

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
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  
«Технология программирования»**

**Направление подготовки** 09.03.03 Прикладная информатика

**Профиль** Прикладная информатика в экономике цифрового общества

**Квалификация выпускника** бакалавр

**Нормативный период обучения** 4 года / 4 года и 11 м.

**Форма обучения** очная / заочная

**Год начала подготовки** 2021

Автор программы  
Заведующий кафедрой  
Систем управления и  
информационных  
технологий в  
строительстве

Т.В.Корелина

Е.Н.Десятирикова

Руководитель ОПОП

Н.Г. Аснина

Воронеж 2021

## 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-1 - Способен проводить обследование организаций, выявлять информационные потребности пользователей, разрабатывать требования и осуществлять проектирование программного обеспечения.

ПК-5 - Способен собирать информацию для инициации проекта, организовывать заключение договоров и дополнительных соглашений в соответствии с полученным заданием.

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ПК-1	знать принципы разработки технического проекта
	уметь составлять требования к техническому решению проекта
	владеть навыками сборки и интеграции проекта
ПК-5	знать методики описания и моделирования бизнес-процессов
	уметь проводить анкетирование и интервьюирование для требований заказчика
	владеть навыками сбора и управления требованиями к программному обеспечению

#### 4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Технология программирования» составляет 4 з.е.

Распределение трудоемкости дисциплины по видам занятий  
**очная форма обучения**

Виды учебной работы	Всего часов	Семестры
		6
<b>Аудиторные занятия (всего)</b>	90	90
В том числе:		
Лекции	36	36
Лабораторные работы (ЛР)	54	54
<b>Самостоятельная работа</b>	54	54
<b>Курсовой проект</b>	+	+
Виды промежуточной аттестации - зачет с оценкой	+	+
Общая трудоемкость: академические часы	144	144
зач.ед.	4	4

**заочная форма обучения**

Виды учебной работы	Всего часов	Семестры
		8
<b>Аудиторные занятия (всего)</b>	12	12
В том числе:		
Лекции	4	4
Лабораторные работы (ЛР)	8	8
<b>Самостоятельная работа</b>	128	128
<b>Курсовой проект</b>	+	+
Часы на контроль	4	4
Виды промежуточной аттестации - зачет с оценкой	+	+
Общая трудоемкость: академические часы	144	144
зач.ед.	4	4

#### 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	Понятие технологии программирования как разработки надежного ПО: Понятие программа и надежность ПО. Проблемы разработки программного обеспечения. Связь технологии программирования и инженерии программного	6	10	8	24

		<p>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>				
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. 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>	6	10	8	24
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> <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>	6	10	8	24
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 базового пути, тестирования условий, циклов. Тестирование "черного ящика". Методы функционального тестирования. Проектирование объектно-ориентированных тестовых</p>	6	8	10	24

		сценариев: Способы тестирования содержания классов. Использование Unit-тестов. Заглушки. Понятие отладки ПО. Рефакторинг Модификация ПО: Эволюция программной архитектуры. Регрессионное тестирование. Реинженеринг. Особенности сопровождения программных продуктов. Проблемы наследуемого ПО.				
5	Вспомогательные процессы разработки ПО	Управление конфигурацией: Возникновение дисциплины. Основные термины управления конфигурацией. Технологии управления SVN, git. Обзор программных средств совместной разработки и управления конфигурацией. Документирование ПО: Стандарты разработки программной документации. Составление инструкции пользователя, рекламно-технических описаний, руководства по сопровождению. Практика использования JavaDocs. Процессы обеспечения качества. Верификация и Аттестация ПО. Модели качества. Метрики и оценка качества ПО	6	8	10	24
6	Управление разработкой программного средства	Управление программными проектами: Понятие проекта. Характеристика основных этапов программного проекта. Построение сетевых графиков. Диаграмма Гранта. Обзор ПО поддержки управления программным проектом. Современные методологии разработки ПО: Тяжеловесные и легковесные процессы. RUP, MFC. XP-программирование. Аngle-техники: Scrum, Kanban. Инструментальные средства технологий программирования: Инструменты разработки (IDE, SDK) Трансляторы, компиляторы. Использование виртуальной машины и технологий контейнеризации. Использование Фреймворков и сторонних библиотек и CASE-средств.	6	8	10	24
<b>Итого</b>			<b>36</b>	<b>54</b>	<b>54</b>	<b>144</b>

