

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный технический университет»

УТВЕРЖДАЮ
Декан факультета информационных технологий
и компьютерной безопасности
/ П.Ю. Гусев/
2023 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
«Технология программирования»

Направление подготовки 09.03.02 ИНФОРМАЦИОННЫЕ СИСТЕМЫ И
ТЕХНОЛОГИИ

Профиль Отраслевые информационные системы
Квалификация выпускника бакалавр
Нормативный период обучения 4 года
Форма обучения очная
Год начала подготовки 2023

Автор программы

 /Минаикова О.В./

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

 /Десятирикова Е.Н./

Руководитель ОПОП

 /Курипта О.В./

Воронеж 2023

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины

Целью данной дисциплины является изучение технологических процессов разработки программного обеспечения, методов и средств их организации, и получения навыков коллективной разработки программного продукта и практики использования типовых алгоритмов и шаблонов проектирования

1.2. Задачи освоения дисциплины

- знать алгоритмы, структуры данных, шаблоны и типовые средства проектирования программ под конкретную практическую задачу;
- получить опыт работы с инструментальными средами разработки программного обеспечения;
- развить навыки программирования на языках высокого уровня и разработки пользовательских интерфейсов;
- закрепить навыки отладки и тестирования программ, документирования и оценки качества программных продуктов.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Технология программирования» относится к дисциплинам части, формируемой участниками образовательных отношений блока Б1.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Технология программирования» направлен на формирование следующих компетенций:

ПК-2 - Способен управлять доступом к данным с учетом архитектур информационных систем и актуальных стандартов деятельности организации

ПК-3 - Способен разрабатывать технологии интеграции и осуществлять прототипирование компонентов информационных систем

ПК-6 - Способен разрабатывать программные решения, включая верификацию требований, проектирование, отладку, проверку работоспособности и модификацию программных решений по предметной области

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ПК-2	знать технологии разработки программного обеспечения
	уметь разрабатывать программные компоненты для работы со сторонними системами
	владеть навыками адаптации программных систем к изменяющимся условиям функционирования

ПК-3	знать языки программирования, шаблоны проектирования и библиотеки программ
	уметь применять современные методы и средства создания программного кода
	владеть навыками планирования процесса разработки программных средств в соответствии с требованиями заказчика и стандартами предметной области
ПК-6	знать принципы построения и виды архитектуры программного обеспечения
	уметь выбирать средства реализации требований к программному обеспечению, кодировать и тестировать программное обеспечение
	владеть навыками разработки и выпуска программного продукта и документации к нему

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Технология программирования» 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
Лабораторные работы (ЛР)	72	36	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

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	Лаб. зан.	СРС	Всего, час
1	Введение в промышленную разработку ПО	<p>Понятие технологии программирования как разработки надежного ПО: Понятие программа и надежность ПО. Проблемы разработки программного обеспечения. Связь технологии программирования и инженерии программного обеспечения.</p> <p>Жизненный цикл ПО: Стандарты в области программной инженерии. Основные и вспомогательные процессы разработки ПО. Этапы разработки ПО. Каскадная и эволюционная модель построения ПО. Итерационная и инкрементная разработка. Спиральная модель и оценка рисков.</p>	4	8	6	18
2	Основные процессы разработки ПО	<p>Инженерия требований к ПО: понятие требования к ПО, спецификация качества, функциональная спецификация. Анализ предметной области и определение требований.</p> <p>Методы извлечения и формирования требований: анкетирование, опорные точки зрения, сценарии использования. Методы контроля внешнего описания.</p> <p>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. Проведение оценки качества.</p> <p>Архитектурное проектирование: Архитектура с общим репозиторием, клиент-серверная, многоуровневая. Модели централизованного и событийного управления. Архитектура канала и фильтра.</p> <p>Декомпозиция программной архитектуры: Структурный подход к разработке ПО. Абстракция и декомпозиция. Модульное программирование. Характеристики программного модуля. Связанность и сцепление.</p> <p>Объектно-ориентированное проектирование: Объектная декомпозиция системной архитектуры. Объекты и классы, зависимости.. Разработка и оценка архитектуры на основе сценариев. Использование UML. Статические и динамические диаграммы.</p>	12	12	6	30
3	Методы конструирования программ	<p>Основные конструкции языков программирования: История языков программирования. Переменные и типы данных: примитивные типы, массивы, структуры, строки. Базовый синтаксис языков программирования. Механизмы абстракции. Рекурсия, структурная декомпозиция, функции и передача параметров.</p>	2	18	4	24

