

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Макаренко Елена Николаевна

Должность:

Документ подписан в:

Дата подписания: 20.06.2026 11:50:33

Уникальный программный ключ:

c098bc0c1041cb2a4cf926cf171d6715d99a6ae00adc8e27b55cbe1e2dbd7c78

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Ростовский государственный экономический университет (РИНХ)»

УТВЕРЖДАЮ

Начальник

учебно-методического управления

Т.К. Платонова

«25» мая 2026 г.

**Рабочая программа дисциплины
Методология программной инженерии**

Направление подготовки
09.04.04 Программная инженерия

Направленность (профиль) программы магистратуры
09.04.04.01 Системное и прикладное программное обеспечение

Для набора 2026 года

Квалификация
магистр

КАФЕДРА Информационные технологии и программирование**Распределение часов дисциплины по семестрам / курсам**

Семестр (<Курс>.<Семестр на курсе>)	1 (1.1)		Итого	
	14			
Неделя	14			
Вид занятий	уп	рп	уп	рп
Лекции	8	8	8	8
Лабораторные	16	16	16	16
Итого ауд.	24	24	24	24
Контактная работа	24	24	24	24
Сам. работа	156	156	156	156
Часы на контроль	36	36	36	36
Итого	216	216	216	216

ОСНОВАНИЕ

Учебный план утвержден учёным советом Университета (протокол № 9 от 03.03.2026 г.).

Программу составил(и): к.э.н., доцент, Жилина Е.В.

Зав. кафедрой: к.э.н., доцент Е.В. Ефимова

Методический совет направления: д.э.н., профессор Е.Н. Тищенко

Директор института магистратуры: д.э.н., профессор Е.А. Иванова

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1.1	разработка моделей, применение методологий, технологий исследования и проектирования объектов профессиональной деятельности на основе общих тенденций развития программной инженерии
-----	--

2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

ОПК-3. Способен анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями;

ОПК-8. Способен осуществлять эффективное управление разработкой программных средств и проектов.

В результате освоения дисциплины обучающийся должен:

Знать:

принципы, методы и средства анализа и структурирования профессиональной информации (соотнесено с индикатором ОПК-3.1); методы эффективного управления разработкой программных средств и проектов (соотнесено с индикатором ОПК-8.1)

Уметь:

анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров (соотнесено с индикатором ОПК-3.2); применять эффективное управление разработкой программных средств и проектов (соотнесено с индикатором ОПК-8.2)

Владеть:

навыками подготовки научных докладов, публикаций и аналитических обзоров с обоснованными выводами и рекомендациями (соотнесено с индикатором ОПК-3.3); навыками эффективного управления разработкой программных средств и проектов (соотнесено с индикатором ОПК-8.3)

3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Раздел 1. Модели и методологии разработки программного обеспечения

№	Наименование темы, краткое содержание	Вид занятия / работы / форма ПА	Семестр / Курс	Количество часов	Компетенции
1.1	Тема 1.1 «Введение. Стандарты и профили стандартов ЖЦ систем и программных средств в программной инженерии» Концепции и задачи системной инженерии. Системный программист. Архитектор программного обеспечения. Руководитель разработки программного обеспечения. Основные понятия. Характеристика деятельности. Основы ЖЦ программных средств. Модели. Этапы. Инструментарий. Сравнительный анализ. Критерии выбора. Системные основы современных технологий программной инженерии. Назначение стандартов ЖЦ в программной инженерии. ЖЦ профилей стандартов систем и программных средств. Модель профиля стандартов ЖЦ сложных программных средств. Основные ресурсы для обеспечения ЖЦ сложных ПС. Ресурсы специалистов для обеспечения ЖЦ сложных ПС.	Лекционные занятия	1	2	ОПК-3 ОПК-8
1.2	Тема 1.2. «Модели, методологии разработки программного обеспечения и управление проектами программных средств» Управление проектами программных средств в системе - СММІ. Стандарты менеджмента качества программ. Стандарты открытых систем, регламентирующие структуру и интерфейсы программных средств. Гибкие методологии процесса разработки программного обеспечения. Agile. Kanban. Scrum. Extreme Programming.	Лекционные занятия	1	2	ОПК-3 ОПК-8
1.3	Тема 1.1 «Введение. Стандарты и профили стандартов ЖЦ систем и программных средств в программной инженерии» Разработка класса на языке C++ (Visual Studio Code). Разработка интерфейса на языке C# (Visual Studio Code). Разработка событий на языке C# (Visual Studio Code).	Лабораторные занятия	1	4	ОПК-3 ОПК-8
1.4	Тема 1.2. «Модели и процессы управления проектами программных средств» Кодогенерация UML – модели для проекта «Телефонный справочник» на языке Java (UMLet). Веб-публикация. Обратное проектирование (Java To UML).	Лабораторные занятия	1	4	ОПК-3 ОПК-8
1.5	Характеристики защиты и безопасности функционирования программных средств, разработка защищенного программного кода	Самостоятельная работа	1	60	ОПК-3 ОПК-8

