

**Федеральное государственное автономное образовательное учреждение
высшего образования
«Санкт-Петербургский политехнический университет Петра Великого»**

УТВЕРЖДАЮ
Директор ИКНК
_____ Д.П. Зегжда
«17» июня 2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

«Технологии разработки качественного программного обеспечения»

Разработчик	Высшая школа программной инженерии
Направление (специальность) подготовки	09.03.04 Программная инженерия
Наименование ООП	09.03.04_01 Технология разработки и сопровождения качественного программного продукта
Квалификация (степень) выпускника	бакалавр
Образовательный стандарт	СУОС
Форма обучения	Очная

СОГЛАСОВАНО	Соответствует СУОС
Руководитель ОП	Утверждена протоколом заседания
_____ А.В. Петров	высшей школы "ВШПИ" от «21» мая 2024 г. № №1

РПД разработал:
Старший преподаватель Л.П. Котлярова

1. Цели и планируемые результаты изучения дисциплины

Цели освоения дисциплины

Целью изучения дисциплины является формирование специалистов, умеющих обоснованно и результативно применять знание принципов и методов разработки программного обеспечения сложных систем, подходов к разработке программного продукта коммерческого качества, методов коллективной работы над программным проектом, понимания современных отечественных и мировых требований и стандартов на программный продукт. Цели и задачи дисциплины предусматривают систему теоретической и практической подготовки разработчиков промышленного программного продукта.

Результаты обучения выпускника

Код	Результат обучения (компетенция) выпускника ООП
ПК-1	Способен проводить сбор и анализ требований к программному обеспечению и разрабатывать технические спецификации на его компоненты
ИД-1 ПК-1	Определяет виды требований к ПО и выбирает методы сбора требований из разных источников информации
ИД-2 ПК-1	Анализирует возможности реализации требований к программному обеспечению, их трудоемкость и определяет ограничения системы
ИД-3 ПК-1	Разрабатывает технические спецификации на программные компоненты и их взаимодействие на основе исходных данных бизнес-процессов заказчика
ПК-2	Способен выбирать модель жизненного цикла программного обеспечения, выполнять проектирование и ревизию его архитектуры
ИД-1 ПК-2	Выбирает модель жизненного цикла программного обеспечения в соответствии со спецификой предметной области
ИД-2 ПК-2	Применяет общие принципы проектирования программного обеспечения с учетом контекста выбранного жизненного цикла разработки и сопровождения
ПК-4	Способен осуществлять все виды тестирования программного продукта
ИД-1 ПК-4	Осуществляет построение тестовых случаев и выполняет тестирование на уровне модуля
ИД-2 ПК-4	Осуществляет построение тестовых случаев и выполняет тестирование на интеграционном и системном уровнях
ИД-3 ПК-4	Автоматизирует системные и интеграционные тесты

ИД-4 ПК-4	Осуществляет тестирование производительности программного продукта
--------------	--

Планируемые результаты изучения дисциплины

знания:

- Знает основные виды требований к программному обеспечению
- Знает способы оценки реализуемости требований с учетом заданных ограничений
- Знает основные принципы разработки спецификаций и формирования ограничений для компонентов программного обеспечения
- Знает основные модели жизненного цикла разработки программного обеспечения
- Знает основные этапы проектирования программного обеспечения в контексте выбранной модели жизненного цикла разработки ПО
- Знает ряд техник (методов) тестирования кода на уровне модулей
- Знает основные принципы разработки интеграционных и системных тестов
- Знает основные методы разработки различных видов тестов в соответствии с планом тестирования
- Знает основные виды тестирования производительности программного обеспечения

умения:

- Умеет проводить анализ и детализацию требований
- Умеет оценить реализуемость требований с учетом заданных ограничений в конкретной среде программирования
- Умеет разрабатывать спецификации для отдельных компонент программного обеспечения в рамках поставленных задач
- Умеет учитывать специфику предметной области при выборе модели жизненного цикла разработки программного обеспечения
- Умеет выполнять декомпозицию разрабатываемого программного обеспечения на компоненты с учетом выбранной модели жизненного цикла разработки ПО
- Умеет применять изученные техники тестирования кода модулей для конкретного языка и среды программирования
- Умеет разрабатывать интеграционные и системные тесты и выполнять интеграционное тестирование
- Умеет проводить автоматизацию выполнения различных видов тестов в соответствии с планом тестирования
- Умеет выполнять отдельные виды тестирования производительности программного обеспечения

навыки:

- Владеет основными методами сбора и разработки требований
- Владеет навыками определения реализуемости требований к программному обеспечению с учетом конкретных библиотек и среды программирования
- Владеет методами обзора спецификаций для проверки их корректности и полноты
- Владеет навыками описания основных этапов модели жизненного цикла разработки программного обеспечения и адаптации модели к конкретному проекту
- Владеет навыками работы с инструментами для описания проектных решений
- Владеет навыками работы с инструментами, предназначенными для модульного тестирования
- Владеет навыками работы с инструментарием, предназначенным для интеграционного и системного тестирования
- Владеет фреймворками автоматизации тестирования
- Владеет навыками работы с отдельными инструментами для тестирования производительности ПО конкретного программного продукта

2. Место дисциплины в структуре ООП

В учебном плане дисциплина «Технологии разработки качественного программного обеспечения» относится к модулю «Модуль цифровых компетенций».

Изучение дисциплины базируется на результатах освоения следующих дисциплин:

- Архитектура программных систем
- Методы оптимизации
- Теория автоматов и формальных языков
- Системы управления базами данных

3. Распределение трудоёмкости освоения дисциплины по видам учебной работы и формы текущего контроля и промежуточной аттестации

3.1. Виды учебной работы

Виды учебной работы	Трудоемкость по семестрам
	Очная форма
Лекционные занятия	50
Практические занятия	16
Самостоятельная работа	111
Часы на контроль	16
Промежуточная аттестация (экзамен)	11
Промежуточная аттестация (зачет)	4
Курсовое проектирование	8
Общая трудоемкость освоения дисциплины	216, ач
	6, зет

3.2. Формы текущего контроля и промежуточной аттестации

Формы текущего контроля и промежуточной аттестации	Количество по семестрам
	Очная форма
Текущий контроль	
Контрольные, шт.	4
Курсовые проекты, шт.	1
Промежуточная аттестация	
Зачеты, шт.	1
Экзамены, шт.	1

4. Содержание и результаты обучения

4.1 Разделы дисциплины и виды учебной работы

№ раздела	Разделы дисциплины, мероприятия текущего контроля	Очная форма		
		Лек, ач	Пр, ач	СР, ач
1.	Программные продукты (изделие). Основные понятия, связанные с технологиями программирования	2	1	3
2.	Жизненный цикл программного обеспечения (ПО), модели жизненного цикла ПО	2	1	4
3.	Процесс производства ПО: этапы, методы, технология и инструментальные средства			
3.1.	Обзор основных этапов производства и сопровождения ПО	2	1	5
3.2.	Требования к ПО	5	2	12
3.3.	Проектирование ПО (высокоуровневое и детальное)	7	2	14
3.4.	Разработка ПО по спецификациям (конструирование, построение ПО)	4	2	18
3.5.	Модульность ПО, эволюция модульного подхода	3	0	2
3.6.	Тестирование и отладка ПО	5	2	18
4.	Виртуализация и контейнерная виртуализация	4	1	6
5.	Автоматизация тестирования, отдельные виды тестирования на уровне системы	10	1	10
6.	Измерения ПО и критерии качества ПО			
6.1.	Основные понятия качества и метрик ПО	2	1	4
6.2.	Измерения в программном проекте	2	1	5
7.	Технологический цикл разработки программных систем, коллективная работа по созданию программ, организация процесса разработки и инструментальные средства поддержки	1	1	6
8.	Перспективные направления развития технологии программирования	1	0	4
Итого по видам учебной работы:		50	16	111
Зачеты, ач				6

