

МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»
(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова)

УТВЕРЖДАЮ
 Декан факультета

 (подпись) Матвеев П.В.
 ФИО
 «___» _____ 20__

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ МЕТОДОЛОГИЯ ПРОГРАММНОЙ ИНЖЕНЕРИИ

Направление/специальность подготовки	09.04.04 Программная инженерия
Специализация/профиль/программа подготовки	Процессы и методы разработки программных продуктов
Уровень высшего образования	Магистратура
Форма обучения	Заочная
Факультет	О Естественнонаучный
Выпускающая кафедра	О7 Информационные системы и программная инженерия
Кафедра-разработчик рабочей программы	О7 Информационные системы и программная инженерия

КУРС	СЕМЕСТР	ОБЩАЯ ТРУДОЁМКОСТЬ (ЗАЧЕТНЫХ ЕДИНИЦ)	ЧАСЫ (по наличию видов занятий)									ВИД ПРОМЕЖУТОЧНОГО КОНТРОЛЯ
			ОБЩАЯ ТРУДОЁМКОСТЬ	АУДИТОРНЫЕ ЗАНЯТИЯ				САМОСТОЯТЕЛЬНАЯ РАБОТА				
				ВСЕГО	ЛЕКЦИИ	ЛАБОРАТОРНЫЙ ПРАКТИКУМ	ПРАКТИЧЕСКИЕ ЗАНЯТИЯ	ВСЕГО	КУРСОВОЙ ПРОЕКТ	КУРСОВАЯ РАБОТА	ДРУГИЕ ВИДЫ САМОСТ. РАБОТЫ	
1	1	6	216	8	2	0	6	208	0	18	190	ЭКЗ.

ЛИСТ СОГЛАСОВАНИЯ

РАБОЧАЯ ПРОГРАММА СОСТАВЛЕНА В СООТВЕТСТВИИ С ТРЕБОВАНИЯМИ ФЕДЕРАЛЬНОГО
ГОСУДАРСТВЕННОГО ОБРАЗОВАТЕЛЬНОГО СТАНДАРТА ВЫСШЕГО ОБРАЗОВАНИЯ (ФГОС ВО)

09.04.04 Программная инженерия

год набора группы: 2024

Программу составили:

Кафедра О7 Информационные системы и программная инженерия
Мажайцев Евгений Александрович, преподаватель

Кафедра О7 Информационные системы и программная инженерия
Семёнова Елена Георгиевна, д.т.н., профессор, профессор

Программа рассмотрена
на заседании кафедры-разработчика
рабочей программы **О7 Информационные системы и программная инженерия**

Заведующий кафедрой Семенова Е.Г., д.т.н., проф.

Программа рассмотрена
на заседании выпускающей кафедры

О7 Информационные системы и программная инженерия

Заведующий кафедрой Семенова Е.Г., д.т.н., проф.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ МЕТОДОЛОГИЯ ПРОГРАММНОЙ ИНЖЕНЕРИИ

Разделы рабочей программы

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ
2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВО
3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ
4. ФОРМЫ КОНТРОЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ
5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ
6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Приложения к рабочей программе дисциплины

- Приложение 1. Аннотация рабочей программы
- Приложение 2. Технологии и формы обучения
- Приложение 3. Фонды оценочных средств

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения дисциплины является формирование следующих компетенций:

ОПК-1 — способность самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте
ПСК-2.1 — способность выполнить постановку новых задач анализа и синтеза новых проектных решений

Формированию компетенций служит достижение следующих результатов образования:

ОПК-1

знания:

На уровне представлений: общее понятие о предметной области программной инженерии; понятие о стандартизации и лучших практиках программной инженерии; понятие качества программных средств.

На уровне воспроизведения: понятие жизненного цикла программных средств; понятие модели жизненного цикла программных средств; понятие процесса и проекта в программной инженерии.

На уровне понимания: взаимосвязь программной и системной инженерии; подходы к классификации задач разработки программных средств; подходы к классификации программного обеспечения;;

умения:

Теоретические: анализировать предметную область для использования программных средств.

Практические: определять требования к элементам показателей качества программных средств;;

навыки:

Формулирования целей разработки программных средств; предварительной оценки возможности разработки или сопровождения программных средств с учетом имеющейся информации о ресурсных

ограничениях и известных ограничений предметной области; предварительной оценки требуемых

ресурсов для обеспечения возможности разработки или сопровождения программных средств с учетом

известных ограничений предметной области;.

ПСК-2.1

знания:

На уровне представлений: понятие о стандартизации и лучших практиках программной инженерии.

На уровне воспроизведения: понятие процесса и проекта в программной инженерии.

На уровне понимания: подходы к классификации задач разработки программных средств; подходы к классификации программного обеспечения;;

умения:

Теоретические: анализировать предметную область для использования программных средств; определять цели и задачи разработки программных средств.

