

**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)**

**«Архитектура программных систем»**

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	<b>бакалавр</b>
Образовательный стандарт	<b>СУОС</b>
Форма обучения	<b>Очная</b>

**СОГЛАСОВАНО**

Руководитель ОП

\_\_\_\_\_ А.В. Петров

«01» апреля 2025 г.

Соответствует СУОС

Утверждена протоколом заседания

высшей школы "ВШПИ"

от «01» апреля 2025 г. № 1

РПД разработали:

Старший преподаватель Д.Ф. Дробинцев

Доцент, к.т.н. И.А. Селин

Старший преподаватель А.В. Гончаров

# 1. Цели и планируемые результаты изучения дисциплины

## Цели освоения дисциплины

Целями изучения дисциплины "Архитектура программных систем»" являются: 1. Формирования специалистов, умеющих обоснованно и результативно применять знание принципов и методов разработки моделей вычислительных процессов вычислительных систем (ВС) и многомашинных ассоциаций (ММА). 2. Формирование навыков проектирования автоматизированных систем разного уровня сложности, с учётом современных шаблонов проектирования, непосредственно применяемых на практике в ИТ-индустрии. В рамках дисциплины рассматриваются фундаментальные модели архитектуры многопроцессорных и многомашинных ВС и сетей, основные модели и алгоритмы организации ВС, теоретические основы и практика моделирования ВС и ММА.

## Результаты обучения выпускника

Код	Результат обучения (компетенция) выпускника ООП
ПК-2	Способен выбирать модель жизненного цикла программного обеспечения, выполнять проектирование и ревизию его архитектуры
ИД-2 ПК-2	Применяет общие принципы проектирования программного обеспечения с учетом контекста выбранного жизненного цикла разработки и сопровождения
ПК-8	Способен создать архитектуру разрабатываемой программной системы, удовлетворяющую функциональным требованиям и соответствующую критериям по производительности, безопасности, мобильности и масштабируемости, в рамках которой обеспечивается взаимодействие компонент на основе популярных протоколов, поддерживается информационная база данных
ИД-1 ПК-8	Производит декомпозицию системы на компоненты и осуществляет оценку и выбор типа каждого компонента и архитектуры развертывания, слоев и шаблонов проектирования и технологии обработки данных
ИД-2 ПК-8	Производит описание технологических и технико-эксплуатационных характеристик возможных типов для каждого компонента, протоколов их взаимодействия, шаблонов проектирования, спецификаций безопасности, моделей обеспечения производительности компонентов, структуры и технологии обработки данных программного средства

## **Планируемые результаты изучения дисциплины**

### **знания:**

- Знает основные этапы проектирования программного обеспечения в контексте выбранной модели жизненного цикла разработки ПО
- Знает основные принципы декомпозиции
- Знает основные принципы описания характеристик компонентов, протоколов их взаимодействия, шаблонов проектирования и спецификаций

### **умения:**

- Умеет выполнять декомпозицию разрабатываемого программного обеспечения на компоненты с учетом выбранной модели жизненного цикла разработки ПО
- Умеет декомпонировать алгоритм и программу на модули, обладающие явным описанием информационных зависимостей
- Умеет описывать технологические и технико-эксплуатационные характеристики компонентов

### **навыки:**

- Владеет навыками работы с инструментами для описания проектных решений

## **2. Место дисциплины в структуре ООП**

В учебном плане дисциплина «Архитектура программных систем» относится к модулю «Модуль цифровых компетенций».

Изучение дисциплины базируется на результатах освоения следующих дисциплин:

- Введение в профессиональную деятельность
- Объектно-ориентированное программирование

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

#### 3.1. Виды учебной работы

Виды учебной работы	Трудоёмкость по семестрам
	Очная форма
Лекционные занятия	60
Практические занятия	42
Самостоятельная работа	124
Промежуточная аттестация (зачет)	4
Промежуточная аттестация (зачет с оценкой)	6
Курсовое проектирование	16
Общая трудоёмкость освоения дисциплины	252, ач
	7, зет