Экзамены, ач		10
Часы на контроль, ач		16
Курсовое проектирование		8
Промежуточная аттестация (экзамен)		11
Промежуточная аттестация (зачет)		4
Общая трудоёмкость освоения: ач / зет		216 / 6

4.2. Содержание разделов и результаты изучения дисциплины

Раздел дисциплины	Содержание
1. Программные продукты (изделие). Основные понятия, связанные с технологиями программирования	1.1. Введение Информатика и технология программирования (ТП). Терминология В ПИ (глоссарии стандартизованных понятий). Классификация программного обеспечения (ПО). Персоналии в РФ и мире (ретроспектива). 1.2. Основные проблемы разработки программного обеспечения Основные проблемы разработки программного обеспечения (ПО). Сложность как основная проблема программирования. Источники сложности. Способы борьбы со сложностью ПП.
2. Жизненный цикл программного обеспечения (ПО), модели жизненного цикла ПО	2.1. Жизненный цикл ПО Понятие жизненного цикла ПП. Современные модели ЖЦ: водопадная (каскадная), поэтапная, V-типа, спиральная, инкрементальная, прототипная, гибкие модели разработки ПО. Поколения промышленных ТП. Сборочная технология: проблемы повторного использования модулей (reuse), возвратной инженерии (reengineering) и переноса на другие платформы (портирование). Стандарты IEEE и ISO/IEC. Международные стандарты терминов ПИ.
3. Процесс производства ПО: этапы, методы, технология и инструментальные средства	

3.1. Обзор основных этапов производства и сопровождения ПО	<p>3.1а. Планирование программного проекта Создание проектного плана. Методы оценки ресурсов и распределения работ. Риск-анализ. Отслеживание и контроль плана. Гант-диаграммы. Использование инструментальных средств. База развития проекта и ее использование в индустриальных ТП.</p> <p>3.1б. Обзор этапов сбора требований, проектирования, кодирования (конструирования) ПО, тестирования. Эти темы рассматриваются в отдельных блоках детально.</p> <p>3.1в. Документирование ПО Требования к ПО как промышленному продукту. Стандарты на оформление программного продукта (IEEE и ЕСПД) и виды программной документации. Средства автоматизации разработки программной документации в индустриальной ТП.</p> <p>3.1г. Сопровождение ПО Сопровождение ПО. Сопровождение или продолжающаяся разработка ПО: проблемы и перспективы. Инструментальные средства, поддерживающие этап сопровождения. Стиль программирования, ориентированный на эффективную поддержку этапа сопровождения.</p>
3.2. Требования к ПО	<p>3.2а. Общие понятия о требованиях. Виды требований: требования к продукту и процессам; функциональные / нефункциональные требования, производные требования. Выявление требований, источники требований. Согласование требований с заказчиком (баланс интересов).</p> <p>3.2б. Анализ требований. Классификация требований. Область действия требований. Изменчивость требований. Отслеживание и контроль требований, атрибуты требований.</p> <p>3.2в. Разработка спецификаций, модели требований. Популярные языки спецификаций UML, SDL, MSC. Требования и критерии проверяемости. Описание требований (неформализованный текст, табличное представление, машины состояний, языки спецификаций). Правила формулировки непротиворечивых требований. Использование инструментальных средств.</p>