Раздел 2. Разработка и сопровождение программного обеспечения

№	Наименование темы, краткое содержание	Вид занятия / работы / форма ПА	Семестр / Курс	Количество часов	Компетенции
2.1	Тема 2.1 «Технико-экономическое обоснование проектов программных средств». Проект и организационная структура компании. Организация проектной команды. Ключевые участники и заинтересованные стороны. График работ. Концепция проекта. Цели, задачи и результаты проекта. Ресурсы. Ограничения. Требования. Цели и процессы технико-экономического обоснования проектов ПС. Экспертное технико-экономическое обоснование (ТЭО) проектов ПС.	Лекционные занятия	1	2	ОПК-3 ОПК-8
2.2	Тема 2.2. «Документирование сложных программных средств». Структура и содержание – шаблоны документов сложных программных средств. Стандарты, регламентирующие документирование проектов сложного программного средства. Техническое задание. Пояснительная записка. Формирование требований к документации сложных программных средств. Разработка спецификаций. Руководство пользователя. Эксплуатация программной системы (компонента, библиотеки). Проведение испытаний. Документирование тестов.	Лекционные занятия	1	2	ОПК-3 ОПК-8
2.3	Тема 2.1 «Технико-экономическое обоснование проектов программных средств». Кодогенерация UML – модели для проекта «Справочники товаров» на языке C/C++ (UMLet). Доработать сгенерированный шаблон проекта в Console-приложении.	Лабораторные занятия	1	4	ОПК-3 ОПК-8
2.4	Тема 2.2. «Документирование сложных программных средств». Кодогенерация UML – модели для проекта «Учет программного обеспечения» .Net (UMLet). Доработать сгенерированный шаблон проекта.	Лабораторные занятия	1	4	ОПК-3 ОПК-8
2.5	Курсовой проект. Перечень тем представлен в Приложении 1 к рабочей программе дисциплины	Самостоятельная работа	1	96	ОПК-3 ОПК-8
2.6	Подготовка к промежуточной аттестации	Экзамен	1	36	ОПК-3 ОПК-8

4. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Структура и содержание фонда оценочных средств для проведения текущего контроля и промежуточной аттестации представлены в Приложении 1 к рабочей программе дисциплины.

5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

5.1. Учебные, научные и методические издания

	Авторы, составители	Заглавие	Издательство, год	Библиотека / Количество
1		Программные продукты и системы: журнал	Тверь: Центрпрограммсистем, 2017	ЭБС «Университетская библиотека онлайн»
2	Леоненков, А. В.	Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose: учебное пособие	Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020	ЭБС «IPR SMART»
3	Романов Е. Л.	Программная инженерия: учебное пособие	Новосибирск: Новосибирский государственный технический университет, 2017	ЭБС «Университетская библиотека онлайн»
4	Суханов, М. Б.	Программная инженерия: учебное пособие	Санкт-Петербург: Санкт-Петербургский государственный университет промышленных технологий и дизайна, 2018	ЭБС «IPR SMART»

5.2. Профессиональные базы данных и информационные справочные системы

Тематический блог, посвященный информационным технологиям, бизнесу и интернету. - [Электронный ресурс]. - <https://habr.com/>
 Тематический журнал Открытые системы. - [Электронный ресурс]. - <https://www.osp.ru/os/>
 Официальный сайт IBM. Раздел посвященный Rational Architect. - [Электронный ресурс]. - <https://www.ibm.com/docs/en/rational-soft-arch/9.7.0>.
 ИСС «КонсультантПлюс»
 ИСС «Гарант» <http://www.internet.garant.ru/>

5.3. Перечень программного обеспечения

Операционная система РЕД ОС
 Visual Studio Code
 UMLet

5.4. Учебно-методические материалы для обучающихся с ограниченными возможностями здоровья

При необходимости по заявлению обучающегося с ограниченными возможностями здоровья учебно-методические материалы предоставляются в формах, адаптированных к ограничениям здоровья и восприятия информации. Для лиц с нарушениями зрения: в форме аудиофайла; в печатной форме увеличенным шрифтом. Для лиц с нарушениями слуха: в форме электронного документа; в печатной форме. Для лиц с нарушениями опорно-двигательного аппарата: в форме электронного документа; в печатной форме.

