

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный технический университет»

УТВЕРЖДАЮ
Декан факультета экономики, менеджмента и
информационных технологий
 / Баркалов С.А./
31 августа 2021 г.

РАБОЧАЯ ПРОГРАММА
дисциплины
«Программная инженерия»

Направление подготовки 09.03.03 Прикладная информатика

Профиль Прикладная информатика в экономике цифрового общества

Квалификация выпускника бакалавр

Нормативный период обучения 4 года/ 4 года 11 месяцев

Форма обучения очная/заочная

Год начала подготовки 2021

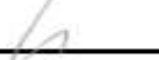
Автор программы

 /Минакова О.В./

И.о. заведующего кафедрой
систем управления и
информационных
технологий в строительстве

 / Десятирикова Е.Н./

Руководитель ОПОП

 / Аснина Н.Г./

Воронеж 2021

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины

формирование профессиональных компетенций по управлению разработкой надежного и качественного функционирования программного обеспечения в изменяющихся социально-экономических условиях

1.2. Задачи освоения дисциплины

- понимать технологии разработки программного обеспечения;
- получить опыт работы с инструментальными средами разработки программного обеспечения;
- развить навыки программирования с использованием сторонних библиотек и фреймворков;
- использовать методы отладки и тестирования программ, документирования и оценки качества программных продуктов

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Программная инженерия» относится к дисциплинам части, формируемой участниками образовательных отношений блока Б1.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Программная инженерия» направлен на формирование следующих компетенций:

ПК-2 - Способен разрабатывать техническую документацию на программное обеспечение

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ПК-2	знать: - методологии разработки программного обеспечения и технологии программирования, - принципы построения и виды архитектур программного обеспечения, - методы и приемы конструирования надежного и защищенного и качественного программного обеспечения
	уметь: - выполнять сбор, систематизация, выявление взаимосвязей и документирование требований к программному обеспечению - выбирать средства реализации требований к программному обеспечению - применять существующие типовые решения и шаблоны проектирования программного обеспечения; - осуществлять разработку технической документации на программное обеспечение по действующим стандартам

	владеть навыками: - верификации и валидации требований к программному обеспечению; - формирования и предоставления отчетности в соответствии с установленными регламентами; - использования инструментов совместной разработки ПО;
--	---

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Программная инженерия» составляет 5 з.е.

Распределение трудоемкости дисциплины по видам занятий
очная форма обучения

Виды учебной работы	Всего часов	Семестры	
		5	6
Аудиторные занятия (всего)	108	54	54
В том числе:			
Лекции	36	18	18
Практические занятия (ПЗ)	18	18	-
Лабораторные работы (ЛР)	54	18	36
Самостоятельная работа	36	18	18
Курсовой проект	+		+
Часы на контроль	36	-	36
Виды промежуточной аттестации - экзамен, зачет	+	+	+
Общая трудоемкость:			
академические часы	180	72	108
зач.ед.	5	2	3

заочная форма обучения

Виды учебной работы	Всего часов	Семестры	
		7	8
Аудиторные занятия (всего)	28	14	14
В том числе:			
Лекции	8	4	4
Практические занятия (ПЗ)	4	2	2
Лабораторные работы (ЛР)	16	8	8
Самостоятельная работа	139	54	85
Курсовой проект	+		+
Часы на контроль	13	4	9
Виды промежуточной аттестации - экзамен, зачет	+	+	+
Общая трудоемкость:			