		<p>Проектирование пользовательского интерфейса: Психологические и физиологические факторы. Принципы создания удобного ПО. Виды интерфейсов пользователя. GUI, элементы управления, компоновки. Стиль MVC. Механизмы обработки событий.</p> <p>Принципы SOLID. Паттерны проектирования: Структурные, порождающие и поведенческие шаблоны построения программ. Области и применения паттернов Компоновщик, Декоратор, Наблюдатель, Абстрактная фабрика, Стратегия</p>				
4	Тестирование ПО	<p>Тестирование – как инструмент разработки ПО: Категории программных ошибок. Типы тестов. Тестирование на этапе планирования. Тестирование на этапе проектирования. Тестирование "белого ящика" на стадии кодирования. Способы тестирования базого пути, тестирования условий, циклов. Тестирование "черного ящика". Методы функционального тестирования.</p> <p>Проектирование объектно-ориентированных тестовых сценариев: Способы тестирования содержания классов. Использование Unit-тестов. Заглушки. Понятие отладки ПО. Рефакторинг</p> <p>Модификация ПО: Эволюция программной архитектуры. Регрессионное тестирование. Реинженеринг. Особенности сопровождения программных продуктов. Проблемы наследуемого ПО.</p>	6	12	8	26
5	Вспомогательные процессы разработки ПО	<p>Управление конфигурацией: Возникновение дисциплины. Основные термины управления конфигурацией. Технологии управления SVN, git. Обзор программных средств совместной разработки и управления конфигурацией.</p> <p>Документирование ПО: Стандарты разработки программной документации. Составление инструкции пользователя, рекламно-технических описаний, руководства по сопровождению. Практика использования JavaDocs.</p> <p>Процессы обеспечения качества. Верификация и Аттестация ПО. Модели качества. Метрики и оценка качества ПО</p>	6	14	6	26
6	Управление разработкой программного средства	<p>Управление программными проектами: Понятие проекта. Характеристика основных этапов программного проекта. Построение сетевых графиков. Диаграмма Гранта. Обзор ПО поддержки управления программным проектом.</p> <p>Современные методологии разработки ПО: Тяжеловесные и легковесные процессы. RUP, MFC. XP-программирование. Agile-техники: Scrum, Kanban.</p> <p>Инструментальные средства технологий программирования: Инструменты разработки (IDE, SDK) Трансляторы,</p>	6	8	6	20

		компиляторы. Использование виртуальной машины и технологий контейнеризации. Использование Фреймворков и сторонних библиотек и CASE-средств.				
Итого			36	72	36	144

5.2 Перечень лабораторных работ

№ п/п	Тема и содержание	Трудоемкость	Вид контроля
1	Создание проекта в стиле MVC в среде разработки IntelliJ IDEA	2	наблюдение
2	Построение графического интерфейса пользователя редактором сцен JavaFX Scene Builder	2	наблюдение
3	Реализация ПС "Строительный калькулятор" с использованием Делегирования	2	отчет
4	Рисование фигур на основе технологии JavaFX	2	наблюдение
5	Разработка программы с использованием паттерна Абстрактный суперкласс	2	отчет
6	Разработка программы с использованием паттерна Фабричный метод	2	отчет
7	Создание анимации средствами JavaFX на основе объектов-фигур	2	наблюдение
8	Разработка программы с использованием паттерна Строитель	2	отчет
9	Разработка программы с использованием паттерна Итератор	2	отчет
10	Разработка ПС Клипмейкер	2	отчет
11	Использование коллекций (структур данных) для реализации программ	2	наблюдение
12	Разработка программы с использованием паттерна Стратегия	2	отчет
13	Разработка программы с использованием паттерна Адаптер	2	отчет
14	Разработка обучающей программы работы со структурами данных	4	отчет
15	Разработка программы с использованием паттерна Цепочка обязанностей	2	отчет
16	Создание визуальных эффектов для анимации графики	2	наблюдение
17	Разработка компьютерной slick-игры	4	отчет
18	Разработка программы с использованием паттерна Наблюдатель	4	отчет
19	Разработка программы с использованием механизма связывания и паттерна Абстрактная фабрика	2	отчет
20	Использование наблюдаемых коллекций (структур данных) для реализации программ	2	отчет
21	Разработка программного обеспечения для контроля знаний с использованием паттерна Посредник	4	отчет
22	Разработка программы с использованием паттерна Декоратор	2	отчет
23	Реализация консольного приложения с использованием паттерна Компоновщик	2	отчет
24	Разработка редактора диаграмм классов с использованием паттерна Компоновщик	4	отчет
25	Работа с git командами	2	наблюдение
26	Оформление программной документации на markdown	2	отчет
27	Создание maven-проекта	2	наблюдение
28	Анализ программного кода	2	наблюдение
29	Разработка Unit-тестов	2	наблюдение
30	Создание функциональных тестов	2	наблюдение
31	Рефакторинг программного кода	2	наблюдение

