Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам.</p>	<p>Т.КС45_1</p>
<p>Кейс-задание - "Система питания детского сада":</p> <p>Применение UML для выполнения этапов анализа и проектирования ПС.</p> <p>1. Построить информационную модель (ИМ) информационной системы, для чего:</p> <p>1.1. Определить основные объекты (сущности), входящие в систему, и их атрибуты (идентификаторы, привилегированные идентификаторы и описательные атрибуты).</p> <p>1.2. Определить или задать связи между объектами, их типы и условность.</p> <p>1.3. Формализовать связи. С этой целью, в зависимости от типа связи,</p> <p>1.3.1. Ввести вспомогательные атрибуты в соответствующие отношения;</p> <p>1.3.2. Добавить ассоциативные объекты в ИМ.</p> <p>2. Построить модели состояний каждого объекта с использованием графического представления диаграммы Мура. Для этого:</p>	<p>Т.КС46_1</p>

<p>2.1. Определить возможные состояния объекта;</p> <p>2.2. Определить события, переводящие объект из одного состояния в другое;</p> <p>2.3. Задать действия, выполняемые объектом в каждом из состояний, в виде укрупненного алгоритма.</p> <p>3. Построить модель взаимодействия объектов и модель доступа к объектам и проконтролировать корректность порождения и приема событий, для чего, при необходимости построить схему канала управления.</p> <p>4. Построить модели процессов для каждого действия каждого объекта системы в виде диаграммы потоков данных действий, для чего:</p> <p>4.1. Определить процессы и их типы –аксессуары, генераторы событий, преобразователи, процессы проверки;</p> <p>4.2. Определить условные и безусловные потоки данных и управления.</p> <p>5. Построить диаграмму класса для каждого объекта системы, для чего:</p> <p>5.1. Определить логические компоненты, исходя из атрибутов объектов;</p> <p>5.2. На основе моделей состояний и моделей процессов определить общедоступные операции, разделив операции, базируемые на классах и базируемые на экземплярах;</p> <p>5.3. Задать входы и выходы операций, исходя из данных, переносимых соответствующими событиями.</p> <p>6. Для каждого объекта построить схему структуры класса, основываясь на моделях процессов.</p> <p>7. Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам.</p>	
<p>Кейс-задание - "Система закупки товаров аптеки":</p> <p>Применение UML для выполнения этапов анализа и проектирования ПС.</p> <p>1. Построить информационную модель (ИМ) информационной системы, для чего:</p> <p>1.1. Определить основные объекты (сущности), входящие в систему, и их атрибуты (идентификаторы, привилегированные идентификаторы и описательные атрибуты).</p> <p>1.2. Определить или задать связи между объектами, их типы и условность.</p> <p>1.3. Формализовать связи. С этой целью, в зависимости от типа связи,</p> <p>1.3.1. Ввести вспомогательные атрибуты в соответствующие отношения;</p> <p>1.3.2. Добавить ассоциативные объекты в ИМ.</p> <p>2. Построить модели состояний каждого объекта с использованием графического представления диаграммы Мура. Для этого:</p> <p>2.1. Определить возможные состояния объекта;</p> <p>2.2. Определить события, переводящие объект из одного состояния в другое;</p> <p>2.3. Задать действия, выполняемые объектом в каждом из состояний, в виде укрупненного алгоритма.</p> <p>3. Построить модель взаимодействия объектов и модель доступа к объектам и проконтролировать корректность порождения и приема событий, для чего, при необходимости построить схему канала управления.</p> <p>4. Построить модели процессов для каждого действия каждого объекта системы в виде диаграммы потоков данных действий, для чего:</p> <p>4.1. Определить процессы и их типы –аксессуары, генераторы событий, преобразователи, процессы проверки;</p> <p>4.2. Определить условные и безусловные потоки данных и управления.</p> <p>5. Построить диаграмму класса для каждого объекта системы, для чего:</p> <p>5.1. Определить логические компоненты, исходя из атрибутов объектов;</p> <p>5.2. На основе моделей состояний и моделей процессов определить общедоступные операции, разделив операции, базируемые на классах и базируемые на экземплярах;</p> <p>5.3. Задать входы и выходы операций, исходя из данных, переносимых соответствующими событиями.</p> <p>6. Для каждого объекта построить схему структуры класса, основываясь на моделях процессов.</p> <p>7. Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам.</p>	<p>Т.КС47_1</p>