Практические: определять модель жизненного цикла программного средства; определять требования к элементам показателей качества программных средств;;

навыки:

Формулирования целей разработки программных средств; предварительной оценки требуемых ресурсов для обеспечения возможности разработки или сопровождения программных средств с учетом

известных ограничений предметной области;.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВО

Дисциплина **МЕТОДОЛОГИЯ ПРОГРАММНОЙ ИНЖЕНЕРИИ** является дисциплиной **обязательной части блока 1** программы подготовки по направлению *09.04.04 Программная инженерия*.

Содержание дисциплины является логическим продолжением содержания физико-математической подготовки бакалавра и служит основой для освоения дисциплин: **МЕТОДЫ ВИРТУАЛИЗАЦИИ И ИЗОЛЯЦИИ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ, ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ, МЕТОДОЛОГИЯ НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ ДЕЯТЕЛЬНОСТИ, ТЕОРИЯ ВЕРИФИКАЦИИ И ВАЛИДАЦИИ ПАРАЛЛЕЛЬНЫХ И РАСПРЕДЕЛЕННЫХ ПРОГРАММНЫХ СИСТЕМ, ВЫПОЛНЕНИЕ И ЗАЩИТА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ**

3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 6 з.е., 216 ч.

3.1. Содержание (дидактика) дисциплины

КУРС	СЕМЕСТР	Наименование разделов и дидактических единиц	ВСЕГО	Аудиторные занятия в контактной форме			Самостоятельная работа студентов	Формируемая компетенция, %	
				ВСЕГО	Лекции	Практические занятия		ОПК-1	ПСК-2.1
1	1	Раздел 1. Основные концепции и понятия программной инженерии. 1.1 Программная инженерия как понятие. 1.2 Проблемы терминологии программной инженерии. 1.3 Программные средства, программное обеспечение, программные системы, программные продукты и программные услуги. Процесс и проект. 1.4 Связь программной инженерии и системной инженерии. 1.5 Жизненный цикл программных средств. Модель жизненного цикла. 1.6 Методы программной инженерии.	53	1	1	0	52	25	25
1	1	Раздел 2. Предметная область программной инженерии. 2.1 Классификация задач разработки программных средств. 2.2 Интерфейсы программных средств. Интерфейсы пользователя и программные интерфейсы. 2.3 Классификация программных средств и программных услуг. 2.4 Программная инженерия как область деятельности и как направление подготовки.	55	3	1	2	52	25	25
1	1	Раздел 3. Оценка качества программных средств. 7.1 Процесс оценки качества. 7.2 Уровни оценки качества. 7.3 Методы оценки качества.	54	2	0	2	52	25	25
1	1	Раздел 4. Автоматизация процессов жизненного цикла программных средств. 8.1 Принципы автоматизация процессов жизненного цикла программных средств. 8.2 Инструментальные средства автоматизации процессов ЖЦ ПС.	54	2	0	2	52	25	25
Всего за 1 семестр			216	8	2	6	208	100	100
Всего по дисциплине			216	8	2	6	208	100	100

3.2. Аудиторный практикум

№ п/п	Номер и наименование раздела дисциплины	Тема практического занятия	Объем, ауд. часов
1	Раздел 2. Предметная область программной инженерии.	Классификация задач разработки программных средств. Интерфейсы программных средств. Интерфейсы пользователя и программные интерфейсы.	1
2		Классификация программных средств и программных услуг. Программная инженерия как область деятельности и как направление подготовки.	1
3	Раздел 3. Оценка качества программных средств.	Процесс оценки качества программных средств.	2
4	Раздел 4. Автоматизация процессов жизненного цикла программных средств.	Инструментальные средства автоматизации процессов ЖЦ ПС.	2
Всего за 1 семестр			6

3.3. Самостоятельная работа студента (СРС)

№ п/п	Номер и наименование раздела дисциплины	Содержание учебного задания	Объем, часов
1	Раздел 1. Основные концепции и понятия программной инженерии.	Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе.	12
2		Выполнение первого этапа курсовой работы	40
3	Раздел 2. Предметная область программной инженерии.	Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе.	13
4		Выполнение этапа 2 курсовой работы.	39
5	Раздел 3. Оценка качества программных средств.	Выполнение этапа 3 курсовой работы.	36
6		Изучение предусмотренных программой дидактических единиц по рекомендуемой	16

		литературе.	
7	Раздел 4. Автоматизация процессов жизненного цикла программных средств.	Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе.	10
8		Выполнение 4 этапа курсовой работы	42
Всего за 1 семестр			208