академические часы	180	72	108
зач.ед.	5	2	3

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий

очная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Прак зан.	Лаб. зан.	СРС	Всего, час
1	Методологии разработки ПО	Понятие технологии программирования как разработки надежного ПО: Проблемы разработки программного обеспечения. Жизненный цикл ПО: Стандарты в области программной инженерии. Основные и вспомогательные процессы разработки ПО. Этапы разработки ПО. Эволюция моделей разработки ПО: каскадная и эволюционная модель. Итерационная и инкрементная разработка. Спиральная модель и оценка рисков. Управление программными проектами: Понятие проекта. Характеристика основных этапов программного проекта. Организация группы и принятие решений. Роли и ответственность в команде разработчиков. Построение сетевых графиков. Диаграмма Гранта. Слежение за состоянием проекта. Анализ рисков. Современные методологии разработки ПО и CASE-средства: Тяжеловесные и легковесные процессы. RUP, MFC. XP-программирование. Agile-техники: Scrum, Kanban. Понятие среды разработки (IDE, SDK) и CASE-средства. Механизмы интеграции инструментальных средств.	6	2	10	6	24
2	Инженерия требований	Понятие требования к ПО, виды и уровни требований, атрибуты и характеристики требований. Методы сбора, извлечения и анализа требований. Спецификация требований через пользовательские истории, сценарии использования, UML-диаграммы. Написание технического задания и спецификации. Инструменты и методы проверки требований на полноту, непротиворечивость, реализуемость. Характеристики качества ПО: Модель качества по ISO 9126. Характеристики и субхарактеристики качества программного средства. Метрики качества программного средства. Оценивание характеристик качества программных средств. Практика составления спецификации качества	6	2	10	6	24
3	Проектирование ПО	Архитектурное проектирование: Архитектура с общим репозиторием, клиент-серверная, многоуровневая. Модели централизованного и событийного управления. Архитектура канала и фильтра. Декомпозиция программной архитектуры: Структурный подход к разработке ПО. Абстракция и декомпозиция. Модульное программирование. Характеристики программного модуля. Связанность и	6	2	10	6	24

		сцепление. Пошаговая детализация. Иерархия модулей/функций. Объектно-ориентированное проектирование: Объектная декомпозиция системной архитектуры. Объекты и классы, зависимости. Основные понятия ООП. Разработка и оценка архитектуры на основе сценариев. UML. Статические и динамические диаграммы для архитектурных представлений					
4	Конструирование ПО	Современные методы конструирования: полиморфизм, обобщение, рефлексия, утверждения, синхронизация и совместный доступ. Применение паттернов проектирования и стилей кодирования для обеспечения качества программного кода. Защитное программирование. Применение принципов безопасного кодирования, использование механизмов обработки исключений и обеспечение отказоустойчивости. Инструменты и методы выявления и устранения ошибок в коде. Логирование. Использование сторонних библиотек и фреймворков для ускорения разработки и повышения качества кода.	6	4	8	6	24
5	Тестирование, верификация и оценка качества ПО	Тестирование – как инструмент разработки ПО: Категории программных ошибок. Виды тестирования. Функциональное и структурное тестирование. Методы эквивалентного разбиения, граничных условий, функциональных диаграмм. Тестирование базового пути, тестирования условий, циклов, комбинаторное покрытие условий. Unit-тестирование. Построение тестовых наборов и автоматизация тестирования. Тестирование модульное, интеграционное, системное, регрессионное и их связь с циклом разработки. Модели оценки надежности ПО. Методы проверки соответствия, оценки качества и просмотра кода. Квалификационное тестирование.	4	4	8	6	22
6	Выпуск, развертывание, эксплуатация и сопровождение ПО	Особенности сопровождения программных продуктов. Реинжиниринг. Унаследованные системы и реверсная инженерия. Управление конфигурацией: Идентификация, аудит и учет конфигураций. Релизы и управление версиями. Инструменты конфигурационного управления SVN, git. Принципы работы программных средств управления проектами, совместной разработки и управления конфигурацией. Документирование ПО: Стандарты разработки программной документации. Составление технических описаний, руководства пользователя. Практика использования JavaDocs и CodeStyle. Непрерывная интеграция и развертывание. Управление выпуском. Практики DevOps	8	4	8	6	26
Итого			36	18	54	36	144

