



Федеральное государственное бюджетное образовательное учреждение высшего образования
«ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ ДВАЖДЫ ГЕРОЯ
СОВЕТСКОГО СОЮЗА, ЛЕТЧИКА-КОСМОНАВТА А.А. ЛЕОНОВА»

«УТВЕРЖДАЮ»

И.о. проректора

А.В. Троицкий

«__» _____ 2023г.

**ИНСТИТУТ
ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ
КАФЕДРА ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»

Направление подготовки: 09.03.04 Программная инженерия

Направленность (профиль): Проектирование и разработка программного обеспечения

Уровень высшего образования: бакалавриат

Форма обучения: очная

Королев 2023

Рабочая программа является составной частью основной профессиональной образовательной программы и проходит рецензирование со стороны работодателей в составе основной профессиональной образовательной программы. Рабочая программа актуализируется и корректируется ежегодно.

Автор: Баранова О.М. **Рабочая программа дисциплины «Проектирование программного обеспечения».** – Королев МО: Технологический университет, 2024 г.

Рецензент: к.т.н. Парафейников И.В.

Рабочая программа составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования (ФГОС ВО) по направлению подготовки бакалавров 09.03.04 Программная инженерия, учебного плана, утвержденного Ученым советом Технологического университета, Протокол № 9 от 11 апреля 2024 года.

Рабочая программа рассмотрена и одобрена на заседании кафедры:

Заведующий кафедрой (ФИО, ученая степень, звание, подпись)	Баранова О.М., к.т.н. 			
Год утверждения (переутверждения)	2024	2024	2025	2026
Номер и дата протокола заседания кафедры	№ 7 от 03.04.2023			

Рабочая программа согласована:

Руководитель ОПОП  О.М. Баранова, к.т.н.

Рабочая программа рекомендована на заседании УМС:

Год утверждения (переутверждения)	2024	2024	2025	2026
Номер и дата протокола заседания УМС	№ 5 от 11.04.2024 г.			

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения ОПОП

Целью изучения дисциплины является:

- получение знаний и умений в области индустриальных способов (инженерных методов и средств) проектирования программного обеспечения (ПО);
- получение практических навыков работы с инструментальными средствами проектирования программного обеспечения.

Дисциплина направлена на формирование следующих компетенций:

- **(ПК-1)** - Способен определять и анализировать требования к программному обеспечению и возможность их реализации, в том числе разрабатывать технические спецификации на программное обеспечение и его компоненты;
- **(ПК-2)** - Способен проектировать программное обеспечение, в том числе разрабатывать проектную документацию;
- **(ПК-5)** - Способен выполнять работы по созданию, модификации и сопровождению интеграционных решений;
- **(ПК-6)** - Способен управлять работами по созданию и модификации информационных ресурсов;
- **(ПК-7)** - Способен анализировать качество программного обеспечения и определять методы его повышения.

Основными задачами дисциплины являются:

- изучение методов анализа и проектирования программного обеспечения;
- получение навыков применения современных CASE - средств проектирования программного обеспечения.

Показатель освоения компетенции отражают следующие индикаторы:

Трудовые действия:

- Определяет требования к программному обеспечению и возможность их реализации;
- Разрабатывает технические спецификации на программное обеспечение и его компоненты;
- Проектирует программное обеспечение;
- Осуществляет поддержку процесса согласования требований к интеграционному решению;
- Конфигурирует интеграционное решение;
- Анализирует требования к информационным ресурсам, разрабатывает технические спецификации на информационные ресурсы;
- Проектирует, разрабатывает и тестирует информационные ресурсы;

- Производит анализ качества программного обеспечения;
- Определяет и применяет методы повышения качества программного обеспечения;

Необходимые умения:

- Анализирует требования к программному обеспечению;
- Определяет состав информации для разработки технических спецификаций на программное обеспечение и его компоненты;
- Применяет различные способы и методы проектирования программного обеспечения;
- Разрабатывает модели информационных процессов;
- Разрабатывает проектную документацию;
- Знает основные принципы, способы и методы проектирования программного обеспечения;
- Знает состав проектной документации и основные принципы ее разработки;
- Применяет принципы определения требований к интеграционному решению;
- Использует основные методы, способы и средства интеграции программных продуктов;
- Применяет основные принципы разработки технических спецификаций на информационные ресурсы;
- Разрабатывает модели информационных ресурсов;
- Проводит тестирование информационных ресурсов;
- Использует различные методы и способы анализа качества и повышения его уровня.

Необходимые знания:

- Знает принципы определения требований к программному обеспечению;
- Знает принципы и правила разработки технических спецификаций на программное обеспечение и его компоненты;
- Знает основные принципы, способы и методы проектирования программного обеспечения;
- Знает состав проектной документации и основные принципы ее разработки;
- Знает принципы определения требований к интеграционному решению;
- Знает основные методы, способы и средства интеграции программных продуктов;
- Знает основные принципы разработки технических спецификаций на информационные ресурсы;
- Знает основные принципы тестирования информационных ресурсов;
- Знает концепции и атрибуты качества;

- Знает методы и способы анализа качества и повышения его уровня.