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.

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;
- Проектирование программной системы, создание пользовательского интерфейса, моделирование последователи взаимодействия пользователя с системой
- Реализация программы, оформление кода и документации к программе

Курсовой проект включает в себя создание программного продукта 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 технологии разработки программного обеспечения	Устный опрос, тестирование	Более 55% правильных ответов	Менее 55% правильных ответов
	уметь разрабатывать программные компоненты для работы со сторонними системами	Разработка системной спецификации к ПС с использованием стандартных нотаций	Соответствие спецификаций стандартным нотациям	Не соответствие спецификаций стандартным нотациям
	иметь опыт адаптации	Реализация ПС в соответствии со	Все заданные функции ПС	Тестирование ПС выявило

	программных систем к изменяющимся условиям функционирования	спецификацией	реализованы	отсутствие заданной функции
ПК-3	знать языки программирования, шаблоны проектирования и библиотеки программ	Устный опрос, тестирование	Более 55% правильных ответов	Менее 55% правильных ответов
	уметь применять современные методы и средства создания программного кода	Разработка системной спецификации к ПС с использованием стандартных нотаций	Соответствие спецификаций стандартным нотациям	Не соответствие спецификаций стандартным нотациям
	владеть навыками планирования процесса разработки программных средств в соответствии с требованиями заказчика и стандартами предметной области	Реализация ПС в соответствии со спецификацией	Все заданные функции ПС реализованы	Тестирование ПС выявило отсутствие заданной функции
ПК-6	знать принципы построения и виды архитектуры программного обеспечения	Устный опрос, тестирование	Более 55% правильных ответов	Менее 55% правильных ответов
	уметь выбирать средства реализации требований к программному обеспечению, кодировать и тестировать программное обеспечение	Разработка системной спецификации к ПС с использованием стандартных нотаций	Соответствие спецификаций стандартным нотациям	Не соответствие спецификаций стандартным нотациям
	иметь опыт разработки и выпуска программного продукта и документации к нему	Реализация ПС в соответствии со спецификацией	Все заданные функции ПС реализованы	Тестирование ПС выявило отсутствие заданной функции

7.1.2 Этап промежуточного контроля знаний

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

«зачтено»

«не зачтено»

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Зачтено	Не зачтено
ПК-2	знать технологии разработки программного обеспечения	Устный опрос, тестирование	Выполнение теста на 70-100%	Выполнение менее 70%
	уметь разрабатывать программные компоненты для работы со сторонними	Разработка системной спецификации к ПС с использованием стандартных нотаций	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

	системами иметь опыт адаптации программных систем к изменяющимся условиям функционирования	Реализация ПС в соответствии со спецификацией	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ПК-3	знать языки программирования, шаблоны проектирования и библиотеки программ	Устный опрос, тестирование	Выполнение теста на 70-100%	Выполнение менее 70%
	уметь применять современные методы и средства создания программного кода	Разработка системной спецификации к ПС с использованием стандартных нотаций	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть навыками планирования процесса разработки программных средств в соответствии с требованиями заказчика и стандартами предметной области	Реализация ПС в соответствии со спецификацией	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ПК-6	знать принципы построения и виды архитектуры программного обеспечения	Устный опрос, тестирование	Выполнение теста на 70-100%	Выполнение менее 70%
	уметь выбирать средства реализации требований к программному обеспечению, кодировать и тестировать программное обеспечение	Разработка системной спецификации к ПС с использованием стандартных нотаций	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	иметь опыт разработки и выпуска программного продукта и документации к нему	Реализация ПС в соответствии со спецификацией	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 технологии разработки 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 разрабатывать	Решение стандартных	Задачи решены в	Продемонстрирован верный	Продемонстрирован верный	Задачи не решены