заочная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Прак зан.	Лаб. зан.	СРС	Всего, час
1	Методологии разработки ПО	Понятие технологии программирования как разработки надежного ПО. Эволюция моделей разработки ПО. Современные методологии разработки ПО и CASE-средства: Тяжеловесные и легковесные процессы. RUP, MFC. XP-программирование. Agile-техники: Scrum, Kanban.	2	-	2	24	286
2	Инженерия требований	Понятие требования к ПО, виды и уровни требований, атрибуты и характеристики требований. Методы сбора, извлечения и анализа требований.	2	-	2	22	26
3	Проектирование ПО	Архитектурное проектирование: Архитектура с общим репозиторием, клиент-серверная, многоуровневая. Модели централизованного и событийного управления. Архитектура канала и фильтра.	2	-	2	24	28
4	Конструирование ПО	Современные методы конструирования: полиморфизм, обобщение, рефлексия, утверждения, синхронизация и совместный доступ. Применение паттернов проектирования и стилей кодирования для обеспечения качества программного кода. Защитное программирование. Применение принципов безопасного кодирования, использование механизмов обработки исключений и обеспечение отказоустойчивости. Инструменты и методы выявления и устранения ошибок в коде. Логирование. Использование сторонних библиотек и фреймворков для ускорения разработки и повышения качества кода.	-	2	6	22	30
5	Тестирование, верификация и оценка качества ПО	Тестирование – как инструмент разработки ПО: Функциональное и структурное тестирование. Методы эквивалентного разбиения, граничных условий, функциональных диаграмм. Тестирование базового пути, тестирования условий, циклов, комбинаторное покрытие условий. Unit-тестирование. Построение тестовых наборов и автоматизация тестирования. Тестирование модульное, интеграционное, системное, регрессионное и их связь с циклом разработки.	2	-	2	24	28
6	Выпуск, развертывание, эксплуатация и сопровождение ПО	Особенности сопровождения программных продуктов. Реинжиниринг. Унаследованные системы и реверсная инженерия. Управление конфигурацией: Идентификация, аудит и учет конфигураций. Релизы и управление версиями. Инструменты конфигурационного управления SVN, git. Принципы работы программных средств управления проектами, совместной разработки и управления конфигурацией. Документирование ПО: Стандарты разработки программной документации. Составление технических описаний,	-	2	2	23	27

		руководства пользователя. Практика использования JavaDocs и CodeStyle. Непрерывная интеграция и развертывание. Управление выпуском. Практики DevOps					
Итого			8	4	16	139	167

5.2 Перечень лабораторных работ

№ п/п	Тема и содержание	Трудоемкость	Вид контроля
1	Создание проекта в стиле MVC в среде разработки ПО	2	наблюдение
2	Построение графического интерфейса пользователя	2	отчет
3	Методы защитного программирования	2	отчет
4	Работа с ресурсами	2	отчет
5	Механизмы связывания программных компонентов	2	отчет
6	Реализация паттерна Стратегия	2	отчет
7	Обработка событий мыши	2	отчет
8	Реализация паттерна Строитель	2	отчет
9	Использование коллекций в реализации программ	2	наблюдение
10	Реализация Итератора для структур данных	2	отчет
11	Обобщенное программирование	2	отчет
12	Реализация паттерна Наблюдатель	4	отчет
13	Работа с наблюдаемыми коллекциями	2	отчет
14	Реализация паттерна Цепочка обязанностей	2	отчет
15	Логгирование	2	отчет
16	Подключение сторонних библиотек	2	отчет
17	Создание и выпуск документации (JavaDocs)	2	отчет
18	Реализация паттерна Компоновщик	4	отчет
19	Обработка исключений	2	отчет
20	Организация доступа к данным из различных источников (DAO)	2	отчет
21	Использование сторонних API	4	отчет
22	Написание unit-тестов	2	отчет
23	Автоматизация развертывания ПО	4	отчет
24	Оформление программного репозитория	2	отчет

6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ) И КОНТРОЛЬНЫХ РАБОТ

В соответствии с учебным планом освоение дисциплины предусматривает выполнение курсового проекта в 6 семестре для очной формы обучения, в 8 семестре для заочной формы обучения.

Примерная тематика курсового проекта:

1. Разработка комплекта программ для оценки расходов на строительные материалы
2. Разработка программного обеспечения для хранения личной информации
3. Разработка программного средства построения блоковых графиков по результатам анализа экспериментальных данных
4. Разработка программного комплекса управления микроклиматом помещений
5. Разработка программного модуля редактирования диаграмм классов для инструментальной среды разработки

Задачи, решаемые при выполнении курсового проекта:

- Анализ предметной области и существующих программных решений, формулировка требований к ПО, написание технического задания, тестовых сценариев;
 - Проектирование программной системы, включая создание пользовательского интерфейса и модели данных, моделирование последовательности взаимодействия пользователя с системой
 - Реализация программы, включая unit-тесты для основной логики, оформление кода и документации к программе
- Курсовой проект состоит в создании программного продукта, включая верификацию, и оформление расчетно-пояснительной записки.