2. Место дисциплины (модуля) в структуре ОПОП ВО

Дисциплина «Проектирование программного обеспечения» относится к части, формируемой участниками образовательных отношений, основной профессиональной образовательной программы подготовки по направлению подготовки бакалавриата 09.03.04 «Программная инженерия».

Дисциплина реализуется кафедрой прикладного программного обеспечения.

Изучение данной дисциплины базируется на знаниях и умениях, полученных в рамках ранее изученных дисциплин «Основы системотехники», «Архитектура программного обеспечения», «Основы системной инженерии» и компетенциях УК-1, УК-2, ПК-3, ПК-4, ПК-5.

3. Объем дисциплины (модуля) и виды учебной работы

Общая трудоемкость дисциплины составляет 3 зачетные единицы, 108 часов.

Таблица 1

Виды занятий	Всего часов	Семестр 5	Семестр	Семестр	Семестр
Общая трудоемкость	108	108			
ОЧНАЯ ФОРМА ОБУЧЕНИЯ					
Аудиторные занятия	48	48			
Лекции (Л)	16	16			
Практические занятия (ПЗ)	32	32			
Лабораторные работы (ЛР)	-	-			
Практическая подготовка	32	32			
Самостоятельная работа	60	60			
Курсовые работы (проекты)	-	-			
Расчетно-графические работы	-	-			
Контрольная работа, домашнее задание	+ -	+ -			
Текущий контроль знаний (7 - 8, 15 - 16 недели)	тест	тест			
Вид итогового контроля	Зачет с оценкой	Зачет с оценкой			

4. Содержание дисциплины

4.1. Темы дисциплины и виды занятий

Таблица 2

Наименование тем	Лекции, час.	Практические занятия, час.	Занятия в интерактивной форме, час.	Код компетенций
Тема 1. Жизненный цикл программного обеспечения	4	4	2	ПК-1, ПК-2, ПК-5, ПК-6, ПК-7
Тема 2. Анализ требований к программному обеспечению	2	4	2	
Тема 3. Структурные (функциональные) методы проектирования программного обеспечения	2	6	2	
Тема 4. Объектно-ориентированные методы проектирования программного обеспечения	4	10	3	
Тема 5. Паттерны проектирования программного обеспечения	4	8	3	
Итого:	16	32	12	

4.2. Содержание тем дисциплины

Тема 1. Жизненный цикл программного обеспечения

Понятие жизненного цикла программного обеспечения. Этапы жизненного цикла программного обеспечения. Место проектирования среди этапов жизненного цикла программного обеспечения. Модели жизненного цикла. Гибкие методологии разработки программного обеспечения.

Тема 2. Анализ требований к программному обеспечению

Классификация требований к программному обеспечению. Анализ требований к программному обеспечению. Документирование требований к программному обеспечению. Разработка технического задания.

Тема 3. Структурные (функциональные) методы проектирования программного обеспечения

Классификация методов проектирования программного обеспечения. Структурные методы проектирования программного обеспечения. Семейство методов проектирования IDEF.

Тема 4. Объектно-ориентированные методы проектирования программного обеспечения

Объектно-ориентированное проектирование программного обеспечения. Унифицированный язык моделирования UML. Основные понятия унифицированного языка моделирования UML. Виды диаграмм UML.

Тема 5. Паттерны проектирования программного обеспечения

Структурные паттерны. Рекурсивная композиция, использование полиморфизма и композиции при едином базовом интерфейсе. Выделение алгоритма в объект. Соккрытие реализации.

Проектирование удаленного и отложенного выполнения.

Порождающие паттерны. “Ленивая” инициализация. Клонирование объектов. Фабрика. Поэтапное конструирование объекта.

Поведенческие паттерны. Архитектуры, управляемые событиями. Отложенное выполнение.

Инкапсуляция поведения. Подходы к реализации поведения, основанного на модели конечного автомата.

5. Перечень учебно-методического обеспечения для самостоятельной работы по дисциплине (модулю)

Методические указания для самостоятельной работы обучающихся по освоению дисциплины представлены в Приложении 2.

6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю)

Структура фонда оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине приведена в Приложении 1.

7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

Основная литература:

1. Обоснование и разработка требований к программным системам : учебное пособие / А. А. Бирюкова, А. М. Володина, К. В. Гусев, А. Н. Миرون. — Москва : РТУ МИРЭА, 2022. — 157 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/240089>. — Режим доступа: для авториз. пользователей.

2. Крючкова, Е. Н. Объектно-ориентированное программирование: Архитектурное проектирование и паттерны программирования : учебно-методическое пособие / Е. Н. Крючкова, С. М. Старолетов. — Барнаул : АлтГТУ, 2020. — 180 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/292790>. — Режим доступа: для авториз. пользователей.