3.4. Курсовая работа

СОДЕРЖАНИЕ ЭТАПА	ПЕРИОД ИСПОЛНЕНИЯ (недели семестра)	ПЛАНИРУЕМОЕ ВРЕМЯ (час)
Этап 2. Предварительный анализ требований к программному средству, оценка изменчивости требований. Выявление требований, существенно определяющих качество программного средства	3 - 6	3
Этап 3. Выделение значимых характеристик и подхарактеристик качества, формирование показателей качества	7 - 9	3
Этап 4. Определение модели ЖЦ ПС как совокупности стадий и их соотношений во времени, отбор процессов ЖЦ ПС для стадий	10 - 14	5
Этап 5. Соотнесение процессов ЖЦ ПС с показателями качества. Выбор уровней и методов оценки элементов показателей качества на основании требований. Оформление пояснительной записки, подготовка к защите и защита курсовой работы	15 - 17	5
Этап 1. Анализ задачи, классификация рассматриваемого программного средства	1 - 2	2
Всего за 1 семестр		18

4. ФОРМЫ КОНТРОЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

СЕМЕСТР	НЕДЕЛИ СЕМЕСТРА																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1		КР				ДР			КР	ДР	КР		КР, Вопр. Экз				

Условные обозначения:

- ДР – диагностическая работа;
- КР – курсовая работа;
- Вопр. Экз – вопросы к экзамену.

Текущий контроль успеваемости студентов проводится в дискретные временные интервалы в следующих формах:

- диагностическая работа;
- курсовая работа;
- вопросы к экзамену.

Промежуточная аттестация проводится в формах:

- экзамен.

5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

5.1. Основная литература по дисциплине:

1. А. А. Москвитин. . Данные, информация, знания: методология, теория, технологии. Санкт-Петербург: Лань, 2022, эл. рес.
2. А. В. Остроух, Н. Е. Суркова. . Проектирование информационных систем. Санкт-Петербург: Лань, 2021, эл. рес.
3. А. Д. Хомоненко, А. Г. Басыров, В. П. Бубнов. . Модели и методы исследования информационных систем. Санкт-Петербург: Лань, 2022, эл. рес.
4. Б. Я. Советов, В. В. Цехановский. . Информационные технологии: теоретические основы. Санкт-Петербург: Лань, 2022, эл. рес.
5. В. К. Волк. . Практическое введение в программную инженерию. Санкт-Петербург: Лань, 2022, эл. рес.
6. Е. М. Лаврищева. . Программная инженерия и технологии программирования сложных систем. Москва: Юрайт, 2022, эл. рес.
7. Е. Р. Пантелеев. . Методы научных исследований в программной инженерии. Санкт-Петербург: Лань, 2021, эл. рес.
8. И. В. Ватаманюк, Д. К. Левоневский, Д. А. Малов. . Модели и способы взаимодействия пользователя с киберфизическим интеллектуальным пространством. Санкт-Петербург: Лань, 2022, эл. рес.
9. Н. А. Соловьёв, Е. Н. Чернопрудова, Н. А. Тишина. . Исследование операций в задачах программной инженерии. Санкт-Петербург: Лань, 2022, эл. рес.
10. Т. В. Гвоздева, Б. А. Баллод. . Проектирование информационных систем. Стандартизация. Санкт-Петербург: Лань, 2021, эл. рес.
11. Т. В. Гвоздева, Б. А. Баллод. . Проектирование информационных систем: технология автоматизированного проектирования. Санкт-Петербург: Лань, 2020, эл. рес.
12. Ю. П. Ехлаков. . Управление программными проектами. Стандарты, модели. Санкт-Петербург: Лань, 2022, эл. рес.

5.2. Дополнительная литература по дисциплине:

не требуется.

5.3. Периодические издания:

не требуются.

5.4. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины, электронные библиотечные системы:

1. <https://e.lanbook.com/> — ЭБС Лань;
2. <https://ibooks.ru/> — ЭБС Айбукс.ру - это большой выбор актуальной литературы для вашей библиотеки в электронном виде;
3. http://library.voenmeh.ru/jirbis2/index.php?option=com_irbis&view=irbis&Itemid=474 — Фундаментальная библиотека БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова;
4. <http://docs.cntd.ru> — Электронный фонд правовой и нормативно-технической информации - docs.cntd.ru;
5. <https://urait.ru/> — Главная – Образовательная платформа Юрайт. Для вузов и ссузов.;
6. <https://www.uml.org/> — Welcome To UML Web Site!;
7. <https://www.eclipse.org/papyrus/> — Papyrus;
8. <https://cruise.umple.org/umple/> — Umple: Merging Modeling with Programming.

Современные профессиональные базы данных:

1. <https://rusneb.ru> – Национальная электронная библиотека (НЭБ);
2. <https://cyberleninka.ru/> - Научная электронная библиотека «Киберленинка»;
<http://www.rfbr.ru/rffi/ru/library> - Полнотекстовая электронная библиотека Российского фонда фундаментальных исследований.

Информационные справочные системы:

1. Техэксперт – Информационный портал технического регулирования: Нормы, правила, стандарты РФ;
2. http://library.voenmeh.ru/jirbis2/index.php?option=com_irbis&view=irbis&Itemid=457 - БД ГОСТов собственной генерации БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова;
3. <http://www.consultant.ru/> - КонсультантПлюс- информационный портал правовой информации.