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

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

7.1.1 Этап текущего контроля

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по следующей системе:

«аттестован»;

«не аттестован».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Аттестован	Не аттестован
ПК-2	знать методологии разработки программного обеспечения и технологии программирования; принципы построения и виды архитектур программного обеспечения; методы и приемы конструирования надежного и защищенного и качественного программного обеспечения	устный опрос	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь выполнять сбор, систематизация, выявление взаимосвязей и документирование требований к программному обеспечению; выбирать средства реализации требований к программному обеспечению; применять существующие типовые решения и шаблоны проектирования программного обеспечения; осуществлять разработку технической документации на программное обеспечение по действующим стандартам	выполнение стандартных практических задач	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть навыками верификации и валидации требований к программному обеспечению;	реализация программного продукта	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

	формирования и предоставления отчетности в соответствии с установленными регламентами; использования инструментов совместной разработки ПО;			
--	---	--	--	--

7.1.2 Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются в 5, 6 семестре для очной формы обучения, 7, 8 семестре для заочной формы обучения по двух/четырёхбалльной системе:

«зачтено»

«не зачтено»

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Зачтено	Не зачтено
ПК-2	знать методологии разработки программного обеспечения и технологии программирования; принципы построения и виды архитектур программного обеспечения; методы и приемы конструирования надежного и защищенного и качественного программного обеспечения	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	уметь выполнять сбор, систематизация, выявление взаимосвязей и документирование требований к программному обеспечению; выбирать средства реализации требований к программному обеспечению; применять существующие типовые решения и шаблоны проектирования программного обеспечения; осуществлять разработку технической документации на программное обеспечение по действующим стандартам	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть навыками верификации и валидации требований к программному обеспечению; формирования и предоставления отчетности в соответствии с установленными регламентами; использования инструментов совместной разработки ПО;	Решение прикладных задач в конкретной предметной области	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

или

«отлично»;

«хорошо»;

«удовлетворительно»;

«неудовлетворительно».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Отлично	Хорошо	Удовл.	Неудовл.
-------------	---	---------------------	---------	--------	--------	----------

ПК-2	знать методологии разработки программного обеспечения и технологии программирования; принципы построения и виды архитектур программного обеспечения; методы и приемы конструирования надежного и защищенного и качественного программного обеспечения	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	уметь выполнять сбор, систематизация, выявление взаимосвязей и документирование требований к программному обеспечению; выбирать средства реализации требований к программному обеспечению; применять существующие типовые решения и шаблоны проектирования программного обеспечения; осуществлять разработку технической документации на программное обеспечение по действующим стандартам	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть навыками верификации и валидации требований к программному обеспечению; формирования и предоставления отчетности в соответствии с установленными регламентами; использования инструментов совместной разработки ПО;	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

7.2 Примерный перечень оценочных средств (типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)

7.2.1 Примерный перечень заданий для подготовки к тестированию

1. *Формализованное описание процесса обработки данных – это ...*
 - a) программа
 - b) алгоритм
 - c) информационная среда
 - d) транслятор
2. *Программа или логически связанная совокупность программ на носителях данных, снабженная документацией называется*
 - a) программным средством
 - b) языком программирования
 - c) технологией программирования
 - d) информационным продуктом
3. *Программное средство состоит из*
 - a) программы на носителе данных и документации
 - b) набора программ
 - c) программ и данных, расположенных на носителе информации
 - d) совокупности логически связанных программ
4. *Способность программного средства безотказно выполнять определенные функции при заданных условиях называется*
 - a) надежность
 - b) качество
 - c) сопровождаемость
 - d) устойчивость
5. *Выберите верное утверждение*
 - a) Надежное программное средство не исключает наличия ошибки в нем
 - b) Если в программном средстве иногда возникают ошибки, то оно ненадежно
 - c) Для надежного ПС важно, чтобы в течение заданного периода времени ошибки не возникали
 - d) ПС надежное, если оно точно соответствует спецификации
6. *В ПС содержится ошибка если оно не соответствует*
 - a) тому, что пользователю разумно от него ожидать
 - b) своей функциональной спецификации
 - c) программной документации по применению
 - d) нефункциональным требованиям, установленным при разработке
7. *Путем тестирования можно убедиться, но ПС обладает свойством*
...
 - a) надежность
 - b) правильность
 - c) модульность
 - d) безошибочность