1. Киреев, Н. В. Архитектурные компоненты и шаблоны проектирования в разработке программного обеспечения мобильных систем : учебное пособие / Н. В. Киреев. — Красноярск : СибГУ им. академика М. Ф. Решетнёва, 2022. — 90 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/330113>. — Режим доступа: для авториз. пользователей.
2. Завьялов, А. В. Диаграммы UML для анализа и проектирования информационных систем : учебно-методическое пособие / А. В. Завьялов. — Москва : РТУ МИРЭА, 2021. — 65 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/218630>. — Режим доступа: для авториз. пользователей.

8. Перечень ресурсов информационно-телекоммуникационной сети интернет, необходимых для освоения дисциплины (модуля)

Интернет-ресурсы:

1. <http://www.infojournal.ru> - научно-образовательный портал,
2. <http://msdn.microsoft.com/ru-ru/library/> - портал разработчика Microsoft,
3. <http://www.interface.ru/> - научно-образовательный портал.
4. <http://www.biblioclub.ru> – электронная библиотечная система
5. <http://znanium.com> – электронная библиотечная система

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Методические указания для обучающихся, по освоению дисциплины, приведены в Приложении 2.

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модуля)

Перечень программного обеспечения: Microsoft Office или свободно распространяемые аналоги, Microsoft Visio, Modelio Open Source, Draw.io, Ramus.

Информационные справочные системы:

Электронные ресурсы образовательной среды Университета.

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

Лекционные занятия:

- аудитория, оснащенная презентационной техникой (проектор, экран);
- комплект электронных презентаций/слайдов.

Практические занятия:

- Аудитория, оснащенная мультимедийными средствами (интерактивная доска).
- рабочее место преподавателя, оснащенное ПК с доступом в глобальную сеть Интернет;
- рабочие места студентов, оснащенные компьютерами с доступом в глобальную сеть Интернет и установленным программным обеспечением.

**Фонд оценочных средств для проведения промежуточной аттестации
обучающихся по дисциплине**

***ИНСТИТУТ
ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ***

КАФЕДРА ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ
ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО
ДИСЦИПЛИНЕ
«ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»**

Направление подготовки: 09.03.04 Программная инженерия

Направленность (профиль): Проектирование и разработка программного обеспечения

Уровень высшего образования: бакалавриат

Форма обучения: очная

1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

№ п/п	Индекс компетенции	Содержание компетенции (или ее части)*	Раздел дисциплины, обеспечивающий формирование компетенции (или ее части)	В результате изучения раздела дисциплины, обеспечивающего формирование компетенции (или ее части), обучающийся приобретает:		
				Трудовые действия	Необходимые умения	Необходимые знания
1	ПК-1	Способен определять и анализировать требования к программному обеспечению и возможность их реализации, в том числе разрабатывать технические спецификации на программное обеспечение и его компоненты	Тема 1-5	<p>Определяет требования к программному обеспечению и возможность их реализации;</p> <p>Разрабатывает технические спецификации на программное обеспечение и его компоненты</p>	<p>Анализирует требования к программному обеспечению;</p> <p>Определяет состав информации для разработки технических спецификаций на программное обеспечение и его компоненты;</p>	<p>Знает принципы определения требований к программному обеспечению;</p> <p>Знает принципы и правила разработки технических спецификаций на программное обеспечение и его компоненты</p>
2	ПК-2	Способен проектировать программное обеспечение, в том числе разрабатывать проектную документацию	Тема 1-5	Проектирует программное обеспечение	<p>Применяет различные способы и методы проектирования программного обеспечения;</p> <p>Разрабатывает модели информационных процессов;</p> <p>Разрабатывает проектную документацию</p>	<p>Знает основные принципы, способы и методы проектирования программного обеспечения;</p> <p>Знает состав проектной документации и основные принципы ее разработки</p>
3	ПК-5	Способен выполнять работы по созданию,	Тема 1-5	Осуществляет поддержку процесса	Применяет принципы определения требований	Знает принципы определения требований к интеграционному

№ п/п	Индекс компетенции	Содержание компетенции (или ее части)*	Раздел дисциплины, обеспечивающий формирование компетенции (или ее части)	В результате изучения раздела дисциплины, обеспечивающего формирование компетенции (или ее части), обучающийся приобретает:		
				Трудовые действия	Необходимые умения	Необходимые знания
		модификации и сопровождению интеграционных решений		согласования требований к интеграционному решению; Конфигурирует интеграционное решение	к интеграционному решению; Использует основные методы, способы и средства интеграции программных продуктов;	решению; Знает основные методы, способы и средства интеграции программных продуктов
	ПК-6	Способен управлять работами по созданию и модификации информационных ресурсов	Тема 1-5	Анализирует требования к информационным ресурсам, разрабатывает технические спецификации на информационные ресурсы; Проектирует, разрабатывает и тестирует информационные ресурсы	Применяет основные принципы разработки технических спецификаций на информационные ресурсы; Разрабатывает модели информационных ресурсов; Проводит тестирование информационных ресурсов;	Знает основные принципы разработки технических спецификаций на информационные ресурсы; Знает основные принципы тестирования информационных ресурсов
	ПК-7	Способен анализировать качество программного обеспечения и определять методы его повышения	Тема 1-5	Производит анализ качества программного обеспечения; Определяет и применяет методы повышения	Использует различные методы и способы анализа качества и повышения его уровня	Знает концепции и атрибуты качества; Знает методы и способы анализа качества и повышения его уровня