### заочная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего, час
1	Введение в промышленную разработку ПО	Понятие технологии программирования как разработки надежного ПО: Понятие программа и надежность ПО. Проблемы разработки программного обеспечения. Связь технологии программирования и инженерии программного обеспечения. 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.	2	2	20	24
2	Основные процессы разработки ПО	Инженерия требований к ПО понятие требования к ПО, спецификация качества, функциональная спецификация. Анализ предметной области и определение требований. Методы извлечения и формирования требований: анкетирование, опорные точки зрения, сценарии использования. Методы контроля внешнего описания. Характеристики качества ПО: Модель качества по ISO 9126. Характеристики и субхарактеристики качества программного	2	2	20	24

		<p>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. 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>				
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> <p>Проектирование пользовательского интерфейса: Психологические и физиологические факторы. Принципы создания удобного ПО. Виды интерфейсов пользователя. GUI, элементы управления, компоновки. Стил 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>	-	2	22	24
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 базового пути, тестирования условий, циклов. Тестирование "черного ящика". Методы функционального тестирования. 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>	-	2	22	24
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, рекламных-технических описаний, руководства по сопровождению. Практика использования JavaDocs.</p>	-	-	22	22

		Процессы обеспечения качества. Верификация и Аттестация ПО. Модели качества. Метрики и оценка качества ПО				
6	Управление разработкой программного средства	Управление программными проектами: Понятие проекта. Характеристика основных этапов программного проекта. Построение сетевых графиков. Диаграмма Гранта. Обзор ПО поддержки управления программным проектом. Современные методологии разработки ПО: Тяжеловесные и легковесные процессы. RUP, MFC. XP-программирование. Angle-техники: Scrum, Kanban. Инструментальные средства технологий программирования: Инструменты разработки (IDE, SDK) Трансляторы, компиляторы. Использование виртуальной машины и технологий контейнеризации. Использование Фреймворков и сторонних библиотек и CASE-средств.	-	-	22	22
<b>Итого</b>			<b>4</b>	<b>8</b>	<b>128</b>	<b>140</b>

## 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	отчет
3	Реализация ПС "Строительный калькулятор" с использованием Делегирования	2	отчет
4	Рисование фигур на основе технологии JavaFX	2	отчет
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	отчет
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	отчет
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	2	отчет
15	Разработка программы с использованием паттерна Цепочка обязанностей	2	отчет
16	Создание визуальных эффектов для анимации графики	2	отчет
17	Разработка компьютерной slick-игры	2	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
----	--	---	-------

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

Курсовой проект включают 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-1	знать принципы разработки технического проекта	Устный опрос, тестирование	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь составлять требования к техническому решению проекта	Разработка системной спецификации к ПС с использованием стандартных нотаций	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть навыками сборки и интеграции проекта	Реализация ПС в соответствии со спецификацией	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ПК-5	знать методики описания и моделирования бизнес-процессов	Устный опрос, тестирование	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь проводить анкетирование и интервьюирование для требований заказчика	Разработка системной спецификации к ПС с использованием стандартных нотаций	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть навыками сбора и управления требованиями к программному обеспечению	Реализация ПС в соответствии со спецификацией	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

### 7.1.2 Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются в 6 семестре для очной формы обучения, 8 семестре для заочной формы обучения по четырехбалльной системе:

«отлично»;

«хорошо»;

«удовлетворительно»;