<p>Кейс-задание - "Система швейного ателье":          Применение UML для выполнения этапов анализа и проектирования ПС.</p> <ol style="list-style-type: none"> <li>1. Построить информационную модель (ИМ) информационной системы, для чего:             <ol style="list-style-type: none"> <li>1.1. Определить основные объекты (сущности), входящие в систему, и их атрибуты (идентификаторы, привилегированные идентификаторы и описательные атрибуты).</li> <li>1.2. Определить или задать связи между объектами, их типы и условность.</li> <li>1.3. Формализовать связи. С этой целью, в зависимости от типа связи,                 <ol style="list-style-type: none"> <li>1.3.1. Ввести вспомогательные атрибуты в соответствующие отношения;</li> <li>1.3.2. Добавить ассоциативные объекты в ИМ.</li> </ol> </li> </ol> </li> <li>2. Построить модели состояний каждого объекта с использованием графического представления диаграммы Мура. Для этого:             <ol style="list-style-type: none"> <li>2.1. Определить возможные состояния объекта;</li> <li>2.2. Определить события, переводящие объект из одного состояния в другое;</li> <li>2.3. Задать действия, выполняемые объектом в каждом из состояний, в виде укрупненного алгоритма.</li> </ol> </li> <li>3. Построить модель взаимодействия объектов и модель доступа к объектам и проконтролировать корректность порождения и приема событий, для чего, при необходимости построить схему канала управления.</li> <li>4. Построить модели процессов для каждого действия каждого объекта системы в виде диаграммы потоков данных действий, для чего:             <ol style="list-style-type: none"> <li>4.1. Определить процессы и их типы –аксессуары, генераторы событий, преобразователи, процессы проверки;</li> <li>4.2. Определить условные и безусловные потоки данных и управления.</li> </ol> </li> <li>5. Построить диаграмму класса для каждого объекта системы, для чего:             <ol style="list-style-type: none"> <li>5.1. Определить логические компоненты, исходя из атрибутов объектов;</li> <li>5.2. На основе моделей состояний и моделей процессов определить общедоступные операции, разделив операции, базируемые на классах и базируемые на экземплярах;</li> <li>5.3. Задать входы и выходы операций, исходя из данных, переносимых соответствующими событиями.</li> </ol> </li> <li>6. Для каждого объекта построить схему структуры класса, основываясь на моделях процессов.</li> <li>7. Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам.</li> </ol>	<p>Т.КС48_1</p>
<p>Кейс-задание - "Система газетного киоска":          Применение UML для выполнения этапов анализа и проектирования ПС.</p> <ol style="list-style-type: none"> <li>1. Построить информационную модель (ИМ) информационной системы, для чего:             <ol style="list-style-type: none"> <li>1.1. Определить основные объекты (сущности), входящие в систему, и их атрибуты (идентификаторы, привилегированные идентификаторы и описательные атрибуты).</li> <li>1.2. Определить или задать связи между объектами, их типы и условность.</li> <li>1.3. Формализовать связи. С этой целью, в зависимости от типа связи,                 <ol style="list-style-type: none"> <li>1.3.1. Ввести вспомогательные атрибуты в соответствующие отношения;</li> <li>1.3.2. Добавить ассоциативные объекты в ИМ.</li> </ol> </li> </ol> </li> <li>2. Построить модели состояний каждого объекта с использованием графического представления диаграммы Мура. Для этого:             <ol style="list-style-type: none"> <li>2.1. Определить возможные состояния объекта;</li> <li>2.2. Определить события, переводящие объект из одного состояния в другое;</li> <li>2.3. Задать действия, выполняемые объектом в каждом из состояний, в виде укрупненного алгоритма.</li> </ol> </li> <li>3. Построить модель взаимодействия объектов и модель доступа к объектам и проконтролировать корректность порождения и приема событий, для чего, при необходимости построить схему канала управления.</li> <li>4. Построить модели процессов для каждого действия каждого объекта системы в виде диаграммы потоков данных действий, для чего:             <ol style="list-style-type: none"> <li>4.1. Определить процессы и их типы –аксессуары, генераторы событий, преобразователи, процессы проверки;</li> </ol> </li> </ol>	<p>Т.КС49_1</p>

<p>4.2. Определить условные и безусловные потоки данных и управления.</p> <p>5. Построить диаграмму класса для каждого объекта системы, для чего:</p> <p>5.1. Определить логические компоненты, исходя из атрибутов объектов;</p> <p>5.2. На основе моделей состояний и моделей процессов определить общедоступные операции, разделив операции, базируемые на классах и базируемые на экземплярах;</p> <p>5.3. Задать входы и выходы операций, исходя из данных, переносимых соответствующими событиями.</p> <p>6. Для каждого объекта построить схему структуры класса, основываясь на моделях процессов.</p> <p>7. Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам.</p>	
--	--

тест по итогам занятия - Раздел 1. Архитектура современных программных систем	Варианты ответов	Номер задания
С каким этапом жизненного цикла программного продукта связано с алгоритмизацией процесса обработки данных, детализацией функций обработки, разработкой структуры ПП, выбором методов и средств создания программ?	<p>1 Документирование</p> <p>2 Программирование</p> <p>3 Сопровождение</p> <p>4 Проектирование</p> <p>5 Нет верного ответа</p>	T.T1_1
С каким этапом жизненного цикла программного продукта связано с технической реализацией проектных решений и выполнение с помощью выбранного инструментария разработчика (алгоритмические языки и системы программирования и т.д.)?	<p>1 Документирование</p> <p>2 Проектирование структуры ПП</p> <p>3 Программирование, тестирование и отладка</p> <p>4 Сопровождение ПП</p> <p>5 Все ответы верны</p>	T.T2_1
На каком этапе жизненного цикла программного продукта составляются необходимые сведения по установке и обеспечению надежной работы ПП и т.д.?	<p>1 Проектирование</p> <p>2 Эксплуатация</p> <p>3 Документирование</p> <p>4 Программирование</p> <p>5 Нет верного объекта</p>	T.T3_1
Жизненный цикл ПО - ...	<p>непрерывный процесс, который начинается с момент его полного изъятия</p> <p>1 из эксплуатации и заканчивается в момент принятия решения о необходимости его создания</p>	T.T4_1

	<p>2 процесс, который начинается с момента его полного описания и заканчивается в момент принятия решения о необходимости его создания</p> <p>3 непрерывный процесс, который начинается с момента принятия решения о необходимости его создания и заканчивается в момент его полного изъятия из эксплуатации</p> <p>4 прерывающийся процесс, который начинается с момента написания структуры программы и заканчивается в момент его полного изъятия из эксплуатации</p> <p>5 Нет верного ответа</p>	
На какие три группы процессов делится структура жизненного цикла ПО по стандарту ISO/IEC 12207?	<p>1 Составные, действующие и вспомогательные процессы</p> <p>2 Основные, дополнительные и остальные процессы</p> <p>3 Вспомогательные, основные и дополнительные процессы</p> <p>4 Основные, вспомогательные и организационные процессы</p> <p>5 Нет верного ответа</p>	T.T5_1
Основные процессы жизненного цикла ПО делятся на ...	<p>1 Процесс документирования, процесс обеспечения качества, процесс верификации</p> <p>2 Процесс поставки, процесс обеспечения качества, процесс верификации</p> <p>3 Процесс управления, процесс создания инфраструктуры, процесс обучения</p> <p>4 Процесс приобретения, процесс поставки, процесс разработки</p> <p>5 Процесс управления, процесс разработки, процесс обучения</p>	T.T6_1
Вспомогательные процессы жизненного цикла ПО делятся на ...	<p>1 Процесс документирования, процесс обеспечения качества, процесс верификации</p>	T.T7_1