№ п/п	Индекс компетенции	Содержание компетенции (или ее части)*	Раздел дисциплины, обеспечивающий формирование компетенции (или ее части)	В результате изучения раздела дисциплины, обеспечивающего формирование компетенции (или ее части), обучающийся приобретает:		
				Трудовые действия	Необходимые умения	Необходимые знания
				качества программного обеспечения;		

2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Код компетенции	Инструменты, оценивающие сформированность компетенции	Показатель оценивания компетенции	Критерии оценки
ПК-1, ПК-2, ПК-5, ПК-6, ПК-7	Практическое задание	<p>А) полностью сформирована (компетенция освоена на <u>высоком</u> уровне) – 5 баллов</p> <p>Б) частично сформирована:</p> <ul style="list-style-type: none"> • компетенция освоена на <u>продвинутом</u> уровне – 4 балла; • компетенция освоена на <u>базовом</u> уровне – 3 балла; <p>В) не сформирована (<u>компетенция не сформирована</u>) – 2 и менее баллов</p>	<p>Проводится в форме письменной работы</p> <p>Время, отведенное на процедуру – 60 мин.</p> <p>Неявка – 0.</p> <p>Критерии оценки:</p> <p>1. Соответствие ответа заявленной тематике (0-5 баллов).</p> <p>Максимальная сумма баллов – 5 баллов.</p> <p>Результаты оценочной процедуры представляются обучающимся в срок не позднее 1 недели после проведения процедуры – для текущего контроля. Оценка проставляется в электронный журнал.</p>
ПК-1, ПК-2, ПК-5, ПК-6, ПК-7	Контрольная работа	<p>А) полностью сформирована (компетенция освоена на <u>высоком</u> уровне) – 5 баллов</p> <p>Б) частично сформирована:</p> <ul style="list-style-type: none"> • компетенция освоена на <u>продвинутом</u> уровне – 4 балла; • компетенция освоена на <u>базовом</u> уровне – 3 балла; <p>В) не сформирована (<u>компетенция не сформирована</u>) – 2 и менее баллов</p>	<p>Проводится в форме письменной работы</p> <p>Время, отведенное на процедуру – 60 мин.</p> <p>Неявка – 0.</p> <p>Критерии оценки:</p> <p>1. Соответствие ответа заявленной тематике (0-5 баллов).</p> <p>Максимальная сумма баллов – 5 баллов.</p> <p>Результаты оценочной процедуры представляются обучающимся в срок не позднее 1 недели после проведения процедуры – для текущего контроля. Оценка проставляется в электронный журнал.</p>

3. Типовые контрольные, практические задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Примерная тематика практических заданий:

1. Определить требования к информационной систем, составить техническое задание на разработку информационной системы.
2. Сформировать укрупненную и детализированные схемы IDEF0 в рамках проектирования заданной информационной системы.
3. Сформировать укрупненную и детализированные схемы IDEF3 в рамках проектирования заданной информационной системы.
4. С использованием языка UML сформировать диаграмму прецедентов в рамках проектирования заданной информационной системы.
5. С использованием языка UML сформировать диаграмму классов в рамках проектирования заданной информационной системы.
6. С использованием языка UML сформировать диаграмму состояний в рамках проектирования заданной информационной системы.
7. С использованием языка UML сформировать диаграмму деятельности в рамках проектирования заданной информационной системы.
8. С использованием языка UML сформировать диаграмму последовательностей в рамках проектирования заданной информационной системы.
9. С использованием языка UML сформировать диаграмму композиции в рамках проектирования заданной информационной системы.
10. С использованием языка UML сформировать диаграмму компонентов в рамках проектирования заданной информационной системы.
11. С использованием языка UML сформировать диаграмму реализации в рамках проектирования заданной информационной системы.
12. С использованием порождающего шаблона проектирования абстрактная фабрика (abstract factory) спроектировать программную реализацию конкретной задачи.
13. С использованием порождающего шаблона проектирования фабричный метод (factory method) спроектировать программную реализацию конкретной задачи.
14. С использованием структурного шаблона проектирования декоратор (decorator) спроектировать программную реализацию конкретной задачи.
15. С использованием структурного шаблона проектирования фасад (facade) спроектировать программную реализацию конкретной задачи.
16. С использованием поведенческого шаблона проектирования интерпретатор (interpreter) спроектировать программную реализацию конкретной задачи.
17. С использованием поведенческого шаблона проектирования хранибель (memento) спроектировать программную реализацию конкретной задачи.