3.3. Проектирование ПО (высокоуровневое и детальное)	<p>3.3а. Основы и принципы проектирования ПО. Ключевые проблемы в проектировании ПО. Структура и архитектура ПО. Архитектурные стили, шаблоны проектирования. Общие принципы проектирования пользовательского интерфейса.</p> <p>3.3б. Концептуальное проектирование (High Level Design) и детальное проектирование (Detail Design). Требования и критерии. Отслеживание и контроль спецификаций. Сравнительный анализ подходов к разработке спецификаций: (плакатная технология; методика спецификаций Парнаса; структурный анализ Росса; подход Кнута и проблемы спецификации мелкомодульных ТП).</p> <p>3.3в. Описания (нотации) проектирования ПО. Популярные языки спецификаций UML, SDL, MSC. Структурные описания и описания поведения. Стратегии и методы проектирования ПО. Методы анализа и оценки качества проектирования ПО. Атрибуты качества проектирования.</p> <p>Методы верификации требований и спецификаций: Model checking, дедуктивный анализ. Инструментальные средства поддержки проектирования спецификаций.</p> <p>Тест по темам о требованиях и проектировании ПО.</p>
3.4. Разработка ПО по спецификациям (конструирование, построение ПО)	<p>3.4а. Задачи, решаемые на этапе построения ПО: уменьшение сложности, повторное использование (reuse), применение стандартов. Планирование и управление на этапе построения ПО. Языки разработки ПО. Кодирование. Тестирование (модульное и интеграционное). Конструирование ПО для повторного использования и с повторным использованием. Вопросы, связанные с разными подходами к интеграции компонентов ПО.</p> <p>3.4б. Дизайн и использование API. Проблемы ОО среды выполнения. Утверждения (assertions); обработка ошибок, исключений. Исполняемые модели. Конфигурирование среды выполнения. Интернационализация.</p> <p>3.4в. Параллелизм. Анализ производительности. Инструментальные средства поддержки этапа разработки ПО: IDE, инструменты для модульного тестирования, для профилирования, анализа производительности.</p>

3.5. Модульность ПО, эволюция модульного подхода	<p>Модульность в ПО. Модульность ПО как средство борьбы со сложностью. Эволюция модульного подхода. Интерфейс. Проблемы экспорта-импорта данных. Контекст. Пакетирование модулей. Задачи. Требования к оформлению модулей в промышленной ТП: средства настройки, раздельная компиляция, статическое и динамическое связывание модулей.</p> <p>Тест по темам о построении ПО и модульности ПО как средство борьбы со сложностью ПО.</p>
3.6. Тестирование и отладка ПО	<p>Основные определения тестирования. Фазы тестирования. Методы тестирования и отладки ПО. Доказательство свойств программы и их экспериментальная проверка. Тестирование и отладка. Неразрешимость проблемы тестирования. Критерии тестирования модулей ПО.</p> <p>Граф программы. Основные проблемы тестирования. Критерии отбора тестов, классы критериев (структурные, функциональные, стохастическое тестирование, мутационные).</p> <p>Метод "белого ящика" и инструментальные средства, поддерживающие тестирование и отладку многомодульных программных комплексов. Макетирование ПО. Моделирование окружения. Интеграционное и системное тестирование. "Метод черного ящика". Регрессионное тестирование. Примеры инструментальных средств поддержки этапа тестирования ПО.</p>
4. Виртуализация и контейнерная виртуализация	<p>Виртуализация. Какие проблемы решает контейнерная виртуализация. Развитие контейнеров (инструмент Docker). Проблемы, созданные контейнерами. Инструмент Kubernetes: развитие, основные компоненты, хранение данных в Kubernetes.</p>
5. Автоматизация тестирования, отдельные виды тестирования на уровне системы	<p>5.1. Предпосылки внедрения автоматизации тестирования. Автоматизируемые виды тестирования. Agile и автоматизация тестирования. Виды документации в тестировании.</p> <p>5.2. Веб-тестирование, жизненный цикл разработки веб-приложения. Инструменты автоматизации тестирования пользовательского интерфейса (UI) веб-приложений.</p> <p>5.3. Нагрузочное тестирование, тестирование производительности высоконагруженных систем. Инструменты диагностики проблем производительности ПП. Виды тестирования производительности ПП и их цели.</p> <p>5.4. Docker и автоматизация тестирования. онлайн тест по пройденным темам.</p>
6. Измерения ПО и критерии качества ПО	