	<p>2 Процесс поставки, процесс обеспечения качества, процесс верификации</p> <p>3 Процесс управления, процесс создания инфраструктуры, процесс обучения</p> <p>4 Процесс приобретения, процесс поставки, процесс разработки</p> <p>5 Процесс управления, процесс разработки, процесс обучения</p>	
Организационные процессы жизненного цикла ПО делятся на ...	<p>Процесс управления, процесс создания инфраструктуры, процесс обучения, процесс усовершенствования</p> <p>1</p> <p>Процесс документирования, процесс обеспечения качества, процесс верификации</p> <p>2</p> <p>Процесс приобретения, процесс поставки, процесс разработки</p> <p>3</p> <p>Процесс управления, процесс создания инфраструктуры, процесс документирования</p> <p>4</p> <p>Нет верного ответа</p> <p>5</p>	T.T8_1
Буква «U» в аббревиатуре «UML» означает:	<p>1 United</p> <p>2 Unified</p> <p>3 Universal</p>	T.T9_1
Модель UML состоит из (укажите лишнее):	<p>1 сущностей</p> <p>2 отношений</p> <p>3 множеств</p>	T.T10_1
Сущности UML подразделяются на (укажите лишнее)	<p>1 структурные</p> <p>2 поведенческие</p> <p>3 графические</p> <p>4 группирующие</p> <p>5 аннотационные</p>	T.T11_1



Отношения UML подразделяются на (укажите лишнее)	1      зависимости 2      ассоциации 3      уточнения 4      обобщения 5      реализации	T.T12_1
Структурные сущности UML включают в себя (укажите лишнее)	1      классы 2      узлы 3      пакеты 4      варианты использования 5      интерфейсы	T.T13_1
Поведенческие сущности UML включают в себя (укажите лишнее)	1      состояния 2      деятельности 3      варианты использования 4      интерфейсы	T.T14_1
Сущностями UML являются (укажите лишнее)	1      классы 2      узлы 3      зависимости 4      примечания 5      варианты использования	T.T15_1
Группирующие сущности UML включают в себя	1      классы 2      узлы 3      пакеты 4      примечания	T.T16_1
Отношения зависимости в UML являются	1      симметричными 2      антисимметричными	T.T17_1

	3 транзитивными	
Множество канонических диаграмм UML	1 определяется стандартом языка 2 является соглашением пользователей языка 3 определяется производителями инструментов, поддерживающих UML	T.T18_1
Множество канонических структурных диаграмм UML включает в себя (укажите лишнее)	1 Диаграммы классов 2 Диаграммы использования 3 Диаграммы компонентов 4 Диаграммы объектов	T.T19_1
Канонические диаграммы объектов предназначены для описания	1 поведения 2 использования 3 структуры	T.T20_1

устный опрос / собеседование - Раздел 1. Архитектура современных программных систем	Номер задания
Опишите процедуру Дельфи-метода.	T.Y1_1
Назовите, какие модели берутся за основания декомпозиции.	T.Y2_1
Дайте определение UML.	T.Y3_1
Объясните, из чего исходит функциональная модель.	T.Y4_1
Перечислите типы шкалы.	T.Y5_1
Правила языка UML. Механизмы и расширения языка UML.	T.Y6_1
Основные типы диаграмм в UML. Диаграммы прецедентов. Сценарии.	T.Y7_1
Основные типы диаграмм в UML. Диаграммы взаимодействия и их виды.	T.Y8_1
Основные типы диаграмм в UML. Диаграммы состояний.	T.Y9_1
Основные типы диаграмм в UML. Диаграммы классов.	T.Y10_1
Особенности жизненного цикла с использованием UML.	T.Y11_1
Диаграммы последовательностей. Указание времени на диаграммах.	T.Y12_1
Диаграммы кооперации. Пакеты анализа и проектирования.	T.Y13_1

Расширение языка UML. Стереотип. Ограничения. Тегированные значения.	T.Y14_1
Интерфейсы и классы. Параметризованные классы. Классы проектирования и пакеты.	T.Y15_1
Описание потоков. Виды деятельности и действия. Ветвление и слияние. Распараллеливание и объединение.	T.Y16_1
Диаграмма видов деятельности. Дорожки. Изменение объектов. Процессы, потоки и активные объекты. Диаграммы видов деятельности и пакеты.	T.Y17_1
События. Состояния, переходы, условия. Машины и диаграммы состояний.	T.Y18_1
Составные состояния. Последовательные подсостояния. Запоминающее состояние. Параллельные подсостояния. Диаграммы состояний и пакеты.	T.Y19_1
Взаимодействие групп объектов. Группы, шаблоны, механизмы и контуры. Системы и подсистемы.	T.Y20_1
Компоненты. Диаграммы компонентов. Узлы. Диаграммы развертывания. Физические диаграммы и пакеты.	T.Y21_1
Понятие жизненного цикла программного продукта.	T.Y22_1
Этапы жизненного цикла.	T.Y23_1
Каскадная модель жизненного цикла	T.Y24_1
Спиральная модель жизненного цикла	T.Y25_1
Модель жизненного цикла с контролем	T.Y26_1
Перечислите виды современного программного обеспечения (ПО).	T.Y27_1
Перечислите требования к современному ПО	T.Y28_1
Перечислите характерные черты современного ПО	T.Y29_1
Какие Вы знаете современные платформы для разработки ПО?	T.Y30_1
Назовите наиболее важные качества программных продуктов.	T.Y31_1
Каковы три классических элемента любой технологии программирования.	T.Y32_1
Что такое клиент-серверная архитектура и ее компоненты - клиент и сервер?	T.Y33_1
Что такое тонкий клиент?	T.Y34_1
Что такое полнофункциональный клиент?	T.Y35_1
Что такое слой ПО?	T.Y36_1
Что такое уровень абстракции ПО?	T.Y37_1
Что такое вертикальный срез (слой) ПО?	T.Y38_1
Что такое ПО промежуточного уровня (middleware)?	T.Y39_1
Что такое ярус ПО (tier)?	T.Y40_1
В чем основная идея многоярусной архитектуры ПО?	T.Y41_1
Что такое многоклиентская (multi-tenant) архитектура ПО?	T.Y42_1
Что такое сервисно-ориентированная архитектура?	T.Y43_1