18. С использованием поведенческого шаблона проектирования посетитель (visitor) спроектировать программную реализацию конкретной задачи.

Примерная тематика заданий контрольной работы:

1. Создать программный продукт для решения конкретной задачи с использованием порождающего шаблона проектирования абстрактная фабрика (abstract factory). При проектировании программного продукта использовать не менее двух диаграмм UML, включая диаграмму классов.

2. Создать программный продукт для решения конкретной задачи с использованием порождающего шаблона проектирования строитель (builder). При проектировании программного продукта использовать не менее двух диаграмм UML, включая диаграмму классов.

3. Создать программный продукт для решения конкретной задачи с использованием порождающего шаблона проектирования фабричный метод (factory method). При проектировании программного продукта использовать не менее двух диаграмм UML, включая диаграмму классов.

4. Создать программный продукт для решения конкретной задачи с использованием порождающего шаблона проектирования объектный пул (object pool). При проектировании программного продукта использовать не менее двух диаграмм UML, включая диаграмму классов.

5. Создать программный продукт для решения конкретной задачи с использованием порождающего шаблона проектирования одиночка (singleton). При проектировании программного продукта использовать не менее двух диаграмм UML, включая диаграмму классов.

6. Создать программный продукт для решения конкретной задачи с использованием порождающего шаблона проектирования прототип (prototype). При проектировании программного продукта использовать не менее двух диаграмм UML, включая диаграмму классов.

7. Создать программный продукт для решения конкретной задачи с использованием структурного шаблона проектирования декоратор (decorator). При проектировании программного продукта использовать не менее двух диаграмм UML, включая диаграмму классов.

8. Создать программный продукт для решения конкретной задачи с использованием структурного шаблона проектирования фасад (facade). При проектировании программного продукта использовать не менее двух диаграмм UML, включая диаграмму классов.

9. Создать программный продукт для решения конкретной задачи с использованием структурного шаблона проектирования компоновщик

(composite). При проектировании программного продукта использовать не менее двух диаграмм UML, включая диаграмму классов.

10. Создать программный продукт для решения конкретной задачи с использованием поведенческого шаблона проектирования цепочка ответственности (chain of responsibility). При проектировании программного продукта использовать не менее двух диаграмм UML, включая диаграмму классов.

11. Создать программный продукт для решения конкретной задачи с использованием поведенческого шаблона проектирования интерпретатор (interpreter). При проектировании программного продукта использовать не менее двух диаграмм UML, включая диаграмму классов.

12. Создать программный продукт для решения конкретной задачи с использованием поведенческого шаблона проектирования посредник (mediator). При проектировании программного продукта использовать не менее двух диаграмм UML, включая диаграмму классов.

13. Создать программный продукт для решения конкретной задачи с использованием поведенческого шаблона проектирования хранитель (memento). При проектировании программного продукта использовать не менее двух диаграмм UML, включая диаграмму классов.

14. Создать программный продукт для решения конкретной задачи с использованием поведенческого шаблона проектирования наблюдатель (observer). При проектировании программного продукта использовать не менее двух диаграмм UML, включая диаграмму классов.

15. Создать программный продукт для решения конкретной задачи с использованием поведенческого шаблона проектирования состояние (state). При проектировании программного продукта использовать не менее двух диаграмм UML, включая диаграмму классов.

16. Создать программный продукт для решения конкретной задачи с использованием поведенческого шаблона проектирования посетитель (visitor). При проектировании программного продукта использовать не менее двух диаграмм UML, включая диаграмму классов.

4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Формой контроля знаний по дисциплине «Проектирование программного обеспечения» являются две текущие аттестации в форме тестов и промежуточная аттестация в форме зачета с оценкой.

Неделя текущего контроля	Вид оценочного средства	Код компетенций, оценивающий знания, умения, навыки	Содержание оценочного средства	Требования к выполнению	Срок сдачи (неделя семестра)	Критерии оценки по содержанию и качеству с указанием баллов
7-8	тестирование	ПК-1 ПК-2 ПК-5 ПК-6 ПК-7	20 вопросов	Компьютерное тестирование; время, отведенное на процедуру - 30 минут	Результаты тестирования предоставляются в день проведения процедуры	Критерии оценки определяются процентным соотношением. Не явка - Удовлетворительно - от 51% правильных ответов. Хорошо - от 70%. Отлично – от 90%. Максимальная оценка – 5 баллов.
15-16	тестирование	ПК-1 ПК-2 ПК-5 ПК-6 ПК-7	20 вопросов	Компьютерное тестирование; время, отведенное на процедуру – 30 минут	Результаты тестирования предоставляются в день проведения процедуры	Критерии оценки определяются процентным соотношением. Не явка -0 Удовлетворительно - от 51% правильных ответов. Хорошо - от 70%. Отлично – от 90%. Максимальная оценка – 5 баллов.
согласно КУГ	Зачет с оценкой	ПК-1 ПК-2 ПК-5 ПК-6 ПК-7	1 вопрос	Зачет с оценкой проводится в устной форме, путем ответа на вопросы. Время, отведенное на процедуру – 30 минут.	Результаты предоставляются в день проведения зачета с оценкой	Критерии оценки: «Отлично»: • знание основных понятий предмета; • умение использовать и применять полученные