8. Технология программирования в широком смысле - это
- совокупность процессов разработки надежных программных средств
 - систематизированное описание методов построения программных средств
 - создание формализованного описания процесса обработки данных
 - методология разработки языков написания программ
9. Поставьте в соответствие этап жизненного цикла разработки ПС и его результат
- | | |
|----------------------------|--|
| a) Этап внешнего описания | A) Фиксация требований к ПС |
| b) Этап конструирования ПС | B) Разработка архитектуры, структуры программ и их детальная спецификация |
| c) Этап кодирования | C) Создание текстов на языках программирования, их отладка с тестированием |
| d) Этап аттестации | D) Оценка качества ПС |
10. Какие этапы жизненного цикла разработки ПС могут перекрываться, т.е. один может быть начат до завершения другого
- конструирования и кодирования
 - внешнего описания и конструирования
 - кодирования и аттестации
 - внешнего описания и аттестации
- Ответы: 1-8) a 9) a-A, b-B, c-C, d-D 10) a

7.2.2 Примерный перечень заданий для решения стандартных задач

- Составьте иерархию заинтересованных лиц и их ролей, используя метод опорных точек для программной системы «Метеостанция».
- Смоделируйте для каждого участника программного проекта варианты использования программы управления совместной работой.
- Постройте диаграммы вариантов использования и сформулируйте функциональных требований к программе «Графический редактор».
- Предложите архитектуру и постройте UML-диаграммы для ее представления для программной системы Калькулятор личных расходов.
- Спроектируйте графический интерфейс ПС для управления микроклиматом и представьте его раскадровку.
- Разработайте модель данных и управления ими для ПС планирования расписания работ и представьте схему физической БД и диаграмму состояний.
- Включите в ПС «Калькулятор» возможность изменения правил расчета.
- Создайте механизм защиты полей ввода, включающий контроль границ допустимых значений для любой ранее разработанной программы
- Напишите unit-тесты с использованием метода граничных значений для любой ранее разработанной программы.
- Составьте лист оценки характеристик качества программных средств для построения UML-диаграмм.

7.2.3 Примерный перечень заданий для решения прикладных задач

1. Реализовать ПС "Раскраска)", которое выполняет следующие функции:

- вывод контура изображений для закрашивания;
- выбор цвета, палитры, рисунка для заполнения фигуры (режим "закрашивание");
- последовательно закрашивание всех объектов выбранным цветом (режим "анимация");
- добавление на рисунок объектов из списка (режим "дорисовка").

В реализации должны быть использованы не менее 3 порождающих паттернов.

2. Реализовать ПС "Демонстрация работы со структурой данных" (стеки, деки, очереди, деревья) ПС должно осуществлять демонстрацию выполнения основных операций - вставки (добавления), удаления заданного элемента структуры, поиск заданного значения над отображаемой структурой данных. Для отображения структуры данных должен быть использован составной компонент (массив кнопок, меток и пр).

В реализации должен быть использован структурный паттерн.

3. Реализовать ПС "Строительный калькулятор". ПС должно осуществлять расчет количества плитки для заданного помещения. В реализации должны быть использованы – Абстрактный суперкласс, Делегирование.

4. Создайте телеграм-бот для проведения контроля знаний с тремя ролями:

L1 "Студент" получает задания, тесты, учебные материалы.

L2 "Преподаватель" создает, редактирует и размещает задания, тесты, учебные материалы, оценивает работу студентов.

L3 "Посетитель" просматривает страницы, но доступа к материалам не имеет.

В реализации использовать паттерн Медиатор.

5. Реализовать программу обеспечивающую решение следующей проблемы. Допустим, имеется некий сервер локального времени (например, считает секунды от запуска). Необходима синхронизация с ним трех различных визуальных компонентов (например, 1 - обновление тестового сообщения "прошло _ с", 2 - запуск музыкального клипа через n локальных секунд, 3) - анимация процесса движения, повторяющаяся через 20 локальных секунд). Сервер и визуальные компоненты реализовать как отдельные программные модули.

6. Реализуйте MVP программной системы статистического анализа данных из различных источников с использованием архитектурного стиля «каналы и фильтры».