В чем суть концепции Software as a Service (SaaS)?	T.Y44_1
--	---------

<b>доклад / конференция / реферат - Раздел 2. Технологии разработки программного обеспечения</b>	<b>Номер задания</b>
Паттерны проектирования ПО	T.D1_2

<b>кейс - Раздел 2. Технологии разработки программного обеспечения</b>	<b>Номер задания</b>
Разработать программное обеспечение, реализующее следующие паттерны проектирования, с использованием унифицированного языка моделирования UML и объектно-ориентированного языка программирования(C++, C#, Java и др). Порождающий паттерн - AbstractFactory Структурные паттерн - Adapter Паттерн поведения - Chain of Responsibility	T.KC1_2
Разработать программное обеспечение, реализующее следующие паттерны проектирования, с использованием унифицированного языка моделирования UML и объектно-ориентированного языка программирования(C++, C#, Java и др). Порождающий паттерн - Builder Структурные паттерн - Bridge Паттерн поведения - Command	T.KC2_2
Разработать программное обеспечение, реализующее следующие паттерны проектирования, с использованием унифицированного языка моделирования UML и объектно-ориентированного языка программирования(C++, C#, Java и др). Порождающий паттерн - Factory Method Структурные паттерн - Composite Паттерн поведения - Interpreter	T.KC3_2
Разработать программное обеспечение, реализующее следующие паттерны проектирования, с использованием унифицированного языка моделирования UML и объектно-ориентированного языка программирования(C++, C#, Java и др). Порождающий паттерн - Prototype Структурные паттерн - Decorator Паттерн поведения - Iterator	T.KC4_2
Разработать программное обеспечение, реализующее следующие паттерны проектирования, с использованием унифицированного языка моделирования UML и объектно-ориентированного языка программирования(C++, C#, Java и др). Порождающий паттерн - Singleton Структурные паттерн - Facade Паттерн поведения - Mediator	T.KC5_2
Разработать программное обеспечение, реализующее следующие паттерны проектирования, с использованием унифицированного языка моделирования UML и объектно-ориентированного языка программирования(C++, C#, Java и др). Порождающий паттерн - AbstractFactory Структурные паттерн - Flyweight Паттерн поведения - Memento	T.KC6_2
Разработать программное обеспечение, реализующее следующие паттерны проектирования, с использованием унифицированного языка моделирования UML и объектно-ориентированного языка программирования(C++, C#, Java и др). Порождающий паттерн - Builder Структурные паттерн - Proxy Паттерн поведения - Observer	T.KC7_2

<p>Разработать программное обеспечение, реализующего следующие паттерны проектирования, с использованием унифицированного языка моделирования UML и объектно-ориентированного языка программирования(C++, C#, Java и др).</p> <p>Порождающий паттерн - Factory Method</p> <p>Структурные паттерн - Adapter</p> <p>Паттерн поведения - State</p>	T.KC8_2
<p>Разработать программное обеспечение, реализующего следующие паттерны проектирования, с использованием унифицированного языка моделирования UML и объектно-ориентированного языка программирования(C++, C#, Java и др).</p> <p>Порождающий паттерн - Prototype</p> <p>Структурные паттерн - Bridge</p> <p>Паттерн поведения - Strategy</p>	T.KC9_2
<p>Разработать программное обеспечение, реализующего следующие паттерны проектирования, с использованием унифицированного языка моделирования UML и объектно-ориентированного языка программирования(C++, C#, Java и др).</p> <p>Порождающий паттерн - Singleton</p> <p>Структурные паттерн - Composite</p> <p>Паттерн поведения - Template Method</p>	T.KC10_2
<p>Разработать программное обеспечение, реализующего следующие паттерны проектирования, с использованием унифицированного языка моделирования UML и объектно-ориентированного языка программирования(C++, C#, Java и др).</p> <p>Порождающий паттерн - AbstractFactory</p> <p>Структурные паттерн - Decorator</p> <p>Паттерн поведения - Visitor</p>	T.KC11_2
<p>Разработать программное обеспечение, реализующего следующие паттерны проектирования, с использованием унифицированного языка моделирования UML и объектно-ориентированного языка программирования(C++, C#, Java и др).</p> <p>Порождающий паттерн - Builder</p> <p>Структурные паттерн - Facade</p> <p>Паттерн поведения - Chain of Responsibility</p>	T.KC12_2
<p>Разработать программное обеспечение, реализующего следующие паттерны проектирования, с использованием унифицированного языка моделирования UML и объектно-ориентированного языка программирования(C++, C#, Java и др).</p> <p>Порождающий паттерн - Factory Method</p> <p>Структурные паттерн - Flyweight</p> <p>Паттерн поведения - Command</p>	T.KC13_2
<p>Разработать программное обеспечение, реализующего следующие паттерны проектирования, с использованием унифицированного языка моделирования UML и объектно-ориентированного языка программирования(C++, C#, Java и др).</p> <p>Порождающий паттерн - Prototype</p> <p>Структурные паттерн - Proxy</p> <p>Паттерн поведения - Interpreter</p>	T.KC14_2
<p>Разработать программное обеспечение, реализующего следующие паттерны проектирования, с использованием унифицированного языка моделирования UML и объектно-ориентированного языка программирования(C++, C#, Java и др).</p> <p>Порождающий паттерн - Singleton</p> <p>Структурные паттерн - Adapter</p> <p>Паттерн поведения - Iterator</p>	T.KC15_2
<p>Разработать программное обеспечение, реализующего следующие паттерны проектирования, с использованием унифицированного языка моделирования UML и объектно-ориентированного языка программирования(C++, C#, Java и др).</p> <p>Порождающий паттерн - AbstractFactory</p> <p>Структурные паттерн - Bridge</p>	T.KC16_2