	программные компоненты для работы со сторонними системами	практических задач	полном объеме и получены верные ответы	ход решения всех, но не получен верный ответ во всех задачах	ход решения в большинстве задач	
	иметь опыт адаптации программных систем к изменяющимся условиям функционирования	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
ПК-3	знать языки программирования, шаблоны проектирования и библиотеки программ	Тест	Выполнение теста на 90- 100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	уметь применять современные методы и средства создания программного кода	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 планирования процесса разработки программных средств в соответствии с требованиями заказчика и стандартами предметной области	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
ПК-6	знать принципы построения и виды архитектуры 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 выбирать средства реализации требований к программному обеспечению, кодировать и тестировать программное обеспечение	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
	иметь опыт разработки и выпуска программного продукта и документации к нему	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 (типичные 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

- A. совокупность процессов разработки надежных программных*

средств

В. систематизированное описание методов построения программных средств

С. создание формализованного описания процесса обработки данных

Д. методология разработки языков написания программ

9. Поставьте в соответствие этап жизненного цикла разработки ПС и его результат

a) Этап внешнего описания А) Фиксация требований к ПС

b) Этап конструирования ПС В) Разработка архитектуры, структуры программ и их детальная спецификация

c) Этап кодирования С) 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

a) конструирования и кодирования

b) внешнего описания и конструирования

c) кодирования и аттестации

d) внешнего описания и аттестации

Ответы: 1-8) а 9) a-A, b-B, c-C, d-D 10) a

7.2.2 Примерный перечень заданий для решения стандартных задач

1. Для программной системы «Метеостанция» составьте иерархию заинтересованных лиц и их ролей, используя метод опорных точек.

2. Для программной системы «Библиотека ВУЗа» смоделируйте для каждого выделенного действующего лица варианты использования.

3. Составьте перечень пользовательских и функциональных требований к программе «Графический редактор».

4. Составьте лист оценки характеристик качества программных средств построения UML-диаграмм и проведение экспертную оценку 3-4 программных продуктов.

5. Для любой ранее разработанной программы создайте механизм защиты полей ввода, включающий:

Контроль границ допустимых значений;

«Эхо»;

Обработка исключений.

7.2.3 Примерный перечень заданий для решения прикладных задач

1. Реализовать ПС "Раскраска" (тема рисунков - части компьютера) ПС должно осуществлять:

- 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 модуль авторизации для ИС поддержки образовательного процесса ВУЗа с тремя правами доступа:

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. На паттерне Наблюдатель реализовать программу обеспечивающую решение следующей задачи. 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.

7.2.4 Примерный перечень вопросов для подготовки к зачету

1. Цели и задачи технологии разработки ПО и программной инженерии. Особенности современных крупных проектов разработки ПО. Понятие программного средства, технологического процесса и продукта.

2. Процессы жизненного цикла ПО.

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

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

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

6. Инкрементальная модель разработки ПО. Развитие инкрементального подхода. Отличие от итерационного подхода к разработке ПО.

7. Методологии ПО: тяжеловесные и легковесные. Экстремальное программирование. RAD, Scrum, TDD, FDD.

8. Внешнее описание программного средства (понятие требований к ПО). Уровни требований бизнес-требования, пользовательские, функциональные, системные.

9. Функциональные и нефункциональные требования. Бизнес-правила, внешние интерфейсы, атрибуты качества и ограничения.

10. Процесс работы с требованиями. Этапы (извлечение, систематизация и анализ, формализация, документирование и аттестация) и виды выполняемых работ.

11. Источники требований. Методы сбора требований к ПО

12. Классификация требований, атрибуты требований. Трассировка требований.

13. Документирование требований (спецификация). Составление технического задания.

14. Проверка требований. Методы проверки требований.