«неудовлетворительно».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Отлично	Хорошо	Удовл.	Неудовл.
ПК-1	знать принципы разработки технического проекта	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	уметь составлять требования к техническому решению проекта	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 сборки и интеграции проекта	Решение прикладных задач в конкретной	Задачи решены в полном объеме и	Продемонстрирован верный ход решения всех,	Продемонстрирован верный ход решения в	Задачи не решены

		предметной области	получены верные ответы	но не получен верный ответ во всех задачах	большинстве задач	
ПК-5	знать методики описания и моделирования бизнес-процессов	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	уметь проводить анкетирование и интервьюирование для требований заказчика	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 сбора и управления требованиями к программному обеспечению	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

### 7.2.1 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 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
  - B. систематизированное описание методов построения программных средств
  - C. создание формализованного описания процесса обработки данных
  - D. 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
  - 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) 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 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

задани

я, 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**

Не предусмотрено учебным планом

#### **7.2.5 Примерный перечень заданий для подготовки к экзамену**

1. Цели и задачи технологий разработки ПО. Особенности современных крупных проектов разработки ПО.

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

3. Структурный подход к проектированию ПО. Сущность структурного подхода.

4. Объектно-ориентированная разработка программ. Объектно-ориентированные языки программирования. Объектно-ориентированные методологии разработки программных систем.

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

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

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

8. Инкрементальная модель разработки ПО. Развитие инкрементального подхода.

9. Техники программирования: организация нескольких потоков управления в программе, блокировки, использование монитора.

10. Техники программирования: механизм обработки исключений.

11. Обзор кода (ревьюирование)/статическое тестирование и связанные с ним техники.

12. Управление документацией разработки программного продукта.

13. Легковесные и тяжеловесные методологии разработки ПО.

Примеры. Сравнение. Области использования.

14. Оценка качества процесса разработки ПО.

15. Внешнее описание программного средства и спецификация. Уровни требований- бизнес-требования, функциональные, интерфейсы, ограничения.
16. Методы сбора и формализация требований к ПО.
17. Понятие качества ПО и его многоуровневая модель. Характеристики и атрибуты качества.
18. Разработка требований к ПО: формирование и анализ, документирование, аттестация. Управление.
19. Принципы SOLID. Модульное программирование. Характеристики программного модуля.
20. Модели архитектур с различными способами обмена данными: репозиторий, «клиент-сервер».
21. Архитектуры с различными моделями управления. Событийно-управляемые архитектуры.
22. Многоуровневые архитектуры. Примеры. Использование слоев/уровней в различных архитектурах ПО.
23. Модели архитектур с различными подходами к обработке данных: непрерывная обработка, каналы и фильтры, микроядерная архитектура.
24. Объектно-ориентированная декомпозиция. Общая характеристика объектов. Виды отношений между объектами. Агрегация.
25. Абстрагирование. Общая характеристика классов. Виды отношений между классами. Наследование. Полиморфизм.
26. Инструментальные средства разработки ПО. Автоматизация разработки ПО. CASE-средства.
27. Принципы проектирования пользовательского интерфейса. Визуальные элементы управления и способы работы с ними.
28. Структурное тестирование. Покрытие операторов, ветвей, условий.
29. Функциональное тестирование. Метод эквивалентного разбиения, граничных значений, причинно-следственных (функциональных) диаграмм.
30. Тестирование интеграции компонентов ПО: нисходящее и восходящее. Понятие драйвер и заглушка. Стохастическое тестирование.
31. Разработка программной документации. С-документация и П-документация.
32. Отладка ПО: цели и методы. Техники – логирование, использование отладчика.
33. Понятие конфигурации ПО. Управление конфигурацией ПО.
34. Выпуск и внедрение ПО. Управление изменениями. Системы контроля версий.
35. Оценка качества ПО. Внешнее и внутренне качество. Цель, этапы, роль измерений в технологии программирования.

36. Сопровождение ПО. Основные подходы: с целью исправления ошибок, адаптации и изменения функциональных возможностей. Решение проблемы эволюции ПО – рефакторинг, реинженерия, реверсная инженерия.