Паттерн поведения - Mediator	
Разработать программное обеспечение, реализующего следующие паттерны проектирования, с использованием унифицированного языка моделирования UML и объектно-ориентированного языка программирования(C++, C#, Java и др). Порождающий паттерн - Builder Структурные паттерн - Composite Паттерн поведения - Memento	T.KC17_2
Разработать программное обеспечение, реализующего следующие паттерны проектирования, с использованием унифицированного языка моделирования UML и объектно-ориентированного языка программирования(C++, C#, Java и др). Порождающий паттерн - Factory Method Структурные паттерн - Facade Паттерн поведения - Observer	T.KC18_2
Разработать программное обеспечение, реализующего следующие паттерны проектирования, с использованием унифицированного языка моделирования UML и объектно-ориентированного языка программирования(C++, C#, Java и др). Порождающий паттерн - Prototype Структурные паттерн - Proxy Паттерн поведения - State	T.KC19_2
Разработать программное обеспечение, реализующего следующие паттерны проектирования, с использованием унифицированного языка моделирования UML и объектно-ориентированного языка программирования(C++, C#, Java и др). Порождающий паттерн - AbstractFactory Структурные паттерн - Proxy Паттерн поведения - Strategy	T.KC20_2
Разработать программное обеспечение, реализующего следующие паттерны проектирования, с использованием унифицированного языка моделирования UML и объектно-ориентированного языка программирования(C++, C#, Java и др). Порождающий паттерн - Builder Структурные паттерн - Flyweight Паттерн поведения - Template Method	T.KC21_2
Разработать программное обеспечение, реализующего следующие паттерны проектирования, с использованием унифицированного языка моделирования UML и объектно-ориентированного языка программирования(C++, C#, Java и др). Порождающий паттерн - Factory Method Структурные паттерн - Facade Паттерн поведения - Visitor	T.KC22_2
Разработать программное обеспечение, реализующего следующие паттерны проектирования, с использованием унифицированного языка моделирования UML и объектно-ориентированного языка программирования(C++, C#, Java и др). Порождающий паттерн - Prototype Структурные паттерн - Decorator Паттерн поведения - Mediator	T.KC23_2
Разработать программное обеспечение, реализующего следующие паттерны проектирования, с использованием унифицированного языка моделирования UML и объектно-ориентированного языка программирования(C++, C#, Java и др). Порождающий паттерн - Singleton Структурные паттерн - Composite Паттерн поведения - Observer	T.KC24_2

<b>тест по итогам занятия - Раздел 2. Технологии разработки программного обеспечения</b>	<b>Варианты ответов</b>	<b>Номер задания</b>
--	-------------------------	----------------------

Класс - это:	<ol style="list-style-type: none"> <li>любой тип данных, определяемый пользователем</li> <li>тип данных, определяемый пользователем и сочетающий в себе данные и функции их обработки</li> <li>структура, для которой в программе имеются функции работы с нею</li> </ol>	T.T1_2
Членами класса могут быть:	<ol style="list-style-type: none"> <li>как переменные, так и функции, могут быть объявлены как private и как public</li> <li>только переменные, объявленные как private</li> <li>только функции, объявленные как private</li> <li>только переменные и функции, объявленные как private</li> <li>только переменные и функции, объявленные как public</li> </ol>	T.T2_2
Что называется конструктором?	<ol style="list-style-type: none"> <li>метод, имя которого совпадает с именем класса и который вызывается автоматически при создании объекта класса</li> <li>метод, имя которого совпадает с именем класса и который вызывается автоматически при объявлении класса (до создания объекта класса)</li> <li>метод, имя которого необязательно совпадает с именем класса и который вызывается при создании объекта класса</li> <li>метод, имя которого совпадает с именем класса и который необходимо явно вызывать из головной программы при объявлении объекта класса</li> </ol>	T.T3_2
Объект - это	<ol style="list-style-type: none"> <li>переменная, содержащая указатель на класс</li> <li>экземпляр класса</li> <li>класс, который содержит в себе данные и методы их обработки</li> </ol>	T.T4_2

Отметьте правильные утверждения	<p>1 конструкторы класса не наследуются</p> <p>конструкторов класса может быть</p> <p>2 несколько, их синтаксис определяется программистом</p> <p>конструкторов класса может быть</p> <p>3 несколько, но их синтаксис должен подчиняться правилам перегрузки функций</p> <p>4 конструктор возвращает указатель на объект</p> <p>5 конструктор не возвращает значение</p>	T.T5_2
Что называется деструктором?	<p>1 метод, который уничтожает объект</p> <p>2 метод, который удаляет объект</p> <p>3 метод, который освобождает память, занимаемую объектом</p> <p>4 системная функция, которая освобождает память, занимаемую объектом</p>	T.T6_2
Выберите правильные утверждения	<p>1 у конструктора могут быть параметры</p> <p>2 конструктор наследуется, но должен быть перегружен</p> <p>3 конструктор должен явно вызываться всегда перед объявлением объекта</p> <p>4 конструктор вызывается автоматически при объявлении объекта</p> <p>5 объявление каждого класса должно содержать свой конструктор</p> <p>6 если конструктор не создан, компилятор создаст его автоматически</p>	T.T7_2
Выберите правильные утверждения:	<p>1 деструктор - это метод класса, применяемый для удаления объекта</p> <p>деструктор - это метод класса,</p> <p>2 применяемый для освобождения памяти, занимаемой объектом</p> <p>деструктор - это отдельная функция</p> <p>3 головной программы, применяемая для освобождения памяти, занимаемой</p>	T.T8_2