#### 3.2. Формы текущего контроля и промежуточной аттестации

Формы текущего контроля и промежуточной аттестации	Количество по семестрам
	Очная форма
<b>Текущий контроль</b>	
Контрольные, шт.	4
Курсовые проекты, шт.	1
Расчетно-графические работы, шт.	1
<b>Промежуточная аттестация</b>	
Зачеты, шт.	1
Зачеты с оценкой, 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.	Введение в корпоративную архитектуру. Основные понятия и области влияния. Виды архитектур.	3	0	0
2.	Погружение в корпоративную архитектуру. Основные артефакты и способы управления. Связь с бизнес-архитектурой.	2	0	11
3.	Оценка критичности систем и отказы.	3	2	5
4.	ЦОД и их роль в ИТ. Классификация.	4	2	11
5.	Системная архитектура. Отличия от корпоративной архитектуры.	2	2	5
6.	Знакомство с data-архитектурой. Основные фокусы и задачи. Примеры артефактов.	2	2	5
7.	Технологическая архитектура Основные фокусы и задачи. Примеры артефактов.	2	2	5
8.	Solution-архитектура. Основные фокусы и задачи. Примеры артефактов.	3	3	8
9.	Роль solution-архитектуры на примере классического цикла разработки ПО.	3	4	11
10.	Основные составляющие solution-архитектуры.	3	3	11
11.	Введение в требования. Функциональные (ФТ) и не функциональные (НФТ) требования. Способы достижения НФТ.	3	4	11
12.	Шаблоны проектирования и их роль. Разбор интеграционных, модульных, системных и прикладных шаблонов.	7	6	30
13.	Технические и архитектурные долги. Управление долгами.	3	4	3
14.	Работа с данными. Разбор принципов и гарантий различных СУБД.	4	3	3
15.	Способы проверки архитектуры на соответствие НФТ.	2	2	5
16.	Клиентские приложения. Классификация, связь с клиент-серверной архитектурой.	2	0	0

17.	Методы оптимизации доставки контента до конечного потребителя. Сети CDN.	2	1	0
18.	Связь архитектуры решения со способами организации работы ИТ-команд.	1	1	0
19.	Монолитная архитектура и её связь с этапами разработки программного обеспечения.	2	0	0
20.	История сервис-ориентированной архитектуры (SOA). Её связь с этапами разработки программного обеспечения.	1	0	0
21.	Способы перехода от монолитной архитектуры к сервисной.	2	1	0
22.	Введение в способы предоставления программного обеспечения конечным потребителям (IaaS, PaaS, SaaS).	2	0	0
23.	Введение в антипаттерны проектирования сервис-ориентированных систем.	1	0	0
24.	Предпосылки появления микросервисной архитектуры.	1	0	0
<b>Итого по видам учебной работы:</b>		60	42	124
Зачеты, ач				0
Зачеты с оценкой, ач				0
<b>Часы на контроль, ач</b>				0
<b>Курсовое проектирование</b>		16		
<b>Промежуточная аттестация (зачет)</b>		4		
<b>Промежуточная аттестация (зачет с оценкой)</b>		6		
<b>Общая трудоёмкость освоения: ач / зет</b>		252 / 7		

## 4.2. Содержание разделов и результаты изучения дисциплины

Раздел дисциплины	Содержание
<b>1. Введение в корпоративную архитектуру. Основные понятия и области влияния. Виды архитектур.</b>	1.1. Основные понятия и области влияния. Виды архитектур. 1.2. Виды архитекторов. Классификация слоев в современном предприятии. 1.3. Базовые принципы описания корпоративной архитектуры.
<b>2. Погружение в корпоративную архитектуру. Основные артефакты и способы управления. Связь с бизнес-архитектурой.</b>	2.1. Основные артефакты и способы управления. Связь с бизнес-архитектурой. 2.2. На какие вопросы отвечает архитектура предприятия. 2.3. Основные доменные области в зоне влияния корпоративной архитектуры. 2.4. Основные предпосылки к появлению корпоративной архитектуры.
<b>3. Оценка критичности систем и отказы.</b>	3.1. Описание шкалы критичности систем и введение в термины доступности. 3.2. Введение в отказы в обслуживании и устойчивость к ним. High availability, Fault tolerance, Disaster recovery. 3.3. Защита предприятий от отказов. Описание BCP и DRP.
<b>4. ЦОД и их роль в ИТ. Классификация.</b>	4.1. Введение в термины ЦОД. Классификация ЦОДов и связь с отказоустойчивостью предприятия. 4.2. Бюджет на ошибки в предприятии. Что это и в каких случаях его можно использовать. 4.3. Введение в понятие "Ус луга" раскрытие понятия SLI, SLO, SL.
<b>5. Системная архитектура. Отличия от корпоративной архитектуры.</b>	5.1. Введение в системную архитектуру. Описание составных слоев. 5.2. Соотношение составных слоев и корпоративной архитектуры.
<b>6. Знакомство с data-архитектурой. Основные фокусы и задачи. Примеры артефактов.</b>	6.1. На какие вопросы отвечает и на каких доменах фокусируется. 6.2. Описание возможных артефактов, описывающих архитектуру данных. 6.3. Предпосылки к появлению артефактов и методы построения архитектуры данных. 6.4. Иерархия моделирования данных: концептуальная, логическая и физическая схемы. ER, Data-flow диаграмма. 6.5. Роль архитектуры данных в предприятии.