6.1. Основные понятия качества и метрик ПО	<p>Концепция обеспечения коммерческого качества для промышленной ТП.</p> <p>Основные понятия качества и метрической теории ПО. Модели оценки отдельных модулей ПО. Свойства метрик проекта.</p> <p>Критерии качества ПО (по фазам жизненного цикла проекта).</p> <p>Методы оценки качества ПО (анкетирование, рабочие списки, бенчмарки).</p>
6.2. Измерения в программном проекте	<p>Измерения в программной инженерии: требования к измерениям, измерения (метрики), ресурсы для измерений. Планирование измерений, процедуры сбора/анализа/отчетности, критерии оценки ПП, инструменты. Оценивание измерений. Шкалы измерений (номинальная, порядковая, интервальная, шкала соотношений).</p>
7. Технологический цикл разработки программных систем, коллективная работа по созданию программ, организация процесса разработки и инструментальные средства поддержки	<p>7.1. Технологический цикл разработки программных систем. Организация процесса качественного программного продукта и управление качеством. Инструментарий отслеживания программного проекта и контроля качества.</p> <p>7.2. Методы коллективной разработки программного продукта. Организационная структура программистского коллектива и оценка ее эффективности. Программистская бригада: распределение ролей. Оценка вклада отдельного программиста в работу коллектива. Классификация программистских организаций по уровню качества выпускаемого продукта. Методы контроля и улучшения процесса. Стандарты серии ISO и CMMI. Методы управления степенью совершенства процесса и инструментальные средства поддержки.</p> <p>Современные индустриальные технологии программирования ПО (сборочная ТП, особенности ТП управляющих систем, отказоустойчивых систем, распределенных систем и сетей).</p>
8. Перспективные направления развития технологии программирования	<p>Перспективные направления в развитии ТП: доказательное программирование и визуальное программирование.</p> <p>Метатехнология в программировании больших программных комплексов.</p>

5. Образовательные технологии

В изучении дисциплины используются такие образовательные технологии, как лекции, практические занятия, разработка курсового проекта. При этом используются следующие аспекты современных образовательных технологий: лекционный материал подается с применением презентаций; предполагается интерактивное обсуждение отдельных тем курса; предусмотрено промежуточное аудиторное или онлайн тестирование по темам курса с целью

контроля уровня знаний студентов. Часть практических занятий реализуются в интерактивной форме с элементами индивидуального обучения и коллективного разбора результатов заданий. При выполнении курсового проекта предусмотрена работа в команде, самостоятельная работа студентов. При необходимости в отдельные периоды времени применяется электронное обучение, дистанционные образовательные технологии.

6. Лабораторный практикум

Не предусмотрено

7. Практические занятия

№ раздела	Наименование практических занятий (семинаров)	Трудоемкость, ач
		Очная форма
1.	Жизненный цикл (ЖЦ) программного обеспечения (ПО), этапы производства ПО. Выбор модели ЖЦ и инструментальных средств для курсового проекта.	2
2.	Требования к ПО. Различные виды требований (в качестве основных видов - функциональные и не функциональные требования). Требования к ПО с точки зрения его сопровождения.	2
3.	Проектирование ПО (высокоуровневое и детальное). Различные виды представления результатов проектирования ПО	2
4.	Конструирование (кодирование) ПО; документирование (виды сопроводительной документации); аспекты кодирования ПО с учетом будущего сопровождения	2
5.	Тестирование и отладка ПО	2
6.	Автоматизация тестирования; тестирование производительности, нагружочное тестирование. Контейнерная виртуализация (Docker, Kubernetes).	3
7.	Технологический цикл разработки программных систем, коллективная работа по созданию программ, организация процесса разработки и инструментальные средства поддержки	1
8.	Управление разработкой программного проекта: измерения ПО (метрики), критерии качества ПО	2
Итого часов		16

8. Организация и учебно-методическое обеспечение самостоятельной работы

Для закрепления знаний, умений и получения новых навыков работы с технологиями разработки программного обеспечения проводится курсовое проектирование. Курсовая работа (проект) выполняется либо по одному из вариантов, предложенных преподавателем, либо

используется предложение бригады студентов для разработки конкретного приложения от требований до проверки его работоспособности и оценки качества приложения.