5.5. Программное обеспечение:

1. Kubuntu 18.04 LTS;
2. Офисный пакет Libre Office;
3. Набор средств трансляции, компоновки и отладки GCC/GNU Make/GDB;
4. Набор библиотек, средств трансляции, компоновки, отладки и интегрированных средств разработки Qt for Application Development;
5. Интегрированная среда разработки Code::Blocks;
6. Интегрированная среда разработки Eclipse IDE;
7. Набор средств трансляции, компоновки, отладки и выполнения Perl;
8. Набор средств трансляции, компоновки, отладки и выполнения Haskell;
9. Набор средств трансляции, компоновки, отладки и выполнения SWI-Prolog;
10. Набор средств трансляции, компоновки, отладки и выполнения CLIPS;
11. Набор средств трансляции, компоновки, отладки и выполнения Python 3.x с интегрированной средой разработки IDLE;
12. Интернет-браузер Mozilla Firefox;
13. Интернет-браузер Chromium;
14. Текстовый редактор Kate;
15. Процессор документов LyX;
16. Файловый менеджер Krusader;
17. Менеджер архивов с комплектом утилит архивации/разархивации Ark;
18. Редактор диаграмм и моделей программного обеспечения Dia;
19. Генератор исходных текстов программ на основе моделей dia2code;
20. Среда разработки на основе моделей программного обеспечения Umple;
21. Среда построения моделей программного обеспечения Papyrus;
22. Среда построения моделей программного обеспечения MODELIO;
23. Среда построения моделей программного обеспечения Umbrello.

5.6. Информационные технологии:

взаимодействие с обучающимися посредством ЭИОС Moodle БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова.

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Лекционные занятия:

специализированные требования по оборудованию отсутствуют; аудитория с посадочными местами по количеству студентов; доска.

6.2. Практические занятия:

1. Kubuntu 18.04 LTS;
2. Офисный пакет Libre Office;
3. Набор средств трансляции, компоновки и отладки GCC/GNU Make/GDB;
4. Набор библиотек, средств трансляции, компоновки, отладки и интегрированных средств разработки Qt for Application Development;
5. Интегрированная среда разработки Code::Blocks;
6. Интегрированная среда разработки Eclipse IDE;
7. Набор средств трансляции, компоновки, отладки и выполнения Perl;
8. Набор средств трансляции, компоновки, отладки и выполнения Haskell;
9. Набор средств трансляции, компоновки, отладки и выполнения SWI-Prolog;
10. Набор средств трансляции, компоновки, отладки и выполнения CLIPS;
11. Набор средств трансляции, компоновки, отладки и выполнения Python 3.x с интегрированной средой разработки IDLE;
12. Интернет-браузер Mozilla Firefox;
13. Интернет-браузер Chromium;
14. Текстовый редактор Kate;
15. Процессор документов LyX;
16. Файловый менеджер Krusader;
17. Менеджер архивов с комплектом утилит архивации/разархивации Ark;
18. Редактор диаграмм и моделей программного обеспечения Dia;
19. Генератор исходных текстов программ на основе моделей dia2code;
20. Среда разработки на основе моделей программного обеспечения Umple;
21. Среда построения моделей программного обеспечения Parugus;
22. Среда построения моделей программного обеспечения MODELIO;
23. Среда построения моделей программного обеспечения Umbrello.

6.3. Прочее:

1. рабочее место преподавателя, оснащенное компьютером с доступом в Интернет;
2. рабочие места студентов, оснащенные компьютерами с доступом в Интернет, предназначенные для работы в электронной образовательной среде.

Аннотация рабочей программы

Дисциплина **МЕТОДОЛОГИЯ ПРОГРАММНОЙ ИНЖЕНЕРИИ** является дисциплиной **обязательной части блока 1** программы подготовки по направлению *09.04.04 Программная инженерия*. Дисциплина реализуется на факультете *О Естественнотехнический БГТУ "ВОЕНМЕХ"* им. Д.Ф. Устинова кафедрой *О7 Информационные системы и программная инженерия*.

Дисциплина нацелена на формирование *компетенций*:

ОПК-1 способность самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте;

ПСК-2.1 способность выполнить постановку новых задач анализа и синтеза новых проектных решений.

Содержание дисциплины охватывает круг вопросов, связанных с методологическими основами и целями программной инженерии, организацией процессов программной инженерии и методами их осуществления. Рассматриваются вопросы определения понятия качества программных средств, общие принципы формулирования требований к качеству и оценки качества программных средств. Основное внимание отводится вопросам организации жизненного цикла программных средств, процессам и моделям жизненного цикла программных средств.

Программой дисциплины предусмотрены следующие **виды контроля**:

Текущий контроль успеваемости студентов проводится в дискретные временные интервалы в следующих формах:

- диагностическая работа;
- курсовая работа;
- вопросы к экзамену.

Промежуточная аттестация проводится в формах:

- экзамен.