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Помещения для всех видов работ, предусмотренных учебным планом, укомплектованы необходимой специализированной учебной мебелью и техническими средствами обучения:

- столы, стулья;
- персональный компьютер / ноутбук (переносной);
- проектор;
- экран / интерактивная доска.

Лабораторные занятия проводятся в компьютерных классах, рабочие места в которых оборудованы необходимыми лицензионными и/или свободно распространяемыми программными средствами и выходом в Интернет, и/или в специализированных лабораториях, предусмотренных образовательной программой.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа к электронной информационно-образовательной среде.

7. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Методические указания по освоению дисциплины представлены в Приложении 2 к рабочей программе дисциплины.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

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

1.1 Критерии оценивания компетенций

ЗУН, составляющие компетенцию	Показатели оценивания	Критерии оценивания	Средства оценивания*
ОПК-3: Способен анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями;			
З: принципы, методы и средства анализа и структурирования профессиональной информации	изучает основную и дополнительную литературу, лекционный материал, использует профессиональные базы при подготовке к экзамену и опросу, при выборе тематики курсового проекта	полнота и содержательность ответа на опросе и экзамене, умение приводить примеры, умение отстаивать свою позицию; соответствие ответов материалам лекций и учебной литературы, сведениям из информационных ресурсов Интернет; актуальность выбора тематики курсового проекта	Вопросы к экзамену (1-50), типовые практико-ориентированные задания к экзамену (1-5), вопросы для опроса (1-10), лабораторные задания (1-1), курсовой проект (1-45)
У: анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров	применяет методы анализа профессиональной информации для решения лабораторных, практико-ориентированных заданий и в курсовом проекте	правильность применения методов анализа профессиональной информации в лабораторных и практико-ориентированных заданиях; обоснованность применения выбранного метода в курсовом проекте	Вопросы к экзамену (1-50), типовые практико-ориентированные задания к экзамену (1-5), вопросы для опроса (1-10), лабораторные задания (1-1), курсовой проект (1-45)
В: навыками подготовки научных докладов,	применяет навыки подготовки научных докладов, публикаций и	объем и индивидуальность выполнения задания с	Вопросы к экзамену (1-50), типовые

публикаций и аналитических обзоров с обоснованными выводами и рекомендациями	аналитических обзоров с обоснованными выводами и рекомендациями на основе результатов лабораторных, практико-ориентированных и курсового проекта с использованием современного инструментария	использованием современного инструментария; корректность интерпретации полученных результатов в задании	практико-ориентированные задания к экзамену (1-5), вопросы для опроса (1-10), лабораторные задания (1-1), курсовой проект (1-45)
ОПК-8: Способен осуществлять эффективное управление разработкой программных средств и проектов.			
З: методы эффективного управления разработкой программных средств и проектов	изучает основную и дополнительную литературу, лекционный материал, использует профессиональные базы данных для изучения основ эффективного управления разработкой программных средств и проектов при подготовке к экзамену и опросу, при выборе тематики курсового проекта	полнота и содержательность ответа на опросе и экзамене, умение приводить примеры, умение отстаивать свою позицию; соответствие ответов материалам лекций и учебной литературы, сведениям из информационных ресурсов Интернет; актуальность выбора исходных данных для моделей курсового проекта	Вопросы к экзамену (1-50), типовые практико-ориентированные задания к экзамену (1-5), вопросы для опроса (1-10), лабораторные задания (1-1), курсовой проект (1-45)
У: применять эффективное управление разработкой программных средств и проектов	применяет методы эффективного управления разработкой программных средств и проектов при решении лабораторных, практико-ориентированных заданий и в курсовом проекте	правильность применения методов эффективного управления разработкой программных средств и проектов в лабораторных и практико-ориентированных заданиях; обоснованность применения выбранного метода в курсовом проекте	Вопросы к экзамену (1-50), типовые практико-ориентированные задания к экзамену (1-5), вопросы для опроса (1-10), лабораторные задания (1-1), курсовой проект (1-45)
В: навыками эффективного управления разработкой	применяет методы эффективного управления разработкой программных	объем и индивидуальность выполнения задания с использованием современного инструментария и применения	Вопросы к экзамену (1-50), типовые практико-

программных средств и проектов	средств и проектов для решения лабораторных, практико-ориентированных заданий и в курсовом проекте с использованием современного инструментария	методов эффективного управления разработкой программных средств и проектов; корректность интерпретации полученных результатов в курсовом проекте	ориентированные задания к экзамену (1-5), вопросы для опроса (1-10), лабораторные задания (1-1), курсовой проект (1-45)
--------------------------------	---	---	---