Для реализации курсового проекта будет использоваться теоретический материал, полученный студентами во время лекций, самостоятельной работы и практических занятий. Процесс выполнения курсового задания и правила оформления отчета описаны в методических указаниях и дополнительно поясняются на занятиях. Выполнение курсового проекта проводится для закрепления учебного материала, а также для развития профессиональных умений и навыков и повышения творческого потенциала студентов. Обсуждение промежуточных результатов работы и консультации по выполнению отдельных пунктов работы проводится на практических занятиях и в виде вопросов-ответов по мере выполнения курсового проекта.

Примерное распределение времени самостоятельной работы студентов

Вид самостоятельной работы	Примерная трудоемкость, ач
	Очная форма
Текущая СР	
работа с лекционным материалом, с учебной литературой	14
опережающая самостоятельная работа (изучение нового материала до его изложения на занятиях)	4
самостоятельное изучение разделов дисциплины	10
выполнение домашних заданий, домашних контрольных работ	10
подготовка к лабораторным работам, к практическим и семинарским занятиям	5
подготовка к контрольным работам, коллоквиумам	5
Итого текущей СР:	48
Творческая проблемно-ориентированная СР	
выполнение расчётно-графических работ	0
выполнение курсового проекта или курсовой работы	52
поиск, изучение и презентация информации по заданной проблеме, анализ научных публикаций по заданной теме	5
работа над междисциплинарным проектом	0
исследовательская работа, участие в конференциях, семинарах, олимпиадах	6
анализ данных по заданной теме, выполнение расчётов, составление схем и моделей на основе собранных данных	0
Итого творческой СР:	63
Общая трудоемкость СР:	111

9. Учебно-методическое обеспечение дисциплины

9.1. Адрес сайта курса

<https://dl.spbstu.ru/course/view.php?id=1857>

9.2. Рекомендуемая литература

Основная литература

№	Автор, название, место издания, издательство, год (годы) издания	Год изд.	Источник
1	Котляров В.П., Петров А.В. Автоматизация формализации требований к программным проектам // Научно-технические ведомости СПбГПУ. 2009. №3(80) : Информатика. Телекоммуникации. Управление	2009	ИБК СПбПУ
2	Карпов А.Н., Котляров В.П. Технология настраиваемой генерации тестов по формальным спецификациям для встроенных приложений и программных интерфейсов, реализованных на Java-подобных языках, 2007. URL: http://elib.spbstu.ru/dl/1297.pdf	2007	ЭБ СПбПУ

Дополнительная литература

№	Автор, название, место издания, издательство, год (годы) издания	Год изд.	Источник
1	Кнут Д.Э. Искусство программирования. Т.3 Сортировка и поиск: М.: Вильямс, 2000.	2000	ИБК СПбПУ
2	Котляров В.П. Основы современного тестирования программного обеспечения, разработанного на C#, 2012. URL: http://elib.spbstu.ru/dl/local/2385.pdf	2012	ЭБ СПбПУ
3	Дробинцев П.Д. и др. Подход к тестированию параллельных систем на основе UCM-спецификаций // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. 2014. № 3 (198) URL: http://elib.spbstu.ru/dl/2/4685.pdf	2014	ЭБ СПбПУ

Ресурсы Интернета

1. Кознов Введение в программную инженерию: <https://www.intuit.ru/studies/courses/497/353/lecture/8405/>
2. Kubernetes. Официальные учебные пособия (tutorials) (без песочницы): <https://kubernetes.io/docs/tutorials>
3. Kubernetes: Up and Running: <https://www.oreilly.com/library/view/kubernetes-up-and-9781491935668>
4. Docker: учебные пособия (tutorials) с практикой: <https://training.play-with-docker.com/>