Общая трудоемкость освоения дисциплины составляет **6 з.е., 216 ч.** Программой дисциплины предусмотрены лекционные занятия (**2 ч.**), практические занятия (**6 ч.**), самостоятельная работа студента (**208 ч.**).

ТЕХНОЛОГИИ И ФОРМЫ ОБУЧЕНИЯ

Рекомендации по освоению дисциплины для студента

Трудоемкость освоения дисциплины составляет 216 ч., из них 8 ч. аудиторных занятий, и 208 ч., отведенных на самостоятельную работу студента.

Рекомендации по распределению учебного времени по видам самостоятельной работы и разделам дисциплины приведены в таблице.

Контроль освоения дисциплины производится в соответствии с Положением о текущем, рубежном контроле успеваемости и промежуточной аттестации обучающихся.

Формы контроля и критерии оценивания приведены в приложении 3 к Рабочей программе.

Наименование работы	Рекомендуемая литература	Трудоемкость, час.
Раздел 1. Основные концепции и понятия программной инженерии.		
Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе.	<p>А. В. Остроух, Н. Е. Суркова. . Проектирование информационных систем: Санкт-Петербург: Лань, 2021 (2.3, 2.4)</p> <p>В. К. Волк. . Практическое введение в программную инженерию: Санкт-Петербург: Лань, 2022 (1.1-1.3, 2.1-2.3, приложение Г)</p> <p>Е. Р. Пантелеев. . Методы научных исследований в программной инженерии: Санкт-Петербург: Лань, 2021 (2.1-2.3)</p> <p>Б. Я. Советов, В. В. Цехановский. . Информационные технологии: теоретические основы: Санкт-Петербург: Лань, 2022 (6.1)</p> <p>Е. М. Лаврищева. . Программная инженерия и технологии программирования сложных систем: Москва: Юрайт, 2022 (2.1, приложение 4)</p> <p>И. В. Ватаманюк, Д. К. Левоневский, Д. А. Малов. . Модели и способы взаимодействия пользователя с киберфизическим интеллектуальным пространством: Санкт-Петербург: Лань, 2022 (1, 2)</p>	12
Выполнение первого этапа курсовой работы	<p>Ю. П. Ехлаков. . Управление программными проектами. Стандарты, модели: Санкт-Петербург: Лань, 2022 (1.1, 1.2, 2.1-2.3)</p>	40
Итого по разделу 1		52
Раздел 2. Предметная область программной инженерии.		
Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе.	<p>В. К. Волк. . Практическое введение в программную инженерию: Санкт-Петербург: Лань, 2022 (Приложения Б, В)</p> <p>А. В. Остроух, Н. Е. Суркова. . Проектирование информационных систем: Санкт-Петербург: Лань, 2021 (1.2, 1.3)</p> <p>Ю. П. Ехлаков. . Управление программными проектами. Стандарты, модели: Санкт-Петербург: Лань, 2022 (1.1, 4.1-4.4, 5.1, 6.1, 7.1, 8.1)</p> <p>Б. Я. Советов, В. В. Цехановский. . Информационные технологии: теоретические основы: Санкт-Петербург: Лань, 2022 (2.1-2.5, 3.1-3.9, 4.1, 5.1, 6.1, 6.2)</p> <p>Е. М. Лаврищева. . Программная инженерия и технологии программирования сложных систем: Москва: Юрайт, 2022 (1.1-1.3, 2.1, 12.1-12.3, приложение 4)</p>	13
Выполнение этапа 2 курсовой работы.	<p>Н. А. Соловьёв, Е. Н. Чернопрудова, Н. А. Тишина. . Исследование операций в задачах программной инженерии: Санкт-Петербург: Лань, 2022 (10.1-10.3, 13.1-13.4)</p>	39

Итого по разделу 2		52
Раздел 3. Оценка качества программных средств.		
Выполнение этапа 3 курсовой работы.	В. К. Волк. . Практическое введение в программную инженерию: Санкт-Петербург: Лань, 2022 (1.2, 1.3, приложение А) Е. М. Лаврищева. . Программная инженерия и технологии программирования сложных систем: Москва: Юрайт, 2022 (6.1-6.4, 10.1-10.3)	36
Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе.	Н. А. Соловьёв, Е. Н. Чернопрудова, Н. А. Тишина. . Исследование операций в задачах программной инженерии: Санкт-Петербург: Лань, 2022 (8.1-8.3, 9.1, 9.2, 10.1-10.3, 11.1-11.4, 12.1, 12.2, 13.1-13.4, 14.1, 14.2, 15.1, 16.1-16.3, 17.1-17.3) Т. В. Гвоздева, Б. А. Баллод. . Проектирование информационных систем. Стандартизация: Санкт-Петербург: Лань, 2021 (3.1) Б. Я. Советов, В. В. Цехановский. . Информационные технологии: теоретические основы: Санкт-Петербург: Лань, 2022 (6.3) А. Д. Хомоненко, А. Г. Басыров, В. П. Бубнов. . Модели и методы исследования информационных систем: Санкт-Петербург: Лань, 2022 (6.1-6.4, 10.1-10.3)	16
Итого по разделу 3		52
Раздел 4. Автоматизация процессов жизненного цикла программных средств.		
Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе.	Т. В. Гвоздева, Б. А. Баллод. . Проектирование информационных систем: технология автоматизированного проектирования: Санкт-Петербург: Лань, 2020 (1.1-1.4) А. А. Москвитин. . Данные, информация, знания: методология, теория, технологии: Санкт-Петербург: Лань, 2022 (п. 5) Б. Я. Советов, В. В. Цехановский. . Информационные технологии: теоретические основы: Санкт-Петербург: Лань, 2022 (6.2, 6.3)	10
Выполнение 4 этапа курсовой работы	А. В. Остроух, Н. Е. Суркова. . Проектирование информационных систем: Санкт-Петербург: Лань, 2021 (3.1-3.3, 4.1-4.4, 5.1, 5.2)	42
Итого по разделу 4		52