7. Выполните перенос настольного ПС «Органайзер» на платформу Telergram.

8. Разработайте программное средство для управления сенсорами и датчиками по протоколу MQTT.

9. Спроектируйте программное средство для решения проблемы фильтрации электронной почты по установленным приоритетам и правилам.

10. Реализуйте новый функционал микросервисной системы управления событиями.

7.2.4 Примерный перечень вопросов для подготовки к зачету

Цели и задачи технологий разработки ПО.

Особенности современных крупных проектов разработки ПО.

Понятие методологии разработки ПО. Работа в команде разработчиков

Структурный подход к проектированию ПО. Сущность структурного подхода.

Объектно-ориентированные методологии разработки программных систем.

Каскадная модель жизненного цикла ПС: содержание этапов, область применения, достоинства и недостатки.

Эволюционная модель жизненного цикла ПС: последовательность действий, область применения, достоинства и недостатки.

Спиральная модель разработки ПО: содержание этапов создания ПС, область применения, достоинства и недостатки.

Инкрементальная модель разработки ПО. Развитие инкрементального подхода.

Основные, вспомогательные и организационные процессы программной инженерии.

Жизненный цикл ПО. Различие понятий процесса и продукта.

Цель, этапы, роль измерений в разработке программ.

Легковесные и тяжеловесные методологии разработки ПО. Примеры

Сравнение легковесных и тяжеловесных методологии разработки ПО. Области использования

Оценка качества процесса разработки ПО.

7.2.5 Примерный перечень заданий для подготовки к экзамену

1. *Понятие программная инженерия. Основные, вспомогательные и организационные процессы программной инженерии.*

2. *Легковесные и тяжеловесные методологии разработки ПО. RUP, RAD, Scrum. Сравнение. Области использования.*

3. *Экстремальное программирование. XP-методологии: Scrum, Kanban, Lean, Feature Driven Development (FDD).*

4. *Каскадная модель жизненного цикла ПС: содержание этапов, область применения, достоинства и недостатки.*

5. *Эволюционная модель жизненного цикла ПС: последовательность действий, область применения, достоинства и недостатки.*

6. *Спиральная модель разработки ПО: содержание этапов создания ПС, область применения, достоинства и недостатки.*

7. *Понятие программного проекта. Управление программным проектом.*
8. *Определение требования к ПО. Уровни требований. Виды требований к ПО: системные, функциональные, характеристики качества*
9. *Роль требований в разработке ПО. Характеристики требований. Характеристики набора требований.*
10. *Источники сбора требований. Методы сбора и извлечения требований*
11. *Методы и средства формализации требований к ПО. Языки спецификаций. Техническое задание и SRS.*
12. *Объектный подход к проектированию ПО. Объектно-ориентированная декомпозиция. Принципы проектирования SOLID*
13. *Структурный подход к проектированию ПО. Сущность структурного подхода. Модульное программирование. Характеристики программного модуля.*
14. *Принципы проектирования пользовательского интерфейса.*
15. *Понятие архитектура ПО. Модели архитектур с различными способами обмена данными: репозиторий, «клиент-сервер». Архитектуры с различными моделями управления. Событийно-управляемые архитектуры. Многоуровневые архитектуры. Примеры. Использование слоев/уровней в различных архитектурах ПО. Модели архитектур с различными подходами к обработке данных: непрерывная обработка, каналы и фильтры, микроядерная архитектура.*
16. *Паттерны проектирования.*
17. *Принципы конструирования ПО. Характеристики видов работ (этапов)*
18. *Методы конструирования ПО: защитное программирование, обработка ошибок, стили кодирования, использование утверждений, безопасных языковых конструкций.*
19. *Инструментальные средства разработки ПО: отладчики, компиляторы, анализаторы, системы сборки и контроля версий, интегрированная среда разработки, средства совместной работы.*
20. *Тестирование интеграции: нисходящее и восходящее. Понятие драйвер и заглушка. Разработка тестовых наборов: Планирование, создание тестовой среды, генерация контрольных примеров на основе сценариев, формальных методов и диаграмм состояний, тестовое покрытие. Автоматизация тестирования.*
21. *Функциональное тестирование. Метод эквивалентного разбиения, граничных значений, таблицы решений.*
22. *Структурное тестирование. Покрытие операторов, ветвей, условий. Комбинаторное покрытие.*
23. *Управление документацией разработки программного продукта. Цели и задачи процесса документирования. Функции документации. Виды программной документации. С-документация и П-документация. Процесс разработки программной документации. Инструменты: вики и markdown.*