9.3. Технические средства обеспечения дисциплины

На компьютеры в учебном классе должен быть инсталлирован MS Office или аналогичный свободно распространяемый инструмент (например, LibreOffice или P-7 Офис). Необходим доступ в Интернет для просмотра студенческих программных проектов в облачных хранилищах. При согласовании приложения по инициативе студентов согласуется и набор используемых инструментов для разработки конкретного приложения. В качестве примера можно привести наиболее популярные языки программирования (Java, C++, JavaScript, C#, Python, Kotlin) и IDE для них, а также фреймворки для тестирования и другие необходимые инструменты разработки. В случае инициативного выбора средств разработки студенты демонстрируют результаты на своих компьютерах или в облачном хранилище.

При дистанционном режиме обучения необходимо использовать инструменты, которые будут доступны студентам без дополнительной оплаты. Решение принимается в соответствии с конкретными обстоятельствами и условиями.

10. Материально-техническое обеспечение дисциплины

Для успешного проведения занятий необходимо использование компьютерного класса, имеющего не менее 10 - 12 компьютеров, оснащенных необходимым программным обеспечением (см.п. технические средства обеспечения дисциплины), локальной сети с сервером на 10-14 рабочих мест. Для проведения лекций необходим класс с наличием мультимедийного проектора.

При использовании дистанционных образовательных технологий и электронного обучения необходимы программные инструменты, одобренные приказами университета и института, а также те, с помощью которых студенты могут выполнить курсовое проектирование.

11. Критерии оценивания и оценочные средства

11.1. Критерии оценивания

Для дисциплины «Технологии разработки качественного программного обеспечения» предусмотрены следующие формы аттестации: зачёт, экзамен. Дисциплина реализуется с применением системы индивидуальных достижений.

Текущий контроль успеваемости

Максимальное значение персонального суммарного результата обучения (ПСРО) по приведенной шкале - 100 баллов

Максимальное количество баллов приведенной шкалы по результатам прохождения двух точек контроля - 80 баллов.

Подробное описание правил проведения текущего контроля с указанием баллов по каждому контрольному мероприятию и критериев выставления оценки размещается в СДО в навигационном курсе дисциплины.

Промежуточная аттестация по дисциплине

Максимальное количество баллов по результатам проведения аттестационного испытания в период промежуточной аттестации – 20 баллов приведенной шкалы.

Промежуточная аттестация по дисциплине проводится в соответствии с расписанием.

1. Форма проведения зачета: устное собеседование с учетом портфолио обучающегося и его достижений.

Критерии оценки: оценка «зачтено» выставляется обучающемуся при условии получения оценок «зачтено» по всем выполненным им в процессе изучения дисциплины работам, входящим в состав портфолио, а также при условии получения оценок «зачтено» по тестам/контрольным работам.

Портфолио включает самостоятельные работы, выполненные в течение семестра, результаты тестов/контрольных работ, написанных в аудитории или в электронном виде. Получение оценок «зачтено» за все (или не менее 75%) предусмотренных программой заданий является основанием проведения промежуточной аттестации по дисциплине.

2. Форма проведения экзамена: устное собеседование по билетам; каждый билет содержит 3 вопроса.

Количество вопросов может быть уменьшено с учетом портфолио студента (результаты выполненных заданий и контрольных в течение семестра).

Критерии оценки: максимальная оценка за прохождение промежуточной аттестации – 5.

Оценка и общие требования к оценке:

5 - даны полные, аргументированные ответы с примерами на все вопросы билета и на дополнительные вопросы, при их наличии; продемонстрировано глубокое понимание теоретического материала.

4 - ответы полные, но имеются незначительные неточности, а также незначительные затруднения с ответами на дополнительные вопросы по теме.