1.2. Шкала оценивания

Текущий контроль успеваемости и промежуточная аттестация осуществляется в рамках накопительной балльно-рейтинговой системы в 100-балльной шкале:

- 84-100 баллов (оценка «отлично»)
- 67-83 баллов (оценка «хорошо»)
- 50-66 баллов (оценка «удовлетворительно»)
- 0-49 баллов (оценка «неудовлетворительно»)

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

Вопросы к экзамену

1. Концепции и задачи системной инженерии.
2. Системный программист.
3. Архитектор программного обеспечения.
4. Руководитель разработки программного обеспечения.
5. Организация разработки требований к сложным программным средствам.
6. Процессы разработки требований к характеристикам сложных программных средств.
7. Структура основных документов, отражающих требования к программным средствам
8. Разработка, тестирование модулей, компонентов и комплексов программ.
9. Цели и принципы системного проектирования сложных программных средств.
10. Процессы системного проектирования программных средств.
11. Структурное проектирование сложных программных средств.
12. Гибкая методология процесса разработки программного обеспечения: Agile.
13. Гибкая методология процесса разработки программного обеспечения: Kanban.
14. Гибкая методология процесса разработки программного обеспечения: Scrum.
15. Гибкая методология процесса разработки программного обеспечения: Extreme Programming.
16. Архитектура ПС.
17. Задачи и особенности объектно-ориентированного проектирования (ООП) ПС.
18. Основные понятия и модели ООП ПС.
19. Варианты представления моделей и средства ООП ПС.
20. Принципы верификации и тестирования программ.

21. Процессы и средства тестирования программных компонентов.
22. Технологические этапы и стратегии систематического тестирования программ.
23. Примеры оценок сложности тестирования программ.
24. Тестирование обработки потоков данных программными компонентами.
25. Организация планирования жизненного цикла сложных программных средств.
26. Основные факторы, определяющие качество сложных программных средств.
27. Конструктивные характеристики качества сложных программных средств.
28. Характеристики качества баз данных.
29. Характеристики защиты и безопасности функционирования программных средств.
30. Документирование процессов и результатов сертификации программных продуктов.
31. Основные понятия ЖЦ.
32. Управление проектами программных средств в системе - СММІ.
33. Стандарты менеджмента качества программ.
34. Стандарты открытых систем, регламентирующие структуру и интерфейсы программных средств.
35. Характеристика деятельности ЖЦ.
36. Основы ЖЦ программных средств.
37. Системные основы современных технологий программной инженерии.
38. Назначение стандартов ЖЦ в программной инженерии.
39. ЖЦ профилей стандартов систем и программных средств.
40. Модель профиля стандартов ЖЦ сложных программных средств.
41. Основные ресурсы для обеспечения ЖЦ сложных ПС.
42. Ресурсы специалистов для обеспечения ЖЦ сложных ПС.
43. Цели и процессы технико - экономического обоснования проектов ПС.
44. Экспертное технико - экономическое обоснование (ТЭО) проектов ПС.
45. Формирование требований к документации сложных программных средств.
46. Стандарты, регламентирующие документирование проектов сложных программного средства.
47. Документальная поддержка разработки программного обеспечения. Техническое задание. Пояснительная записка.
48. Документирование требований. Разработка спецификаций.
49. Документальное сопровождение программного обеспечения. Руководство пользователя.
50. Эксплуатация программной системы (компонента, библиотеки). Проведение испытаний.

Типовые практико-ориентированные задания к экзамену*

1. Разработка диаграммы прецедентов.
2. Разработки диаграммы классов.
3. Разработка диаграммы компонентов.
4. Разработка диаграммы деятельности.
5. Разработка диаграммы состояний.

Примечание *: Практические задачи на экзамене могут формироваться преподавателем согласно любой предметной области.

Критерии оценивания:

- 84-100 (оценка «отлично») – изложенный материал фактически верен, наличие глубоких исчерпывающих знаний в объеме пройденной программы дисциплины в соответствии с поставленной программой курса целью обучения; правильные, уверенные действия по применению полученных навыков и умений при решении практико-ориентированного задания, грамотное и логически стройное изложение материала при ответе, усвоение основной и знакомство с дополнительной литературой;

- 67-83 (оценка «хорошо») – наличие твердых и достаточно полных знаний в объеме пройденной программы дисциплины в соответствии с целью обучения, правильные действия по применению навыков и умений при решении практико-ориентированного задания, четкое изложение материала, допускаются отдельные логические и стилистические погрешности, обучающийся усвоил основную литературу, рекомендованную в рабочей программе дисциплины;