37. Практические задания:

выделить опорные точки зрения для определения требований к ПО; предложить анкету для сбора требований к ПО;

построить диаграмму вариантов использования для детализации требований к ПО;

предложить графический интерфейс пользователя для заданной функциональной задачи и выполнить его «раскадровку» для заказчика;

обосновать выбор архитектуры ПС для поставленного ТЗ;

представить архитектуру ПС на основе компонентного подхода, модульного (блок-схемы), в виде CRC-карточки, с использованием UML диаграмм;

составить структурную модель для заданной системы;

разработать класс для реализации программного компонента и Unit тест для него;

предложить схему взаимодействия программных компонентов;

разработать диаграмму классов (объектов) приложения;

составить тестовый набор оценки функциональности (надежности, удобства использования) заданной

программы;

составить сценарий тестирования

заданной ПС; написать руководство

пользователя.

#### **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-1, ПК-5	Тест, практическое задание, индивидуальное задание на разработку программы, требования к курсовому проекту – разработка программной документация и готовый ПП
2	Основные процессы разработки ПО	ПК-1, ПК-5	Тест, практическое задание, индивидуальное задание на разработку программы, требования к курсовому проекту – разработка программной документация и готовый ПП
3	Методы конструирования программ	ПК-1, ПК-5	Тест, практическое задание, индивидуальное задание на разработку программы, требования к курсовому проекту – разработка программной документация и готовый ПП
4	Тестирование ПО	ПК-1, ПК-5	Тест, практическое задание, индивидуальное задание на разработку программы, требования к курсовому проекту – разработка программной документация и готовый ПП
5	Вспомогательные процессы разработки ПО	ПК-1, ПК-5	Тест, практическое задание, индивидуальное задание на разработку программы, требования к курсовому проекту – разработка программной документация и готовый ПП

6	Управление разработкой программного средства	ПК-1, ПК-5	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 выданных тест-заданий на бумажном носителе. Время тестирования 30 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методики выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Защита курсовой работы, 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. Терехов, А. Н. Технология программирования [Электронный ресурс] : учебное пособие / А. Н. Терехов. — Электрон. текстовые данные. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование, 2017. — 152 с. — 978-5-4487-0070-5. — Режим доступа: <http://www.iprbookshop.ru/67370.html>

2. Орлов С. А. Теория и практика языков программирования [Текст] : учебник. – Москва ; Санкт-Петербург ; Нижний Новгород [и др.] : Питер, 2014 (Чехов : Первая Образцовая тип., фил. "Чеховский Печатный Двор", 2014). - 688 с.

3. Кулямин, В. В. Технологии программирования.

Компонентный подход [Электронный ресурс] / В. В. Кулямин. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2019. — 590 с. — 5-9556-0067-1. — Режим

доступа:

<http://www.iprbookshop.ru/73733.html>

4. Практикум «Паттерны проектирования»/О.В.Минакова – Электронный ресурс <https://sites.google.com/view/study-pattern>

5. Леоненков, А. В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose. Курс лекций [Электронный ресурс] : учебное пособие для студентов вузов, обучающихся по специальностям в области информационных технологий / А. В. Леоненков. — Электрон. текстовые данные. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование, 2017. — 318 с. — 978-5-4487-0081-1. — Режим доступа: <http://www.iprbookshop.ru/67388.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](http://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](http://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»

## **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» читаются лекции, проводятся 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.

Лабораторные работы выполняются на лабораторном оборудовании 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 проверкой курсового проекта, защитой курсового проекта.

Вид учебных занятий	Деятельность студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; пометить 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"><li>- работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций;</li><li>- выполнение домашних заданий и расчетов;</li><li>- работа над темами для самостоятельного изучения;</li></ul>

	<ul style="list-style-type: none"><li>- участие в работе студенческих научных конференций, олимпиад;</li><li>- подготовка к промежуточной аттестации.</li></ul>
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 с оценкой, зачетом с оценкой 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
----------	-----------------------------	----------------------------	--