24. *Верификация и валидация ПО. Методы верификации ПО: статическое и динамическое тестирование. Инспекции, сквозной контроль, парные проверки. Измерение производительности в реальных условиях эксплуатации. Использование специализированных инструментов для тестирования асинхронного кода. Использование формальных методов для критических систем. Точность и стабильность численных алгоритмов.*

25. *Выпуск ПО. Системы сборки. Непрерывная интеграция, развертывание и доставка. Регрессионное тестирование*

26. *Понятие качества ПО и его многоуровневая модель. Характеристики, атрибуты и метрики качества программного средства. Оценка качества ПО. Внешнее и внутренне качество.*

27. *Сопровождение ПО. Виды сопровождения. Основные подходы к процессу сопровождения ПО: с целью исправления ошибок, адаптации и изменения функциональных возможностей. Рефакторинг, реинженерия, реверсная инженерия. Решение проблемы устаревшего ПО: миграция, поддержка устаревшего через адаптацию и инкапсуляцию.*

28. *Понятие конфигурации ПО. Примеры конфигурационных единиц. Понятие базовой версии. Семантическое версионирование. Работы по конфигурационному управлению. Системы контроля версий.*

7.2.6. Методика выставления оценки при проведении промежуточной аттестации

Экзамен проводится по билетам, содержащим 2 теоретических вопроса и практическое задание.

1. *Оценка «Неудовлетворительно» ставится в случае, если студент не изложил по существу ни одного теоретического вопроса и не выполнил практическое задание.*

2. *Оценка «Удовлетворительно» ставится в случае, если студент изложил теоретические вопросы по существу и продемонстрировал ход решения задачи*

3. *Оценка «Хорошо» ставится в случае, если студент правильно изложил теоретические вопросы и правильно обосновал ход решения практической задачи.*

4. *Оценка «Отлично» ставится, если студент полно изложил теоретические вопросы и полностью пояснил ход своего, полностью самостоятельно выполненного решения практической задачи.*

7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Методологии разработки ПО	ПК-2	Тест, защита лабораторных работ, выполнение практических заданий, требования к курсовому проекту

2	Инженерия требований	ПК-2	Тест, защита лабораторных работ, выполнение практических заданий, требования к курсовому проекту
3	Проектирование ПО	ПК-2	Тест, защита лабораторных работ, выполнение практических заданий, требования к курсовому проекту
4	Конструирование ПО	ПК-2	Тест, защита лабораторных работ, выполнение практических заданий, требования к курсовому проекту
5	Тестирование, верификация и оценка качества ПО	ПК-2	Тест, защита лабораторных работ, выполнение практических заданий, требования к курсовому проекту
6	Выпуск, развертывание, эксплуатация и сопровождение ПО	ПК-2	Тест, защита лабораторных работ, выполнение практических заданий, требования к курсовому проекту

7.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Тестирование осуществляется, либо при помощи компьютерной системы тестирования. Время тестирования 10 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методики выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется по индивидуальным заданиям. Время решения задач 45 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется самостоятельно по индивидуальному варианту. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Защита курсового проекта или отчета по всем видам практик осуществляется согласно требованиям, предъявляемым к работе, описанным в методических материалах. Примерное время защиты на одного студента составляет 20 мин.

8 УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ)

8.1 Перечень учебной литературы, необходимой для освоения дисциплины

1. Кулямин, В. В. Технологии программирования. Компонентный подход : учебное пособие / В. В. Кулямин. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 590 с. — ISBN 978-5-4497-0884-7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/102071.html> (дата обращения: 25.09.2023). — Режим доступа: для авторизир. Пользователей)

2. Минакова, О. В. Программная инженерия. Основные принципы, методы и инструменты : учебник для вузов / О. В. Минакова. — Санкт-Петербург : Лань, 2024. — 212 с. — ISBN 978-5-507-49278-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/414989> (дата обращения: 12.08.2024). — Режим доступа: для авториз. пользователей.