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Фонд оценочных средств, позволяющие оценить результаты обучения по данной дисциплине, включают в себя:

- диагностическая работа
- вопросы к экзамену;
- курсовая работа;
- экзамен.

Критерии оценивания

Диагностическая работа

Диагностическая работа проводится в форме теста в ЭИОС Moodle:

- при правильном ответе менее чем на 60% вопросов - не аттестация;
- при правильном ответе на 60% вопросов и более - аттестация.

Вопросы к экзамену

Перечень вопросов размещен в УМК дисциплины.

Курсовая работа

Тема курсовой работы: «Определение модели жизненного цикла программного средства».

Курсовая работа выполняется в соответствии с индивидуальным заданием.

Курсовая работа допускается к защите при следующих условиях:

- все пункты работы выполнены в соответствии с индивидуальным заданием;
- электронная и печатная версии пояснительной записки соответствуют установленным требованиям;
- предлагаемые решения корректно обоснованы в тексте пояснительной записки.

Оценка защиты курсовой работы производится по четырехбалльной шкале («отлично», «хорошо», «удовлетворительно» и «неудовлетворительно») на основании обсуждения порядка решения предусмотренных ее тематикой задач, включая проверку усвоения студентом соответствующих сведений из теории и степени самостоятельности при обосновании предлагаемых решений.

Экзамен

Итоговый контроль по дисциплине проходит в форме сочетания тестирования и письменного экзамена.

Основанием для сдачи экзамена является наличие сданной курсовой работы

Для получения оценки «удовлетворительно» необходимо пройти итоговое тестирование с рейтингом теста не менее 70%. При успешном прохождении тестирования на оценку «удовлетворительно», оценки «хорошо» и «отлично» выставляются по результатам ответа по билету. Экзаменационный билет письменного экзамена включает в себя теоретический вопрос и ситуативную задачу. Полный и развернутый ответ на каждый пункт экзаменационного билета соответствует одному дополнительному баллу к оценке «удовлетворительно»: развернутый ответ на два теоретических вопроса соответствует оценке «отлично», оценке «хорошо» соответствует либо полный ответ на одно из заданий билета, либо ответ на оба задания с существенными замечаниями, но позволяющими оценить данное задание как выполненное.

Для студентов, планомерно и успешно освоивших содержание учебной дисциплины, предусматривается возможность оформления экзаменационной оценки «хорошо» или «отлично» по результатам работы в семестре при следующих условиях:

- если курсовая работа защищена с оценкой «хорошо» или «отлично», а итоговое тестирование пройдено с рейтингом 90% и более – выставляется экзаменационная оценка «отлично»;
- если курсовая работа защищена с оценкой «хорошо» или «отлично», а итоговое тестирование пройдено с рейтингом 80% и более, но менее 90% – выставляется экзаменационная оценка, соответствующая оценке защиты курсовой работы.

В случае несогласия с предлагаемой оценкой студент сохраняет право сдавать экзамен по билету по расписанию экзаменационной сессии.

Паспорт фонда оценочных средств

КУРС	СЕМЕСТР	Наименование разделов и дидактических единиц	ВСЕГО	Аудиторные занятия в контактной форме			Самостоятельная работа студентов	Формируемая компетенция, %		НАИМЕНОВАНИЕ ОЦЕНОЧНОГО СРЕДСТВА
				ВСЕГО	Лекции	Практические занятия		ОПК-1	ПСК-2.1	
1	1	Раздел 1. Основные концепции и понятия программной инженерии.	53	1	1	0	52	25	25	Вопросы к экзамену
1	1	Раздел 2. Предметная область программной инженерии.	55	3	1	2	52	25	25	Вопросы к экзамену, Курсовая работа
1	1	Раздел 3. Оценка качества программных средств.	54	2	0	2	52	25	25	Вопросы к экзамену, Курсовая работа
1	1	Раздел 4. Автоматизация процессов жизненного цикла программных средств.	54	2	0	2	52	25	25	Вопросы к экзамену
Всего за 1 семестр			216	8	2	6	208	100	100	
Всего по дисциплине			216	8	2	6	208	100	100	