	<p>объектом</p> <p>4 деструктор не наследуется</p> <p>5 деструктор наследуется, но должен быть перегружен</p>	
Что называется наследованием	<p>это механизм, посредством которого</p> <p>1 производный класс получает элементы родительского и может дополнять либо изменять их свойства и методы</p> <p>2 это механизм переопределения методов базового класса</p> <p>это механизм, посредством которого</p> <p>3 производный класс получает все поля базового класса</p> <p>это механизм, посредством которого</p> <p>4 производный класс получает элементы родительского, может их дополнить, но не может переопределить</p>	T.T9_2
Выберите правильное объявление производного класса	<p>1 <code>class MoreDetails:: Details</code></p> <p>2 <code>class MoreDetails: public class Details</code></p> <p>3 <code>class MoreDetails: public Details</code></p> <p>4 <code>class MoreDetails: class(Details)</code></p>	T.T10_2
Выберите верные утверждения	<p>если элементы класса объявлены как <code>private</code>, то они доступны только наследникам класса, но не внешним функциям</p> <p>1</p> <p>если элементы класса объявлены как <code>private</code>, то они недоступны ни наследникам класса, ни внешним функциям</p> <p>2</p> <p>если элементы объявлены как <code>public</code>, то они доступны наследникам класса, но не внешним функциям</p> <p>3</p> <p>если элементы объявлены как <code>public</code>, то они доступны и наследникам класса, и внешним функциям</p> <p>4</p>	T.T11_2
Возможность и способ обращения производного класса к элементам базового определяется	<p>1 ключами доступа: <code>private</code>, <code>public</code>,</p>	T.T12_2

	<p>protected в теле производного класса</p> <p>только ключом доступа protected в заголовке объявления производного класса</p> <p>ключами доступа: private, public, protected в заголовке объявления производного класса</p> <p>ключами доступа: private, public, protected в теле базового класса</p>	
<p>Выберите правильные соответствия между спецификатором базового класса, ключом доступа в объявлении производного класса и правами доступа производного класса к элементам базового</p>	<p>ключ доступа - public; в базовом классе: private; права доступа в производном классе - protected</p> <p>ключ доступа - любой; в базовом классе: private; права доступа в производном классе - нет прав</p> <p>ключ доступа - private; в базовом классе: public; права доступа в производном классе - public</p> <p>ключ доступа – любой; в базовом классе: public; права доступа в производном классе – такие же, как ключ доступа</p>	T.T13_2
<p>Дружественная функция - это</p>	<p>функция другого класса, среди аргументов которой есть элементы данного класса</p> <p>функция, объявленная в классе с атрибутом friend, но не являющаяся членом класса</p> <p>функция, являющаяся членом класса и объявленная с атрибутом friend</p> <p>функция, являющаяся членом класса и объявленная с атрибутом friend</p>	T.T14_2
<p>Выберите правильные утверждения про дружественные функции:</p>	<p>одна функция может быть дружественной нескольким классам</p> <p>дружественная функция не может быть обычной функцией, а только методом другого класса</p> <p>дружественная функция объявляется внутри класса, к элементам которого ей нужен доступ</p>	T.T15_2

	4 дружественная функция не может быть методом другого класса	
Шаблон функции - это...	<p>определение функции, в которой типу обрабатываемых данных присвоено условное обозначение</p> <p>2 прототип функции, в котором вместо имен параметров указан условный тип</p> <p>определение функции, в котором указаны возможные варианты типов обрабатываемых параметров</p> <p>3</p> <p>определение функции, в котором в прототипе указан условный тип, а в определении указаны варианты типов обрабатываемых параметров</p> <p>4</p>	T.T16_2
Выберите правильные ответы:	<p>1 по умолчанию члены класса имеют атрибут private</p> <p>2 по умолчанию члены класса имеют атрибут public</p> <p>3 члены класса имеют доступ только к элементам public</p> <p>4 элементы класса с атрибутом private доступны только членам класса</p>	T.T17_2
Переопределение операций имеет вид	<p>1 имя_класса, ключевое слово operation, символ операции</p> <p>имя_класса, ключевое слово operator, символ операции, в круглых скобках могут быть указаны аргументы</p> <p>2</p> <p>имя_класса, ключевое слово operator, список аргументов</p> <p>3</p> <p>имя_класса, два двоеточия, ключевое слово operator, символ операции</p> <p>4</p>	T.T18_2
Для доступа к элементам объекта используются	<p>при обращении через имя объекта – точка, при обращении через указатель – операция «-&gt;»</p> <p>1</p> <p>при обращении через имя объекта – два двоеточия, при обращении через</p> <p>2</p>	T.T19_2

	<p>указатель – операция «точка»</p> <p>при обращении через имя объекта – точка, при обращении через указатель – два двоеточия</p> <p>при обращении через имя объекта – два двоеточия, при обращении через указатель – операция «-&gt;»</p>	
Полиморфизм – это	<p>средство, позволяющее использовать одно имя для обозначения действий, общих для родственных классов</p> <p>средство, позволяющее в одном классе использовать методы с одинаковыми именами</p> <p>средство, позволяющее в одном классе использовать методы с разными именами для выполнения одинаковых действий</p> <p>средство, позволяющее перегружать функции для работы с разными типами или разным количеством аргументов</p>	T.T20_2
Полиморфизм реализован через механизмы	<p>перегрузки функций, виртуальных функций, шаблонов</p> <p>перегрузки функций, наследования методов, шаблонов</p> <p>наследования методов, виртуальных функций, шаблонов</p> <p>перегрузки функций, наследования, виртуальных функций</p>	T.T21_2

устный опрос / собеседование - Раздел 2. Технологии разработки программного обеспечения	Номер задания
Объектно-ориентированный подход (ООП).	T.Y1_2
Основные понятия ООП.	T.Y2_2
Определение ООП и его основные концепции.	T.Y3_2
Недостатки ООП.	T.Y4_2