<p><b>7. Технологическая архитектура</b>  <b>Основные фокусы и задачи.</b>  <b>Примеры артефактов.</b></p>	<p>7.1. Основные составляющие: программный, аппаратный и сетевой уровни. Отличие программной архитектуры от архитектуры приложений.  7.2. На какие вопросы отвечает. ЦОД, СХД, RPO/RTO, классы оборудования, системы виртуализации и контейнеризации.  7.3. Учет ИТ-активов и важность этого процесса. CMDB.  7.4. Зоны ответственности сетевого архитектора, архитектора программной и аппаратной платформ.</p>
<p><b>8. Solution-архитектура.</b>  <b>Основные фокусы и задачи.</b>  <b>Примеры артефактов.</b></p>	<p>8.1. Основные фокусы (интеграция, технологии, frameworks, качество кода, архитектурные шаблоны, долги).  8.2. Примеры артефактов.  8.3. Анализ требований (бизнес, системные).  8.4. Концепция управления технологиями через Technology Radar.  8.5. Зона ответственности solution-архитектора.</p>
<p><b>9. Роль solution-архитектуры на примере классического цикла разработки ПО.</b></p>	<p>9.1. Важные свойства автоматизированных систем и их влияние на стоимость итогового решения.  9.2. Жизненный цикл разработки (SDLC) и роль solution-архитектуры на каждом из этапов.  9.3. Описание сквозного пути проектирования решений задач бизнеса (solution journey map).</p>
<p><b>10. Основные составляющие solution-архитектуры.</b></p>	<p>10.1 Идея управления архитектурой во времени.  10.2. Описание основных доменов solution-архитектуры (БТ/ФТ, ЯП, API, gitflow, CI/CD, Метрики, шаблоны проектирования, презентации и защита решений перед бизнесом).  10.3. Методы оценки эффективности программных решений.  10.4. Принципы эффективного архитектора.  10.5. Раскрытие ответственности solution-архитектора в виде подготовки "решения под ключ".  10.6. Путь клиентского запроса (web) через все слои архитектуры.</p>
<p><b>11. Введение в требования. Функциональные (ФТ) и не функциональные (НФТ) требования. Способы достижения НФТ.</b></p>	<p>11.1. Классификация требований по К.Вигерсу.  11.2. Степень влияния ФТ и НФТ на архитектуру.  11.3. Разбор требований.  11.4. Примеры бизнес-правил, атрибутов качества, внешнего интерфейса, ограничений.</p>