Неделя текущего контроля	Вид оценочного средства	Код компетенций, оценивающий знания, умения, навыки	Содержание оценочного средства	Требования к выполнению	Срок сдачи (неделя семестра)	Критерии оценки по содержанию и качеству с указанием баллов
						<p>знания на практике;</p> <ul style="list-style-type: none"> • работа на практических занятиях; • знание основных научных теорий, изучаемых предметов; • ответ на вопросы билета. <p>«Хорошо»:</p> <ul style="list-style-type: none"> • знание основных понятий предмета; • умение использовать и применять полученные знания на практике; • работа на практических занятиях; • знание основных научных теорий, изучаемых предметов; • ответы на вопросы билета • неправильно решено практическое задание <p>«Удовлетворительно»:</p> <ul style="list-style-type: none"> • демонстрирует

Неделя текущего контроля	Вид оценочного средства	Код компетенций, оценивающий знания, умения, навыки	Содержание оценочного средства	Требования к выполнению	Срок сдачи (неделя семестра)	Критерии оценки по содержанию и качеству с указанием баллов
						<p>частичные знания по темам дисциплин;</p> <ul style="list-style-type: none"> • незнание неумение использовать и применять полученные знания на практике; • не работал на практических занятиях; <p>«Неудовлетворительно»:</p> <ul style="list-style-type: none"> • демонстрирует частичные знания по темам дисциплин; • незнание основных понятий предмета; • неумение использовать и применять полученные знания на практике; • не работал на практических занятиях; <p>не отвечает на вопросы.</p>

4.1. Примерная тематика тестовых заданий

1. Целью проектирования информационной системы является

определение внутренних свойств системы и детализации её внешних (видимых) свойств на основе выданных заказчиком требований к программному обеспечению

минимизация использования ресурсов в связи с необходимостью экономии финансовых средств заказчика

составление технического задания, детально описывающего требования заказчика к разрабатываемой системе

реализация функций системы с использованием определенной среды программирования

2. Период времени, который начинается с момента принятия решения о создании программного продукта и заканчивается в момент его полного изъятия из эксплуатации – это

этап разработки программного обеспечения

период жизни программного обеспечения

этап проектирования программного обеспечения

жизненный цикл программного обеспечения

3. Позднее обнаружение проблем, связанных со сборкой проекта, в связи с одновременной интеграцией всех результатов в конце разработки относится к недостаткам

каскадной модели жизненного цикла

промежуточной модели жизненного цикла

инкрементной модели жизненного цикла

спиральной модели жизненного цикла

4. Возможность изменения требований в ходе разработки программного обеспечения относится к достоинствам

каскадной модели жизненного цикла

промежуточной модели жизненного цикла

инкрементной модели жизненного цикла

спиральной модели жизненного цикла

5. Декомпозиция, заключающаяся в разбиении системы на функциональные подсистемы, которые в свою очередь делятся на подфункции, подразделяемые на задачи, характерна для

объектно-ориентированного подхода к разработке программного обеспечения

универсального подхода к разработке программного обеспечения

структурного подхода к разработке программного обеспечения

поэлементного подхода к разработке программного обеспечения

6. Принципы структурных методов анализа и проектирования ПО включают в себя (возможно несколько правильных вариантов ответа):

принцип универсальности

принцип "разделяй и властвуй"

принцип тестирования

принцип непротиворечивости
принцип структурирования данных

7. Наиболее распространенными видами моделей (диаграмм) структурного анализа являются следующие (возможно несколько правильных вариантов ответа):

ERD
SADT
CSV
DFD
ERP

8. Назовите элементы диаграммы потоков данных DFD (возможно несколько правильных вариантов ответа):

хранилище данных
обработка данных
поток данных
база данных

9. Фундаментальный элемент, содержащий в себе структуры данных и поведение, – это

класс
объект
сообщение
атрибут

10. Запрос на выполнение действия, дополненный набором аргументов, которые могут понадобиться при выполнении действия, – это

класс
объект
сообщение
атрибут

11. Поименованное свойство класса, определяющее диапазон допустимых значений, которые могут принимать экземпляры данного свойства, – это

класс
объект
сообщение
атрибут

12. Назовите типы диаграмм UML (возможно несколько правильных вариантов ответа):