- 50-66 (оценка «удовлетворительно») – наличие твердых знаний в объеме пройденного курса в соответствии с целью обучения, изложение ответов с отдельными ошибками, уверенно исправленными после дополнительных вопросов; правильные в целом действия по применению навыков и умений при решении практико-ориентированного задания;

- 0-49 (оценка «неудовлетворительно») – ответы не связаны с вопросами, наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять умения и навыки при решении практико-ориентированного задания, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

Перечень теоретических типовых вопросов для опроса*

1. Назовите цели и принципы системного проектирования сложных программных средств.
2. Что понимаете под архитектурой программного средства?
3. Приведите задачи и особенности объектно-ориентированного проектирования (ООП).
4. Перечислите основные факторы, определяющие качество сложных программных средств.
5. Перечислите стандарты менеджмента качества программных систем.
6. Какие Вы знаете модели ЖЦ программных средств.
7. Приведите цели и процессы технико-экономического обоснования программных проектов.
8. Что Вы понимаете под экспертным технико-экономическим обоснованием программных проектов.
9. Каковы преимущества языка UML?
10. Что такое RUP?

Примечание *: опрос проводится при проверке всех лабораторных заданий для выявления знаний при изучении соответствующих тем дисциплины в рамках текущей аттестации.

Критерии оценивания:

- 1 балл выставляется обучающемуся, если изложенный материал фактически верен и логически обоснован.

Максимальное количество баллов: 10 баллов.

Лабораторные задания

Тематика лабораторных заданий

1. Разработка класса на языке C++ (Visual Studio Code) - **(10 баллов)**
2. Разработка интерфейса на языке C# (Visual Studio Code) - **(10 баллов)**
3. Разработка событий на языке C# (Visual Studio Code) - **(10 баллов)**
4. Кодогенерация UML – модели для проекта «Телефонный справочник» на языке Java (UMLet). Веб-публикация. Обратное проектирование (Java To UML) - **(10 баллов)**

Критерии оценивания каждого лабораторного задания (1-4):

- 10 баллов выставляется студенту, если все задания, предусмотренное лабораторным заданием, выполнены, и студент может объяснить их выполнение;
- 9-6 балла выставляется студенту, если все задания, предусмотренное лабораторным заданием, выполнены, и студент затрудняется объяснить их выполнение;
- 5-1 балла выставляется студенту, если не все задания, предусмотренное лабораторным заданием, выполнены, и студент затрудняется объяснить их выполнение;
- 0 баллов выставляется студенту, если задание, предусмотренное лабораторным заданием, не выполнено.

5. Кодогенерация UML – модели для проекта «Справочники товаров» на языке C/C++ (UMLet) Доработать сгенерированный шаблон проекта в Console-приложении - (25 баллов)

Описание проекта «Справочники товаров»

- Проект должен содержать абстрактный класс Goods с функциями, позволяющими вывести на экран информацию о товаре, а также определить, соответствует ли она искомому типу.

- В абстрактном классе Goods реализовать метод CompareTo так, чтобы можно было отсортировать базу данных о товарах по возрасту детей, на которых он рассчитан.

- Создать производные классы: Toy (название, цена, производитель, материал, возраст, на который рассчитана), Book (название, автор, цена, издательство, возраст, на который рассчитана), SportsEquipment (название, цена, производитель, возраст, на который рассчитан).

- Создать базу из n товаров, вывести полную информацию из базы на экран, отсортировав базу данных о товарах по возрасту детей, на которых он рассчитан, а также организовать поиск товаров определенного типа.

6. Кодогенерация UML – модели для проекта «Учет программного обеспечения» .Net (UMLet). Доработать сгенерированный шаблон проекта в Windows-приложении - (25 баллов)

Описание проекта «Учет программного обеспечения»

- Проект должен содержать абстрактный класс Software с методами, позволяющими вывести на экран информацию о программном обеспечении, а также определить соответствие возможности использования (на момент текущей даты).

- В абстрактном классе Software реализовать метод CompareTo так, чтобы можно было отсортировать базу данных по названию ПО.

- Создать производные классы: FreeSoftware (название, производитель), SharewareSoftware (название, производитель, дата установки, срок бесплатного использования), ProprietarySoftware (название, производитель, цена, дата установки, срок использования).

- Создать базу из n видов программного обеспечения, вывести полную информацию из базы на экран, отсортировав базу данных по названию ПО, а также организовать поиск программного обеспечения, которое допустимо использовать на текущую дату.