<b>12. Шаблоны проектирования и их роль. Разбор интеграционных, модульных, системных и прикладных шаблонов.</b>	<p>12.1. Классификация шаблонов проектирования: интеграционные, модульные, системные, прикладные.</p> <p>12.2. Разбор интеграционных шаблонов: file transfer, RPC, shared database, messaging.</p> <p>12.3. Отличие Mtssaging broker от Persistent long систем.</p> <p>12.4. Уровни "Гарантии доставки сообщений".</p> <p>12.5. Примеры модульных шаблонов. Порождающие, структурные и поведенческие шаблоны. Варианты организации разработки WEB API.</p> <p>12.6. Введение в работу с состояниями приложений: stateless, stateful, serverless.</p> <p>12.7. Примеры системных шаблонов. Монолитная архитектура, SOA и микросервисная архитектура. Обзор системы акторов.</p> <p>12.8. Примеры прикладных шаблонов. Балансировка нагрузки.</p> <p>12.9. Обзор антипаттернов проектирования.</p>
<b>13. Технические и архитектурные долги. Управление долгами.</b>	<p>13.1. Введение в определение долгов. Принципиальное отличие между техническим и архитектурным долгом.</p> <p>13.2. Способы управления долгами.</p>
<b>14. Работа с данными. Разбор принципов и гарантий различных СУБД.</b>	<p>14.1. Важность проектирования данных. Влияние на Application слой.</p> <p>14.2. Классификация хранилищ. РСУБД/NoSQL, OLTP/OLAP.</p> <p>14.3. CAP теорема. Отличие от ACID и BASE принципов. Введение в проблему согласованности данных в распределенных СУБД.</p> <p>14.4. Изоляция и нормализация в РСУБД.</p> <p>14.5. Бизнес-анализ на основе данных. Введение в DWH и Data Lake. Основы BI.</p>
<b>15. Способы проверки архитектуры на соответствие НФТ.</b>	<p>15.1. Классификация и способы расчета метрик для автоматизированных систем.</p> <p>15.2. Оценка надежности системы через нагрузку. Введение в нагрузочное и стресс-тестирование.</p>
<b>16. Клиентские приложения. Классификация, связь с клиент-серверной архитектурой.</b>	<p>Клиентские приложения. Классификация, связь с клиент-серверной архитектурой. SuperApp. Толстый клиент, тонкий клиент. Уведомления.</p>
<b>17. Методы оптимизации доставки контента до конечного потребителя. Сети CDN.</b>	<p>Нюансы клиентов и распределенных ресурсов. Content Delivery Network (CDN). Origin и Edge сервера, инфраструктура .</p> <p>Динамический и статический контент. Кеширование.</p> <p>Визуализация контента.</p>

<b>18. Связь архитектуры решения со способами организации работы ИТ-команд.</b>	Организация цикла разработки программного обеспечения. Подходы к организации (Waterfall, DevOps, V-Model, Incremental SDLC, Iterative SDLC, RAD)
<b>19. Монолитная архитектура и её связь с этапами разработки программного обеспечения.</b>	Этапы цикла разработки монолита. Проектирование, разработка, тестирование, выпуск новой версии, поддержка, эксплуатация. Когда нужен монолит?
<b>20. История сервис-ориентированной архитектуры (SOA). Её связь с этапами разработки программного обеспечения.</b>	История SOA. Архитектурный стиль и его основные принципы. Цели SOA и подходы к реализации (top-down или bottom-up). ESB.
<b>21. Способы перехода от монолитной архитектуры к сервисной.</b>	Выделение контекстов. Проектирование. Отработка нюансов. Реализация. Управление миграцией. Результат.
<b>22. Введение в способы предоставления программного обеспечения конечным потребителям (IaaS, PaaS, SaaS).</b>	X as a service. On-premise (свое облако). Infrastructure as a service. Platform as a service. Software as a service.
<b>23. Введение в антипаттерны проектирования сервис-ориентированных систем.</b>	Loose coupling. High cohesion. Single purpose/responsibility. N- слойная архитектура. Feature-sliced. Разделение ответственности.
<b>24. Предпосылки появления микросервисной архитектуры.</b>	Предпосылки появления микросервисной архитектуры. Связь с сервис-ориентированной архитектурой. Базовые идеи SOA.

## 5. Образовательные технологии

1. В изучении дисциплины «Архитектура программных средств используются традиционные образовательные технологии, такие как: – лекции, – лабораторный практикум. Вместе с тем, аспектами современных образовательных технологий является следующее: - лекционный материал подается с применением презентаций с использованием мультимедийного проектора и электронной доски; предполагается интерактивное обсуждение отдельных тем курса и чтение лекций с диалогом со студентами, повышая вовлечённость аудитории; - предусмотрено промежуточное аудиторное тестирование по темам курса с целью контроля уровня знаний студентов; - предусматривается опережающая самостоятельная работа студентов; - курс предусматривает выполнение и защиту курсового проекта. Часть лабораторных работ реализуются в интерактивной форме с элементами индивидуального обучения и опережающей самостоятельной работы. 2. Также, в рамках курса реализована программа геймификации, включающая выполнение мини-заданий в домашних условиях с последующей защитой

результата после лекции. Успешно выполненные мини-задания позволяют студентам получить баллы, которые будут учтены по усмотрению преподавателя.