7.2.5 Примерный перечень вопросов для подготовки к экзамену

1. Архитектурное, детальное и концептуальное проектирование. Понятие архитектуры ПО. Представление архитектуры и других артефактов проектирования.

2. Процесс проектирования ПО.

3. Принципы проектирования SOLID. Декомпозиция

4. Модульная декомпозиция. Характеристики программного модуля

5. Объектно-ориентированная декомпозиция. Общая характеристика объектов. Виды отношений между объектами. Делегирование. Агрегация.

6. Абстрагирование. Общая характеристика классов. Абстрактный суперкласс. Виды отношений между классами. Наследование. Полиморфизм.

7. Проектирование механизма обработки событий

8. Распределение функций по компонентам. CRC-карточки.

9. Обработка ошибок и исключительных ситуаций

10. Обеспечение отказоустойчивости ПО.

11. Моделирование взаимодействия с пользователем. Техника раскадровки (проектирование пользовательских представлений)

12. Многоуровневая архитектура ПО. Архитектура с общим репозиторием. Клиент-серверная архитектура.

13. Стиль MVC и его разновидности.

14. Порождающие шаблоны: Строитель, Абстрактная фабрика, Прототип, Одиночка.

15. Структурные паттерны проектирования: Адаптер, Компоновщик, Декоратор, Прокси.

16. Шаблоны поведения: Итератор, Посредник, Наблюдатель, Стратегия, Посетитель.

17. Общие принципы конструирования: минимизация сложности, ожидание изменений, конструирование с возможностью проверки.

18. Тестирование интеграции компонентов ПО: нисходящее и восходящее. Понятие драйвер и заглушка. Стохастическое тестирование.

19. Контрактное и защитное программирование. Современные

методы программирования: использование API, обобщения и параметризация, лямбда-выражения, callback.

20. Уровни и виды тестирования. Методы структурного тестирования. Оценка выполненных тестов. Анализ результатов Планирование тестирования. Составление тестовых наборов. Генерация тестовых сценариев.

21. Статическое и динамическое тестирование. Использование метода тестирования в различных видах деятельности по разработке ПО. Типы дефектов их классификация и статистика возникновения

22. Оценка надежности ПО. Модели роста надежности.

23. Определение конфигурационного управления и роль этого процесса в жизненном цикле ПО. Идентификация программных конфигураций. Учет статусов конфигураций. Аудит конфигураций. Управление выпуском и поставкой.

24. Конфигурация ПО. Инструменты сборки программного проекта.

25. Контроль версий. Инструменты управления версиями.

26. Отладка ПО: цели и методы. Использование утверждений. Логгирование.

27. Повторное использование кода. Библиотеки программ. Фреймворки.

28. Понятие качества ПО и его многоуровневая модель. Характеристики и атрибуты качества ПО. Измерения программных продуктов. Метрики.

29. Обеспечение качества ПО и процесса разработки. Методы обеспечения качества: инспекции, обзор кода, документирующие комментарии, статический анализ. Рефакторинг.

30. Виды программной документации. Средства разработки. Управление программной документацией.

31. Сопровождение ПО. Виды сопровождения. Общие принципы сопровождения. Тестирование. Анализ влияния Процесс сопровождения Анализ потребности и возможности сопровождения ПО Методы сопровождения: введение программной документации, реинжиниринг, реверсная инженерия. Решение проблемы миграции и удаления ПО.

32. Функциональное тестирование. Метод эквивалентного разбиения, граничных значений, причинно-следственных (функциональных) диаграмм и таблиц принятия решений.

7.2.6. Методика выставления оценки при проведении промежуточной аттестации

Экзамен проводится по билетам, каждый из которых содержит 2 вопроса и задачу. Каждый правильный ответ на вопрос оценивается от 1 до 5 баллов, задача оценивается в 10 баллов (5 баллов – обоснование и корректность использования выбранного метода решения и 5 баллов за полученный результат. Максимальное количество набранных баллов – 20.

1. Оценка «Неудовлетворительно» ставится в случае, если студент набрал менее 6 баллов.

2. Оценка «Удовлетворительно» ставится в случае, если студент набрал от 6 до 10 баллов

3. Оценка «Хорошо» ставится в случае, если студент набрал от 11 до 15 баллов.