Критерии оценивания каждого лабораторного задания (5-6):

- 21-25 баллов выставляется студенту, если все задания, предусмотренное лабораторным заданием, выполнены, и студент может объяснить их выполнение;
- 11-20 балла выставляется студенту, если все задания, предусмотренное лабораторным заданием, выполнены, и студент затрудняется объяснить их выполнение;
- 1-10 балла выставляется студенту, если не все задания, предусмотренное лабораторным заданием, выполнены, и студент затрудняется объяснить их выполнение;

- 0 баллов выставляется студенту, если задание, предусмотренное лабораторным заданием, не выполнено.

Максимальное количество баллов, которые могут быть получены обучающимся - 90.

Курсовой проект

Типовая тематика курсового проекта

1. Разработка методов анализа объектов программной инженерии.
2. Анализ и характеристика областей знаний SWEBOK.
3. Методы и инструменты работы с требованиями в программной инженерии.
4. Методы и инструменты проектирования программного обеспечения.
5. Инструменты конструирования в программной инженерии.
6. Инструменты конфигурационного управления и сопровождения программных систем.
7. Инструменты управления инженерной деятельностью.
8. Инструменты обеспечения качества программного обеспечения.
9. Эвристические методы программной инженерии.
10. Формальные методы программной инженерии.
11. Методы прототипирования программной инженерии.
12. Реализация моделей жизненного цикла программного обеспечения.
13. Принципы объектно-ориентированного подхода в программной инженерии.
14. Принципы структурного подхода в программной инженерии.
15. Методы верификации и тестирования инженерных программ и систем.
16. Модели качества и надежности в программной инженерии.
17. Методы управления программным проектом, рисками и конфигурацией.
18. Сравнительный анализ Case-средств структурного подхода к разработке программного обеспечения.
19. Сравнительный анализ Case-средств объектно-ориентированного подхода к разработке программного обеспечения.
20. «Гибкие» (agile) методы разработки программного обеспечения.
21. Технология командной разработки.
22. Технология командной разработки. Принципы Git.
23. Распознавание графических (видео) образов на основе библиотеки OpenCV.
24. Распознавание звука (синтезатора речи) на основе Yandex SpeechKit (Google Speech API).
25. Интеллектуальная обработка данных.
26. Проектирование «умного дома».
27. Разработка компонент «умного дома».
28. Разработка компонент нейронной сети для решения задачи «*».
29. Разработка компонент нечеткой логики для решения задачи «*».
30. Разработка dll-библиотеки использования «генетических» алгоритмов.
31. Разработка параллельных алгоритмов для решения задач «*».
32. Проектирование и разработка клиентского модуля для решения задачи «*» на основе Google Map API (Yandex Map API, OpenStreetMap API).
33. Принципы облачных технологий.
34. Сравнительный анализ PaaS-технологий.

35. Разработка сложных программных систем.
36. Сравнительный анализ способов обработки и хранения данных.
37. Методы обработки больших объемов данных (Big Data).
38. Проектирование и разработка desktop-приложения «*».
39. Проектирование и разработка web-системы (сервиса, портала) «*».
40. Проектирование и разработка web CRM-системы (сервиса, портала) «*».
41. Проектирование и разработка мобильного приложения «*».
42. Проектирование и разработка кроссплатформенного приложения «*».
43. Проектирование и разработка игровых алгоритмов «*» на движке Unity.
44. Разработка систем автоматизации с использованием Arduino (Raspberry Pi).
45. Разработка робототехники с использованием Arduino (Raspberry Pi).

Задачи курсового проекта:

1. Изучить теоретический материал по теме работы и особенности применения на практике полученных теоретических знаний по текущей дисциплине.
2. Решить прикладную задачу из конкретной предметной области, создав программный продукт (проект) в одной из объектно-ориентированных инструментальных сред.
3. Оформить результаты выполнения курсового проекта, в соответствии с предъявляемыми требованиями. Вместе с пояснительной запиской студент предъявляет файл исходного текста работающей программы (файлы исходных текстов работающих программ).

Тематику исследования можно сформулировать самостоятельно, предварительно согласовав с преподавателем.

Обучающимся может быть выбран любой стек ИТ-технологий, направленный на самостоятельное решение, программирование с использованием библиотек, анализ производительности, времени обучения, адекватности моделей и т.д.

Требования к оформлению курсового проекта приведены в Приложении 2. Максимальное количество баллов за проект – 100 баллов.