3. Минакова О.В. Технологии программирования: паттерны проектирования в реализации JavaFX приложений : практикум / Минакова О.В.. — Воронеж : Воронежский государственный технический университет, ЭБС АСВ, 2020. — 156 с. — ISBN 978-5-7731-0911-2. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/111505.html> (дата обращения: 12.08.2024). — Режим доступа: для авторизир. пользователей

8.2 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения, ресурсов информационно-телекоммуникационной сети «Интернет», современных профессиональных баз данных и информационных справочных систем:

Программное обеспечение:

Среда разработки ПО (IDE IntelliJ IDEA)

Информационно-справочные системы:

1. Сайт CITForum www.citforum.ru Библиотека технических материалов по информационным технологиям

2. Сайт компании Sun для поддержки языка программирования Java <http://docs.oracle.com/javase/8/javafx/get-started-tutorial/> - описание технологии JavaFx

3. Сайт поддержки разработчиков ПО <https://www.jetbrains.com/ru-ru/support/>, Справочная техническая документация по среде разработки IntelliJ IDEA и поддержки языка программирования Java

4. Комитет по стандартизации в области радиоэлектроники и вычислительной техники www.ieee.org Нормативно-справочная документация по информатике, программированию и вычислительной техники

5. SWEBOK (Software Engineering Body of Knowledge) <http://www.swebok.org/> международный стандарт ISO/IEC TR 19759, в котором описана общепринятая сумма знаний по программной инженерии <https://ligurio.github.io/swebok-ru/> — в авторской интерпретации (С. Орлик 2014 г.)

9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Компьютерный класс с предустановленной средой разработки приложений.

Мультимедийные средства: наборы файлов презентаций по темам лекционных занятий, комплект видеороликов по установке, настройке и примерам использования инструментальных средств технологии программирования.

Средства мониторинга – программа тестирования по модулям дисциплины с базами тестовых вопросов

10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

По дисциплине «Программная инженерия» читаются лекции, проводятся практические занятия и лабораторные работы, выполняется курсовой проект.

Основой изучения дисциплины являются лекции, на которых излагаются наиболее существенные и трудные вопросы, а также вопросы, не нашедшие отражения в учебной литературе.

Практические занятия направлены на приобретение практических навыков выполнения работ по созданию программного продукта с использованием инструментальных средств. Занятия проводятся путем решения конкретных задач в аудитории.

Лабораторные работы выполняются на специально ПО в соответствии с методиками, приведенными в указаниях к выполнению работ.

Методика выполнения курсового проекта изложена в учебно-методическом пособии. Выполнять этапы курсового проекта должны своевременно и в установленные сроки.

Контроль усвоения материала дисциплины производится защитой курсового проекта.

Вид учебных занятий	Деятельность студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на лекции или на практическом занятии.
Практическое занятие	Конспектирование рекомендуемых источников. Работа с конспектом лекций, подготовка ответов к контрольным вопросам, просмотр рекомендуемой литературы. Прослушивание аудио- и видеозаписей по заданной теме, выполнение расчетно-графических заданий, решение задач по алгоритму.
Лабораторная работа	Лабораторные работы позволяют научиться применять теоретические знания, полученные на лекции при решении конкретных задач. Чтобы

	наиболее рационально и полно использовать все возможности лабораторных для подготовки к ним необходимо: следует разобрать лекцию по соответствующей теме, ознакомиться с соответствующим разделом учебника, проработать дополнительную литературу и источники, решить задачи и выполнить другие письменные задания.
Самостоятельная работа	Самостоятельная работа студентов способствует глубокому усвоения учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: - работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций; - выполнение домашних заданий и расчетов; - работа над темами для самостоятельного изучения; - участие в работе студенческих научных конференций, олимпиад; - подготовка к промежуточной аттестации.
Подготовка к промежуточной аттестации	Готовиться к промежуточной аттестации следует систематически, в течение всего семестра. Интенсивная подготовка должна начаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед зачетом, экзаменом, экзаменом три дня эффективнее всего использовать для повторения и систематизации материала.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

№ п/п	Перечень вносимых изменений	Дата внесения изменений	Подпись заведующего кафедрой, ответственной за реализацию ОПОП
----------	-----------------------------	----------------------------	--