## Контрольные задания. Промежуточная аттестация

Зачет. Практическое задание	Номер задания
Реализуйте WCF-сервис в среде VisualStudio в рамках тематики "Система поликлиники "	П.П1
Реализуйте WCF-сервис в среде VisualStudio в рамках тематики "Система малого предприятия "	П.П2
Реализуйте WCF-сервис в среде VisualStudio в рамках тематики "Система организационного управления "	П.П3
Реализуйте WCF-сервис в среде VisualStudio в рамках тематики "Система парикмахерской "	П.П4
Реализуйте WCF-сервис в среде VisualStudio в рамках тематики "Система библиотеки "	П.П5
Реализуйте WCF-сервис в среде VisualStudio в рамках тематики "Система управления учебным процессом "	П.П6
Реализуйте WCF-сервис в среде VisualStudio в рамках тематики "Система "Клиент-Банк" "	П.П7
Реализуйте WCF-сервис в среде VisualStudio в рамках тематики "Универсальная система "Склад" "	П.П8
Реализуйте WCF-сервис в среде VisualStudio в рамках тематики "Система столовой "	П.П9
Реализуйте WCF-сервис в среде VisualStudio в рамках тематики "Система СТО "	П.П10
Реализуйте WCF-сервис в среде VisualStudio в рамках тематики "Система автозаправки "	П.П11
Реализуйте WCF-сервис в среде VisualStudio в рамках тематики "Система управления расписанием школы "	П.П12
Реализуйте WCF-сервис в среде VisualStudio в рамках тематики "Система закупки товаров аптеки "	П.П13
Реализуйте WCF-сервис в среде VisualStudio в рамках тематики "Система швейного ателье "	П.П14
Реализуйте WCF-сервис в среде VisualStudio в рамках тематики "Система газетного киоска"	П.П15
Реализуйте WCF-сервис в среде VisualStudio в рамках тематики "Система оформления подписки на почте"	П.П16
Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам в рамках тематики "Система поликлиники".	П.П17
Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам в рамках тематики "Система малого предприятия ".	П.П18
Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам в рамках тематики "Система организационного управления ".	П.П19
Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам в рамках тематики "Система парикмахерской".	П.П20
Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам в рамках тематики "Система библиотеки ".	П.П21
Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной	П.П22

модели, модели взаимодействия объектов и модели доступа к объектам в рамках тематики "Система управления учебным процессом ".	
Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам в рамках тематики "Система "Клиент-Банк" ".	П.П23
Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам в рамках тематики "Универсальная система "Склад" ".	П.П24
Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам в рамках тематики "Система столовой ".	П.П25
Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам в рамках тематики "Система СТО ".	П.П26
Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам в рамках тематики "Система управления расписанием школы".	П.П27
Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам в рамках тематики "Система питания детского сада".	П.П28
Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам в рамках тематики "Система закупки товаров аптеки".	П.П29
Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам в рамках тематики "Система швейного ателье".	П.П30
Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам в рамках тематики "Система газетного киоска".	П.П31
Построить диаграмму наследования и диаграмму зависимостей, исходя из информационной модели, модели взаимодействия объектов и модели доступа к объектам в рамках тематики "Система оформления подписки на почте".	П.П32

<b>Зачет. Теоретический вопрос</b>	<b>Номер задания</b>
Классы задач, решаемых современным программным обеспечением.	П.ТВ1
Характеристики и свойства современного ПО.	П.ТВ2
Определения основных терминов, связанных с видами архитектур программного обеспечения.	П.ТВ3
Требования к современному программному обеспечению.	П.ТВ4
ИКТ и вычислительное оборудование как инструментарий автоматизации и информатизации прикладных задач.	П.ТВ5
Клиент-серверная архитектура и ее компоненты - клиент и сервер.	П.ТВ6
Современные платформы для разработки ПО.	П.ТВ7
Понятие жизненного цикла программного продукта(ПП).	П.ТВ8
Этапы жизненного цикла ПП.	П.ТВ9

Международный стандарт ISO/IEC 12207.	П.ТВ10
Модели жизненного цикла(каскадная, с промежуточным контролем, спиральная).	П.ТВ11
Преимущества и недостатки моделей жизненного цикла, области их применения.	П.ТВ12
Основные подходы к разработке программного обеспечения.	П.ТВ13
Структурный подход к разработке программного обеспечения, основанный на принципе функциональной декомпозиции.	П.ТВ14
Объектно-ориентированный подход к разработке программного обеспечения, основанный на объектной декомпозиции.	П.ТВ15
Принципы объектно-ориентированного представления программных систем(основные-абстрагирование, абстрагирование, инкапсуляция, модульность, иерархия; дополнительные – типизация, параллелизм, устойчивость).	П.ТВ16
Основные понятия объектно-ориентированного подхода(объект, класс, данные, методы, доступ, наследование свойств, системы объектов и классов)	П.ТВ17
Определение объектно-ориентированного программирования как метода построения программ в виде множества взаимодействующих объектов.	П.ТВ18
Определение объектно-ориентированного языка программирования.	П.ТВ19
Деление объектных языков на языки, использующие объекты, и объектно-ориентированные языки.	П.ТВ20
История развития объектно-ориентированных языков программирования.	П.ТВ21
Назначение и история языка UML.	П.ТВ22
Структура языка UML (сущности, отношения, диаграммы).	П.ТВ23
Основные сущности языка UML.	П.ТВ24
Отношения языка UML.	П.ТВ25
Диаграммы языка UML. Их назначение, структура, правила построения.	П.ТВ26
Представление архитектуры программных систем (ее видов) диаграммами языка UML.	П.ТВ27
Моделирование программных систем (ее видов) диаграммами UML.	П.ТВ28
CASE-средства построения UML-диаграмм.	П.ТВ29
Унифицированный процесс разработки программного обеспечения, его базовые принципы.	П.ТВ30
Последовательность действий при анализе: составление и документирование функциональных и нефункциональных требований.	П.ТВ31
Проектирование программного обеспечения, переход от диаграмм анализа к диаграммам проектирования.	П.ТВ32
Выполнение этапов реализации и проверки программного обеспечения.	П.ТВ33
Паттерны проектирования.	П.ТВ34
Понятие тестирования, верификации, валидации.	П.ТВ35
Организация процесса тестирования программного обеспечения.	П.ТВ36
Тестирование объектно-ориентированных программных систем.	П.ТВ37
Тенденции развития технологии разработки ПО.	П.ТВ38