3 - ответы фрагментарны, основные положения раскрыты поверхностно, затрудняется ответить на дополнительные вопросы по теме вопросов билета.

2 - не раскрыты ключевые аспекты вопросов и дополнительных вопросов, обнаружены серьезные пробелы в базовом теоретическом материале; данная оценка выставляется также, если студент после начала экзамена отказался его сдавать или нарушил правила сдачи экзамена.

Дополнительные параметры:

Время ответа: 10-15 мин. для зачета, 15-18 мин. для экзамена.

Разрешается использование:

- личных конспектов,
- материалов дисциплины, предоставленных преподавателем.

Запрещается:

- подсказки третьих лиц.

При использовании дистанционного режима зачет и экзамен принимаются с использованием средств Интернета способом, приближенном к обычному.

3. Форма проведения защиты курсового проекта.

Работа над курсовым проектом происходит в два этапа. Для 7-го семестра для оценки "зачтено" необходимо продемонстрировать реализованный проект, соответствующий требованиям, но не проверенный с помощью различных видов тестирования. Для 8-го семестра необходимо предъявить реализованный проект, соответствующий первоначальным требованиям или

изменённым по согласованию с преподавателем, и проверенный видами тестов, указанных в задании. Критерии оценивания описываются в соответствующем навигационном курсе.

Результаты промежуточной аттестации, определяются на основе баллов, набранных в рамках применения, СИД

Баллы по приведенной шкале в рамках применения СИД (ПСРО+ ПА)	Оценка по результатам промежуточной аттестации
	Экзамен/диф.зачет/зачет
0 - 60 баллов	Неудовлетворительно/не зачтено
61 - 75 баллов	Удовлетворительно/зачтено
76 - 89 баллов	Хорошо/зачтено
90 и более	Отлично/зачтено

11.2. Оценочные средства

Оценочные средства по дисциплине представлены в фонде оценочных средств, который является неотъемлемой частью основной образовательной программы и размещается в электронной информационно-образовательной среде СПбПУ на портале etk.spbstu.ru

12. Методические рекомендации по организации изучения дисциплины

Методические рекомендации по каждому разделу приводятся в лекциях в качестве дополнительных слайдов, со ссылкой на рекомендуемую литературу для самостоятельного изучения, и предварительного списка вопросов или заданий для самостоятельной работы. Успешное изучение дисциплины требует посещения лекций, выполнения всех учебных заданий (практических и выбранных для курсового проектирования), ознакомления с основной и дополнительной литературой.

Разбор наиболее часто встречающихся ошибок при выполнении самостоятельных заданий происходит с помощью преподавателя в часы аудиторных занятий или в виде вопросов-ответов в онлайн режиме.

Рекомендуется изучать материал по мере его прочтения в лекциях. Если тема или отдельные вопросы по теме вызывают затруднения, необходимо постараться найти ответы на них, используя рекомендуемую литературу. Если студенту самостоятельно не удалось разобраться в материале, необходимо сформулировать вопросы и обратится за помощью к преподавателю на

ближайшем занятии или задать вопрос, используя предоставленный преподавателем способ коммуникации с ним. Обучающемуся необходимо регулярно отводить время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам. Практические занятия и курсовой проект составляют важную часть профессиональной подготовки студентов. Они направлены на подтверждение теоретических положений и формирование учебных и профессиональных практических навыков, умений.

Для более хорошего усвоивания материала особенно плохо успевающими студентами, возможно добавление дополнительного часа в неделю для факультативных занятий.

13. Адаптация рабочей программы для лиц с ОВЗ

Адаптированная программа разрабатывается при наличии заявления со стороны обучающегося (родителей, законных представителей) и медицинских показаний (рекомендациями психолого-медико-педагогической комиссии). Для инвалидов адаптированная образовательная программа разрабатывается в соответствии с индивидуальной программой реабилитации.