4. Оценка «Отлично» ставится, если студент набрал от 16 до 20 баллов.

7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Введение в промышленную разработку ПО	ПК-2, ПК-3, ПК-6	Тест, практическое задание, индивидуальное задание на разработку программы, требования к курсовому проекту – разработка программной документация и готовый ПП
2	Основные процессы разработки ПО	ПК-2, ПК-3, ПК-6	Тест, практическое задание, индивидуальное задание на разработку программы, требования к курсовому проекту – разработка программной документация и готовый ПП
3	Методы конструирования программ	ПК-2, ПК-3, ПК-6	Тест, практическое задание, индивидуальное задание на разработку программы, требования к курсовому проекту – разработка программной документация и готовый ПП
4	Тестирование ПО	ПК-2, ПК-3, ПК-6	Тест, практическое задание, индивидуальное задание на разработку программы, требования к курсовому проекту – разработка программной документация и готовый ПП
5	Вспомогательные процессы разработки ПО	ПК-2, ПК-3, ПК-6	Тест, практическое задание, индивидуальное задание на разработку

			программы, требования к курсовому проекту – разработка программной документация и готовый ПП
6	Управление разработкой программного средства	ПК-2, ПК-3, ПК-6	Тест, практическое задание, индивидуальное задание на разработку программы, требования к курсовому проекту – разработка программной документация и готовый ПП

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

Тестирование осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием тест-заданий из 10 вопросов в презентации лекции. Время тестирования 15 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка «зачтено», если на 6 и более вопросов был дан правильный ответ.

Решение стандартных задач осуществляется по индивидуальным вариантам самостоятельно в течение 2 недель. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется в ходе лабораторных занятий и самостоятельно в течение 1 недели. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Защита курсового проекта осуществляется согласно требованиям, предъявляемым к разрабатываемой программе, 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. Терехов А.Н. *Технология программирования : учебное пособие / Терехов А.Н. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 148 с. — ISBN 978-5-4497-0702-4. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/97587.html>*

2. Минакова О.В. *Технологии программирования: Паттерны проектирования в реализации javafx-приложений/Минакова О.В. -Воронежский государственный технический университет, 2020 – 186с.*

3. Кулямин В.В. *Технологии программирования. Компонентный подход : учебное пособие* / Кулямин В.В.. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 590 с. — ISBN 978-5-4497-0884-7. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/102071.html>

4. Леоненков А.В. *Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose : учебное пособие* / Леоненков А.В.. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 317 с. — ISBN 978-5-4497-0667-6. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/97554.html>.

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

1. Среда разработки IntelliJ IDEA
2. git
3. MySQL Client

Информационно-справочные системы

1. Сайт <https://www.jetbrains.com/ru-ru/idea/resources/> документации продукта и другие источники полезной информации для эффективной разработки в IntelliJ IDEA
2. Сайт CITForum www.citforum.ru Библиотека технических материалов по информационным технологиям
3. Сайты поддержки разработчиков ПО на Java <https://docs.oracle.com/javase/tutorial/>
4. Комитет по стандартизации в области радиоэлектроники и вычислительной техники www.ieee.org Нормативно-справочная документация по вычислительной технике
5. Программная инженерия <http://www.software-engin.com/>
<http://www.cs.st-andrews.ac.uk> Авторские обзоры по современным тенденциям в инженерии ПО, обновление глав учебника «Программная инженерия»
Авторский электронный образовательный ресурс по дисциплине <https://github.com/olgmina/SWEngineering-technics.github.io>

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 «Технология программирования» проводятся лекции и лабораторные работы.

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

Лабораторные работы выполняются с использованием прикладного программного обеспечения в соответствии с методиками, приведенными в указаниях к выполнению работ.

Методика выполнения курсового проекта изложена в методических указаниях. Выполнять этапы курсовой работы должны своевременно и в установленные сроки.

Контроль усвоения материала дисциплины производится проверкой каждого этапа курсового проекта, 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.
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 усвоению учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: <ul style="list-style-type: none">- 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	Готовиться к промежуточной аттестации следует систематически, в

промежуточной аттестации	течение всего семестра. Интенсивная подготовка должна начаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед зачетом, экзаменом три дня эффективнее всего использовать для повторения и систематизации материала.
--------------------------	--