Паттерны проектирования и их представление в нотации UML.	П.ТВ39
Архитектурные паттерны. Примеры	П.ТВ40
Паттерны проектирования. Примеры	П.ТВ41
Паттерны анализа. Примеры	П.ТВ42
Паттерны тестирования. Примеры	П.ТВ43
Паттерны реализации. Примеры	П.ТВ44
Основы ООП-методологии.	П.ТВ45
Концепция MVC	П.ТВ46
Основные понятия MVC.	П.ТВ47
Определение MVC и его основные концепции.	П.ТВ48
Достоинства и недостатки MVC	П.ТВ49
Назначение MVC.	П.ТВ50

#### Балльная система оценивания по дисциплине

ОФО

Семестр (Курс) - 2 (1)			
Форма текущего контроля	Раздел дисциплины	Максимальный балл	Максимальный приведенный балл
доклад / конференция / реферат	Раздел 1. Архитектура современных программных систем	10	
доклад / конференция / реферат	Раздел 2. Технологии разработки программного обеспечения	10	
кейс	Раздел 1. Архитектура современных программных систем	10	
кейс	Раздел 2. Технологии разработки программного обеспечения	10	
тест по итогам занятия	Раздел 1. Архитектура современных программных систем	10	
тест по итогам занятия	Раздел 2. Технологии разработки программного	10	



	обеспечения		
устный опрос / собеседование	Раздел 1. Архитектура современных программных систем	10	
устный опрос / собеседование	Раздел 2. Технологии разработки программного обеспечения	10	
Максимальный текущий балл		80	80
<b>Промежуточная аттестация</b>		зачет	
Максимальный аттестационный балл		20	20
Критерии оценивания		<p>18-20 баллов: Обучающийся, достигающий должного уровня:</p> <ul style="list-style-type: none"> <li>- даёт полный, глубокий, выстроенный логично по содержанию вопроса ответ, используя различные источники информации, не требующий дополнений</li> <li>- доказательно иллюстрирует основные теоретические положения практическими примерами;</li> <li>- способен глубоко анализировать теоретический и практический материал, обобщать его, самостоятельно делать выводы, вести диалог и высказывать свою точку зрения.</li> </ul> <p>14-17 баллов: Обучающийся на должном уровне:</p> <ul style="list-style-type: none"> <li>- раскрывает учебный материал: даёт содержательно полный ответ, требующий незначительных дополнений и уточнений, которые он может сделать самостоятельно после наводящих вопросов преподавателя;</li> <li>- демонстрирует учебные умения и навыки в области решения практико-ориентированных задач;</li> <li>- владеет способами анализа, сравнения, обобщения и обоснования выбора методов решения практико-ориентированных задач.</li> </ul> <p>11-13 баллов: Достигнутый уровень оценки результатов обучения обучающегося показывает:</p> <ul style="list-style-type: none"> <li>- знания имеют фрагментарный характер, отличаются поверхностностью и малой содержательностью; студент раскрывает содержание вопроса, но не глубоко, бессистемно, с некоторыми неточностями;</li> <li>- слабо, недостаточно аргументированно может обосновать связь теории с практикой;</li> <li>- способен понимать и интерпретировать основной теоретический материал по дисциплине.</li> </ul> <p>0-10 баллов: Результаты обучения обучающегося свидетельствуют:</p> <ul style="list-style-type: none"> <li>- об усвоении им некоторых элементарных знаний, но студент не владеет понятийным аппаратом изучаемой образовательной области (учебной дисциплины);</li> <li>- не умеет установить связь теории с практикой;</li> <li>- не владеет способами решения практико-ориентированных</li> </ul>	

	задач.	
Общий балл по дисциплине	100	100

Общий балл по дисциплине за семестр складывается из результатов, полученных по формам текущего контроля в течение семестра и аттестационного балла.

Оценка успеваемости по дисциплине в семестре пересчитывается по приведенной 100-балльной шкале независимо от шкалы, определенной преподавателем.

Перевод баллов из 100-балльной шкалы в числовой и буквенный эквивалент:

**- для зачета:**

Сумма баллов	Отметка
51-100	Зачтено
0-50	Не зачтено

### Список используемых сокращений

#### Текущая аттестация

Тип задания	Сокращение
внеаудиторное чтение	Т.В
доклад / конференция / реферат	Т.Д
индивидуальное задание (перевод / презентация / план урока / тезаурус / глоссарий / сценарий деловой игры / алгоритм задачи / программа / конспектирование научной литературы)	Т.И
итоговая лабораторная работа	Т.ЛР
кейс	Т.КС
коллоквиум	Т.К
контрольная работа	Т.КР
лабораторная работа	Т.Л
отчет (по научно-исследовательской работе / практике)	Т.О
письменная работа	Т.ПР
практическая работа	Т.П
расчетно-графическая работа	Т.РГ
семестровая работа	Т.СР
ситуационная задача / ситуационное задание / проект	Т.СЗ
творческая работа	Т.ТР
тест по итогам занятия	Т.Т

устный опрос / собеседование	Т.У
эссе	Т.Э

Промежуточная аттестация

Тип задания	Сокращение
Практическое задание	П.П
Теоретический вопрос	П.ТВ
Тестовый вопрос	П.Т