интеграции
поведения
взаимодействия
определения

13. Разновидность диаграмм UML, которая служит для описания сервисов, которые система предоставляет актору, – это
- диаграмма состояния
 - диаграмма классов
 - диаграмма деятельности
 - диаграмма вариантов использования
14. Назовите отношения, используемые в диаграмме вариантов использования (прецедентов) UML (возможно несколько правильных вариантов ответа):
- обобщения
 - включения
 - агрегации
 - ассоциации
 - композиции
15. Квантор видимости атрибута класса, указываемый на диаграмме UML символом "-", обозначает атрибут с областью видимости типа
- закрытый
 - общедоступный
 - открытый
 - защищенный
16. В диаграмме состояний UML отношение между двумя последовательными состояниями, которое указывает на факт смены одного состояния другим, – это
- класс
 - состояние действия
 - переход
 - квантор видимости
17. Для моделирования бизнес-процессов в диаграмме деятельности UML используется специальная конструкция, которая называется
- переход
 - дорожка
 - этап
 - операция
18. Диаграммы последовательностей UML описывают
- типы классов системы и статические связи, которые существуют между ними
 - особенности и реализации выполняемых системой операций
 - процесс изменения состояний экземпляра определенного класса
 - взаимодействие объектов, включая сообщения, которыми они обмениваются
19. Паттерны проектирования, абстрагирующие процесс наследования, – это

- поведенческие паттерны
 - порождающие паттерны
 - структурные паттерны
 - классические паттерны
20. Паттерны проектирования, связанные с алгоритмами и распределением обязанностей между объектами, - это
- поведенческие паттерны
 - порождающие паттерны
 - структурные паттерны
 - классические паттерны

4.2. Типовые вопросы, выносимые на зачет с оценкой

1. Системный подход к проектированию ПО.
2. Понятие программного обеспечения, проекта и проектирования ПО.
3. Определение жизненного цикла ПО.
4. Этапы (стадии) ЖЦ ПО.
5. Каскадная (водопадная) модель ЖЦ ПО
6. Инкрементная модель ЖЦ ПО
7. Спиральная модель ЖЦ ПО
8. Определение требований к ПО
9. Проблемы проектирования ПО.
10. Структурные методы анализа и проектирования ПО.
11. Метод функционального моделирования SADT (IDEF0).
12. Метод моделирования процессов IDEF3.
13. Моделирование потоков данных DFD.
14. Основные принципы и свойства объектно-ориентированного проектирования.
15. Основные элементы объектной модели.
16. Объекты, свойства объектов.
17. Понятие классов.
18. Общая характеристика классов.
19. Пример использования объектно-ориентированного подхода.
20. Сопоставление и взаимосвязь структурного и объектно-ориентированного подходов.
21. Язык UML: основные понятия, виды диаграмм.
22. Язык UML: диаграмма вариантов использования (прецедентов).
23. Язык UML: диаграмма классов.
24. Язык UML: диаграмма состояний.
25. Язык UML: диаграмма деятельности.
26. Язык UML: диаграмма последовательностей.
27. Язык UML: диаграмма коммуникации.
28. Язык UML: диаграмма компонентов.

29. Язык UML: диаграмма развертывания.
30. Структурные паттерны. Рекурсивная композиция, использование полиморфизма и композиции при едином базовом интерфейсе.
31. Структурные паттерны. Выделение алгоритма в объект. Соккрытие реализации.
32. Структурные паттерны. Проектирование удаленного и отложенного выполнения.
33. Порождающие паттерны. “Ленивая” инициализация.
34. Порождающие паттерны. Клонирование объектов.
35. Порождающие паттерны. Фабрика. Поэтапное конструирование объекта.
36. Поведенческие паттерны. Архитектуры, управляемые событиями. Отложенное выполнение.
37. Поведенческие паттерны. Инкапсуляция поведения. Подходы к реализации поведения, основанного на модели конечного автомата.

Итоговое начисление баллов по дисциплине осуществляется в соответствии с разработанной и внедренной балльно-рейтинговой системой контроля и оценивания уровня знаний и внеучебной созидательной активности обучающихся.

***ИНСТИТУТ
ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ***

КАФЕДРА ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО
ОСВОЕНИЮ ДИСЦИПЛИНЫ
«ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»**

Направление подготовки: 09.03.04 Программная инженерия

Направленность (профиль): Проектирование и разработка программного обеспечения

Уровень высшего образования: бакалавриат

Форма обучения: очная

Королев
2023

1. Общие положения

Целью изучения дисциплины является:

- получение знаний и умений в области индустриальных способов (инженерных методов и средств) создания программного обеспечения (ПО);
- получение практических навыков работы с инструментальными средствами разработки программного обеспечения.

Задачи дисциплины:

- ознакомление с понятием качества программных продуктов, жизненным циклом программных продуктов и его моделями;
- изучение методов анализа и проектирования ПО;
- получение навыков применения современных CASE - средств поддержки этапов жизненного цикла ПО;
- изучение методов тестирования ПО.