Критерии оценивания

ОПК-1

- Вопросы открытого типа:*
- № 1 Какие типы шкал выделяют в соответствии со стандартом ИСО/МЭК 15939 «Процессы измерения»?
- № 2 Согласно ГОСТ Р ИСО/МЭК 25010–2015, показатель внутреннего качества программного обеспечения (internal measure of software quality) — это...
- № 3 Дополните предложение:
- Серия международных стандартов «Информационные технологии. Системная и программная инженерия. Требования и оценка _____ систем и программного обеспечения (SQuaRE)» включает в себя _____ основных разделов.
- № 4 Согласно ГОСТ Р ИСО/МЭК 25010–2015, качество при использовании — это...
- № 5 Дополните предложение:
- Проверка корректности взаимодействий внутри отдельных групп компонентов— это
- № 6 Дополните предложение:
- Проверка работы системы в целом, выполнения ею своих основных функций, с использованием определенных ресурсов, в окружении с заданными характеристиками – это
- № 7 Что означает аббревиатура SWEBOK?
- № 8 Дополните предложение:
- Основной задачей нагрузочного тестирования является проверка производительности системы при высоких_____.
- № 9 Какие тенденции в разработке программного обеспечения вы считаете наиболее важными в настоящее время?
- № 10 Что такое DevOps и как он влияет на процесс разработки программного обеспечения?
- Вопросы закрытого типа:*
- № 1 К какой группе инструментов программной инженерии можно отнести редакторы программ, компиляторы, интерпретаторы и отладчики?
1. Инструменты работы с требованиями
 2. Инструменты проектирования
 3. Инструменты конструирования
 4. Инструменты тестирования
- № 2 Генераторы тестов, средства выполнения тестов, инструменты анализа тестов, средства управления тестами, инструменты анализа производительности — все используются в процессах:
1. Проектирования
 2. Верификации
 3. Конфигурационного управления
 4. Анализа требованиями
- № 3 Ролевые платформы разработки программного обеспечения, инструменты конфигурационного управления, инструменты управления проектами и инструменты моделирования — можно обобщенно назвать...
1. Инструментами поддержки процессов программной инженерии

2. Инструментами тестирования
 3. Инструментами конструирования
 4. Инструментами работы с требованиями
- № 4 Необходимость одновременного использования программ работы с "картами памяти" (MindMap), локальными базами знаний организации, построения моделей на языках визуального моделирования, создания прототипов пользовательских интерфейсов, создания формальных спецификаций, создания и хранения заметок, текстовых и графические редакторов и процессоров скорее всего может возникать в процессах:
1. Анализа требований, проектирования архитектуры и детального проектирования программных средств
 2. Детального проектирования, конструирования и функционирования программных средств
 3. Анализа требований, проектирования архитектуры и аудита программных средств
 4. Верификации, валидации и аудита программных средств
- № 5 К каскадным моделям жизненного цикла программных систем можно отнести:
1. Спиральную модель Боэма, модель RUP (Rational Unified Process)
 2. RAD-модель («быстрая разработка приложений»), итерационная модель
 3. Каскадную (водопадную) модель, «водопад с возвратами», V-образную модель (V-Model, VEE модель), двойная V-модель (Dual Vee Model)
 4. Инкрементную модель, XP (экстремальное программирование)
- № 6 К спиральным моделям жизненного цикла программных систем можно отнести:
1. «Водопад с возвратами», V-образную модель (V-Model, VEE модель)
 2. Спиральную модель Боэма, модель RUP (Rational Unified Process)
 3. «Водопад с возвратами», диаграммы Лейбница, червячная передача Больцмана, двойная V-модель (Dual Vee Model)
 4. Модель «программирование – отладка»
- № 7 Сопоставьте соответствующие высказывания SWEBOK
1. Определение и управление требованиями, предъявляемыми к программному продукту
 2. Оценка и улучшение качества программного продукта, включая процессы и стандарты
 3. Анализ требований и спецификаций программных продуктов
 4. Управление изменениями и доработками программного продукта после его развертывания
 5. Исследование и разработка методов для улучшения процессов разработки программного обеспечения
- А- Software Process
- Б- Software Maintenance
- В- Software Quality
- Г- Software Requirements
- Д- Software Testing
- № 8 В соответствии со стандартом ИСО/МЭК 15939 «Процессы измерения», информационная потребность (information need) — это...
1. Информация, необходимая пользователю для приобретения знаний