Критерии оценивания:

- 84-100 баллов (оценка «отлично») – разработанный интерфейс программы и ее функциональные возможности соответствуют требованиям задания курсового проекта; текстовое описание курсового проекта составлено в полном объеме и аккуратно; программа работает без сбоев для всех типовых задач, для которых она разрабатывалась, предусмотрена защита от ввода некорректных данных; во время защиты обучающийся показал свободное владение темой проекта, знание используемых компонентов, их свойств и специальных функций; изложенный материал фактически верен, наличие глубоких исчерпывающих знаний в объеме пройденной программы дисциплины в соответствии с поставленными программой курса целями и задачами обучения; правильные, уверенные действия по применению полученных знаний на практике, грамотное и логически стройное изложение материала при ответе, усвоение основной и знакомство с дополнительной литературой;

- 67-83 баллов (оценка «хорошо») – разработанный интерфейс программы и ее функциональные возможности соответствуют требованиям задания курсового проекта; текстовое описание курсового проекта составлено в не полном объеме; программа работает без сбоев для всех типовых задач, для которых она разрабатывалась, не предусмотрена защита от ввода некорректных данных; во время защиты студент показал свободное владение темой проекта, знание используемых компонентов, их свойств и специальных

функций; наличие твердых и достаточно полных знаний в объеме пройденной программы дисциплины в соответствии с целями обучения, четкое изложение материала, допускаются отдельные логические и стилистические погрешности, обучающийся усвоил основную литературу, рекомендованную в рабочей программе дисциплины;

- 50-66 баллов (оценка «удовлетворительно») – разработанный интерфейс программы и ее функциональные возможности соответствуют требованиям задания курсового проекта; текстовое описание курсового проекта составлено не в полном объеме и не достаточно аккуратно; программа работает без сбоев, не для всех типовых задач, для которых она разрабатывалась, не предусмотрена защита от ввода некорректных данных; во время защиты студент показал слабые знания по теме проекта; наличие твердых знаний в объеме пройденного курса в соответствии с целями обучения, изложение ответов с отдельными ошибками, уверенно исправленными после дополнительных вопросов;

- 0-49 баллов (оценка «неудовлетворительно») – разработанный интерфейс программы и ее функциональные возможности не соответствуют требованиям задания курсового проекта; текстовое описание курсового проекта составлено не в полном объеме и не аккуратно; программа работает со сбоями, не для всех типовых задач, для которых она разрабатывалась, не предусмотрена защита от ввода некорректных данных; во время защиты студент показал слабые знания по теме проекта; ответы не связаны с вопросами, наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

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

Процедуры оценивания включают в себя текущий контроль и промежуточную аттестацию.

Текущий контроль успеваемости проводится с использованием оценочных средств, представленных в п. 2 данного приложения. Результаты текущего контроля доводятся до сведения студентов до промежуточной аттестации.

Промежуточная аттестация проводится в форме защиты курсового проекта, экзамена.

Экзамен проводится по расписанию промежуточной аттестации в письменном виде. Количество вопросов в экзаменационном билете – 3. Проверка ответов и объявление результатов производится в день экзамена. Результаты аттестации заносятся в ведомость и зачетную книжку студента. Студенты, не прошедшие промежуточную аттестацию по графику, должны ликвидировать задолженность в установленном порядке.

Защита курсового проекта проводится по расписанию промежуточной аттестации.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Учебным планом предусмотрены следующие виды занятий:

- лекции;
- лабораторные занятия.

В ходе лекционных занятий рассматриваются теоретические вопросы с учетом практико-ориентированности изучаемой дисциплины, даются рекомендации для самостоятельной работы и подготовке к лабораторным занятиям.

В ходе лабораторных занятий углубляются и закрепляются знания студентов по ряду рассмотренных на лекциях вопросов, развиваются навыки программирования, применения интеллектуальных методов и инструментария разработки программных проектов.

При подготовке к лабораторным занятиям каждый обучающийся должен:

- изучить рекомендованную учебную литературу;
- изучить практические примеры, рассмотренные на лекциях.

В процессе подготовки к лабораторным занятиям студенты могут воспользоваться консультациями преподавателя.

Вопросы, не рассмотренные на лекциях и лабораторных занятиях, должны быть изучены студентами в ходе самостоятельной работы. Контроль самостоятельной работы студентов над учебной программой курса осуществляется в ходе занятий методом опроса или при выполнении лабораторных заданий с учетом индивидуальности представленного решения. В ходе самостоятельной работы каждый студент обязан прочитать основную и по возможности дополнительную литературу по изучаемой теме, дополнить конспекты лекций недостающим материалом, выписками из рекомендованных первоисточников. Выделить непонятные термины, найти их значение в энциклопедических словарях.

Для подготовки к занятиям, текущему контролю и промежуточной аттестации студенты могут воспользоваться электронно-библиотечными системами. Также обучающиеся могут взять на дом необходимую литературу на абонементе университетской библиотеки или воспользоваться читальными залами.