2. Указания по проведению практических занятий

Практическое занятие 1-2

Вид практического занятия: смешанная форма практического занятия.

Образовательная технология: командная работа

Тема и содержание практического занятия: Жизненный цикл программного обеспечения

Продолжительность занятия – 4 ч.

Практическое занятие 3-4

Вид практического занятия: смешанная форма практического занятия.

Образовательная технология: кейс-технология

Тема и содержание практического занятия: Анализ требований к программному обеспечению

Продолжительность занятия - 4 ч.

Практическое занятие 5-7

Вид практического занятия: смешанная форма практического занятия.

Образовательная технология: кейс-технология

Тема и содержание практического занятия: Структурные (функциональные) методы проектирования программного обеспечения

Продолжительность занятия - 6 ч.

Практическое занятие 8-12

Вид практического занятия: смешанная форма практического занятия.

Образовательная технология: кейс-технология

Тема и содержание практического занятия: Объектно-ориентированные методы проектирования программного обеспечения

Продолжительность занятия - 10 ч.

Практическое занятие 13-16

Вид практического занятия: смешанная форма практического занятия.

Образовательная технология: кейс-технология

Тема и содержание практического занятия: Паттерны проектирования программного обеспечения

Продолжительность занятия - 8 ч.

3. Указания по проведению лабораторного практикума

Не предусмотрено учебным планом.

4. Указания по проведению самостоятельной работы студентов

№ п/п	Наименование блока (раздела) дисциплины	Виды СРС
1	Тема 1. Жизненный цикл программного обеспечения	Самостоятельное изучение тем Вопросы, выносимые на самостоятельное изучение: 1. ISO/IEC 12207:1995 «Информационные технологии. Процессы жизненного цикла программного обеспечения» 2. ISO 15226:1999 «Техническая документация на продукцию. Модель жизненного цикла и назначение документов» 3. Стандарт ГОСТ 34.601. Автоматизированные системы. Стадии создания.
2	Тема 2. Анализ требований к программному обеспечению	Самостоятельное изучение тем. Вопросы, выносимые на самостоятельное изучение: 1. Сравнение различных подходов к составлению спецификаций
3	Тема 3. Структурные (функциональные) методы проектирования программного обеспечения	Самостоятельное изучение тем. Вопросы, выносимые на самостоятельное изучение: 1. Фреймворковые модели
4	Тема 4. Объектно-ориентированные методы проектирования программного обеспечения	Самостоятельное изучение тем. Вопросы, выносимые на самостоятельное изучение: 1. Сравнение UML 1.x и UML 2.x
5	Тема 5. Паттерны проектирования программного обеспечения	Самостоятельное изучение тем. Вопросы, выносимые на самостоятельное изучение: 1. Классификация паттернов

5. Перечень основной и дополнительной учебной литературы,

необходимой для освоения дисциплины (модуля)

Основная литература:

3. Обоснование и разработка требований к программным системам : учебное пособие / А. А. Бирюкова, А. М. Володина, К. В. Гусев, А. Н. Миرونнов. — Москва : РТУ МИРЭА, 2022. — 157 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/240089>. — Режим доступа: для авториз. пользователей.
4. Крючкова, Е. Н. Объектно-ориентированное программирование: Архитектурное проектирование и паттерны программирования : учебно-методическое пособие / Е. Н. Крючкова, С. М. Старолетов. — Барнаул : АлтГТУ, 2020. — 180 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/292790>. — Ре-

Дополнительная литература:

3. Киреев, Н. В. Архитектурные компоненты и шаблоны проектирования в разработке программного обеспечения мобильных систем : учебное пособие / Н. В. Киреев. — Красноярск : СибГУ им. академика М. Ф. Решетнёва, 2022. — 90 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/330113>. — Режим доступа: для авториз. пользователей.
4. Завьялов, А. В. Диаграммы UML для анализа и проектирования информационных систем : учебно-методическое пособие / А. В. Завьялов. — Москва : РТУ МИРЭА, 2021. — 65 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/218630>. — Режим доступа: для авториз. пользователей.

8. Перечень ресурсов информационно-телекоммуникационной сети интернет, необходимых для освоения дисциплины (модуля)

Интернет-ресурсы:

1. <http://www.infojournal.ru> - научно-образовательный портал,
2. <http://msdn.microsoft.com/ru-ru/library/> - портал разработчика Microsoft,
3. <http://www.interface.ru/> - научно-образовательный портал.
4. <http://www.biblioclub.ru> – электронная библиотечная система
5. <http://znanium.com> – электронная библиотечная система

6. Перечень информационных технологий

Перечень программного обеспечения: Microsoft Office или свободно распространяемые аналоги, Microsoft Visio, Modelio Open Source, Draw.io, Ramus.

Информационные справочные системы:

Электронные ресурсы образовательной среды Университета.