2. Понимание, необходимое для управления целями, задачами, рисками и проблемами
 3. Понимание пользователем, зачем ему информация
 4. Основание пирамиды потребностей Маслоу
- № 9 Неформально, модель жизненного цикла ПО — это...
1. Структура данных о выполняемых процессах жизненного цикла программного обеспечения
 2. Алгоритм действий по разработки программного обеспечения
 3. Структура, определяющая последовательность выполнения и взаимосвязи процессов, действий и задач на протяжении жизненного цикла
 4. Взаимосвязанная совокупность алгоритмов, используемых в процессах проектирования, конструирования и сопровождения программного обеспечения
- № 10 Какие разделы SWEBOOK посвящены управлению проектами?
1. Раздел 3 (Software Engineering Process)
 2. Раздел 9 (Software Engineering Management)
 3. Раздел 5 (Software Engineering Modeling)
 4. Раздел 8 (Software Quality)
 5. SWEBOOK не содержит разделов, посвященных управлению проектами

ПСК-2.1

- Вопросы открытого типа:*
- № 1 Что такое принцип SOLID в объектно-ориентированном программировании?
- № 2 Какие преимущества может предоставить использование паттернов проектирования в разработке программного обеспечения?
- № 3 Что представляет собой принцип DRY в программировании?
- № 4 В чем состоит преимущество микросервисной архитектуры перед монолитной?
- № 5 Дополните предложение:
- _____ это технология, позволяющая упаковывать приложения и их зависимости в изолированные контейнеры. Это обеспечивает удобство развертывания, масштабируемость и портативность приложений.
- № 6 Какие методы обеспечения безопасности широко используются в современной разработке программного обеспечения?
- № 7 Дополните предложение:
- _____ тестирование предназначено для проверки работоспособности программных модулей и функций в отдельности.
- № 8 Дополните предложение:
- Применение шифрования данных помогает обеспечить конфиденциальность и _____ информации в системах.
- № 9 Дополните предложение:
- Принцип DRY (Don't Repeat Yourself) нацелен на избегание _____ кода и создание единственного источника правды.
- № 10 Дополните предложение:
- Контейнеры обеспечивают изоляцию приложений и их зависимостей для более легкого _____ и управления.
- Вопросы закрытого типа:*
- № 1 Проверка корректности работы отдельных компонентов системы, выполнения ими своих функций и предполагаемых проектом характеристик – это...
1. Модульное или компонентное тестирование
 2. Интеграционное тестирование
 3. Системное тестирование
 4. Нагрузочное тестирование

- № 2 Что означает "принцип инкрементального развития" в методологии разработки ПО?
1. Построение ПО по частям, с каждым этапом добавления новой функциональности
 2. Создание всего ПО сразу в одном большом цикле разработки
 3. Применение только одного языка программирования
- № 3 Какие компетенции могут быть улучшены с помощью изучения SWEBOK?
1. Только навыки программирования
 2. Управление проектами, архитектура программного обеспечения, тестирование и другие аспекты инженерии программного обеспечения
 3. Медицинские навыки и знания
- № 4 Какие из следующих практик относятся к тестированию в разработке ПО?
1. Модульное тестирование
 2. Анализ требований
 3. Интеграционное тестирование
 4. Проектирование баз данных
 5. Тестирование баз данных
- № 5 Какой метод тестирования ПО используется для проверки функциональности отдельных компонентов программы, таких как функции или методы?
1. Тестирование производительности
 2. Модульное тестирование
 3. Тестирование безопасности
 4. Тестирование совместимости
- № 6 Верно ли утверждение: "Методика белого ящика" (или структурное тестирование) основана на анализе внутренней структуры программы и проверке кода, а "методика черного ящика" (или функциональное тестирование) оценивает поведение программы без знания её внутренней структуры. "Белый ящик" проверяет внутреннюю логику и структуру программы, в то время как "черный ящик" сосредотачивается на внешних аспектах, таких как входы и выходы, для проверки соответствия спецификациям и ожидаемому поведению программы.
- № 7 Верно ли утверждение: "Тестирование вертикальной совместимости" оценивает, как программное обеспечение взаимодействует с разными версиями и конфигурациями операционных систем, браузеров и других факторов, в то время как "тестирование горизонтальной совместимости" фокусируется на проверке взаимодействия программы с различными компонентами или службами внутри одной конкретной версии операционной системы.
- № 8 Какие принципы лежат в основе методологии Scrum?
1. Непрерывная интеграция
 2. Итеративность и обратная связь
 3. Разделение на короткие спринты
 4. Линейная последовательность этапов
- № 9 Какие основные преимущества предоставляет методология Lean в программной инженерии?
1. Увеличение избыточных процессов
 2. Максимизация рабочего процесса
 3. Оптимизация через итерации и улучшения
 4. Увеличение сложности проекта
- № 10 Что представляет собой Continuous Integration (непрерывная интеграция) в контексте разработки ПО?
1. Постоянное тестирование кода
 2. Интеграция кода разработчиков в общую кодовую базу

3. Проверка кода только в конце проекта
4. Интеграция только одного типа кода в проекте