Методические рекомендации по оформлению курсового проекта.

Задание на курсовой проект определяет научный руководитель в соответствии с ее темой. Выполнение задания требует от студента теоретических знаний и умений осуществлять на практике программирование и проектирование на основе объектно-ориентированной парадигмы.

Структура курсового проекта и требования к его оформлению:

1. Титульный лист
2. Содержание (с указанием страниц)
3. ВВЕДЕНИЕ
4. Основной раздел (с разбиением на пункты)
5. ЗАКЛЮЧЕНИЕ
6. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ
7. Приложение (Приложения)

Во введении определяется актуальность проблемы, предмет (объект), цель и задачи исследования.

Основной раздел работы должен содержать результаты решения конкретной задачи программной инженерии.

Заключение в курсовом проекте включает обсуждение полученных результатов, приводятся выводы по работе. Выводы должны полностью соответствовать цели работы и характеризовать ее результаты.

В список использованных источников включаются обязательно все используемые работы, ресурсы Интернет и др. по авторскому алфавиту. Список оформляется в соответствии с библиографическими требованиями.

Приложение(я) включается в работу в случае необходимости. В приложениях приводятся листинги исходных текстов программ с основными комментариями, большие блок-схемы алгоритмов, таблицы, экспериментальных данных, результатов работы программы и т.д., если они занимают слишком много места в основном разделе работы. Материал приложений должен способствовать более четкому изложению материала, иллюстрировать отдельные положения и результаты курсовой работы.

Общие требования к тексту проекта:

Язык изложения курсового проекта должен быть четким, ясным, изложение - логичным и последовательным.

Каждая таблица в тексте должна иметь общий заголовок, номер, четкое обозначение строк и столбцов. В тексте дается анализ таблицы. Приводится общая нумерация рисунков и отдельно — нумерация таблиц.

Курсовой проект оформляется на одной стороне листа А-4, листы скрепляются. Содержание начинается со второй страницы. Наименование пунктов (разделов) содержания должно точно соответствовать наименованию разделов курсового проекта. Каждый раздел начинается в курсовом проекте с новой страницы.

Текст курсового проекта выполняется шрифтом 12 шрифтом, выравнивание по ширине, межстрочный интервал — 1,5. Текст печатается на одной стороне листа. Нумерация страниц выполняется внизу страницы справа в рамке во втором столбце, титульный лист (первая страница) не нумеруется.

Представление и защита курсового проекта

Законченный курсовой проект должен быть сдан студентом в установленный срок на кафедру. По результатам защиты выставляется оценка по курсовому проекту.

Порядок защиты:

1. Выступление студента, выполнившего курсовой проект (длительность выступления - 3-5 мин). Демонстрация работы программного продукта, созданного при выполнении курсового проекта (длительность демонстрации - 5 - 10 мин).

2. Вопросы

3. Выступление научного руководителя — не более 2-х мин.

Оценка выставляется после обсуждения результатов защиты. Отмечаются достоинства и недостатки работы.

Общие требования к реализации практической части курсового проекта:

В практической части курсового проекта обязательно должно быть реализовано следующее:

1. Техническое задание на проект, включая:

- Основания для разработки
- Назначение разработки и область применения
- Требования к программным и аппаратным средствам
- Входные и выходные данные
- Требования к основным режимам работы
- Требования к программной документации.
- Стадии и этапы разработки.

- Порядок контроля и приемки

2. Спроектирована UML-модель проекта (диаграммы прецедентов, диаграммы деятельности, диаграммы классов, диаграммы последовательностей действий, диаграммы компонентов).

3. Обоснование выбора языка и среды разработки.

4. Программно реализована решаемая задача. Даже в теоретических темах обязательно должны быть примеры программной реализации на любом языке программирования.

Программный код должен содержать обработчики ошибок (логических, ввода/вывода), используя условия и конструкции `try {} catch {}`.

Программная реализация проекта предполагает наличие пользовательского меню. Необходимо привести прототипы всех интерфейсов проекта.

Чтение/запись данных проекта может осуществляться через структуру файлов или базу данных .

5. Приведен листинг программного кода (или его фрагмент) в Приложении курсового проекта.

6. Произведено ручное (модульное Unit) тестирование разработанного программного проекта с демонстрацией экранов по реализации проекта. Примеров должно быть несколько, демонстрирующих различные возможности работы проекта.

7. Расчет экономической эффективности проекта.

8. Анализ рисков проекта.

9. Диаграмма Ганта для отображения календарного графика проекта.

10. Управление проектом (например, технология GIT).