## 6. Лабораторный практикум

Не предусмотрено

## 7. Практические занятия

№ раздела	Наименование практических занятий (семинаров)	Трудоемкость, ач
		Очная форма
1.	Распространенные архитектуры современных компьютеров	4
2.	Модели вычислительных процессов. Сети Петри	6
3.	Методы и средства программирования синхронизации процессов	5
4.	Основы имитационного моделирования ВС.	12
5.	Реализация вычислительных процессов в операционной системе (ОС)	3
6.	Методы и алгоритмы управления ресурсом времени в вычислительном процессе	4
7.	Методы и алгоритмы управления ресурсом памяти в вычислительном процессе	2
8.	Методы и алгоритмы управления ресурсом внешних устройств в вычислительном процессе	2
9.	Оценки эффективности функционирования ВС и сетей	4
Итого часов		42

## 8. Организация и учебно-методическое обеспечение самостоятельной работы

## Примерное распределение времени самостоятельной работы студентов

Вид самостоятельной работы	Примерная трудоемкость, ач
	Очная форма
<b>Текущая СР</b>	
работа с лекционным материалом, с учебной литературой	16
опережающая самостоятельная работа (изучение нового материала до его изложения на занятиях)	11
самостоятельное изучение разделов дисциплины	16
выполнение домашних заданий, домашних контрольных работ	0
подготовка к лабораторным работам, к практическим и семинарским занятиям	8
подготовка к контрольным работам, коллоквиумам	19
<b>Итого текущей СР:</b>	<b>70</b>
<b>Творческая проблемно-ориентированная СР</b>	
выполнение расчётно-графических работ	0
выполнение курсового проекта или курсовой работы	24
поиск, изучение и презентация информации по заданной проблеме, анализ научных публикаций по заданной теме	14
работа над междисциплинарным проектом	0
исследовательская работа, участие в конференциях, семинарах, олимпиадах	5
анализ данных по заданной теме, выполнение расчётов, составление схем и моделей на основе собранных данных	11
<b>Итого творческой СР:</b>	<b>54</b>
<b>Общая трудоемкость СР:</b>	<b>124</b>

## 9. Учебно-методическое обеспечение дисциплины

### 9.1. Адрес сайта курса

<https://dl.spbstu.ru/course/view.php?id=5502>

## 9.2. Рекомендуемая литература

### Основная литература

№	Автор, название, место издания, издательство, год (годы) издания	Год изд.	Источник
1	Юсупов Ю.В., Котляров В.П. Автоматизация построения формальных поведенческих моделей из программного кода // Научно-технические ведомости СПбГПУ. 2008. №4(63): Основной выпуск	2008	ИБК СПбПУ
2	Птицына Л.К., Соколова Н.В. Параллельные вычислительные процессы в системах мониторинга и управления: Санкт-Петербург: Изд-во Политехн. ун-та, 2008.	2008	ИБК СПбПУ

### Ресурсы Интернета

1. НОУ ИНТУИТ | Теория и практика параллельных вычислений ....: <https://www.intuit.ru/studies/courses/1156/190/info>

## 9.3. Технические средства обеспечения дисциплины

Для успешного выполнения лабораторных работ и курсового проекта необходимо наличие VS Code (или OpenIDE), MS Office 2007 или новее (OpenOffice), Linux

## 10. Материально-техническое обеспечение дисциплины

Для успешного проведения лабораторных занятий необходимо использование компьютерного класса, имеющего не менее 10 компьютеров, оснащенных необходимым программным обеспечением и локальной сети с сервером на 8-10 рабочих мест. Для проведения лекций необходим класс с наличием мультимедийного проектора.

## 11. Критерии оценивания и оценочные средства

### 11.1. Критерии оценивания

Для дисциплины «Архитектура программных систем» предусмотрены следующие формы аттестации: зачёт, зачёт с оценкой. 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.

Критерии оценивания результата контрольных работ следующие:

- Оценки **«отлично»** заслуживает студент, правильно выполнивший все задания контрольной работы;
- Оценки **«хорошо»** заслуживает студент, правильно выполнивший не менее 75% заданий из возможных;
- Оценки **«удовлетворительно»** заслуживает студент, правильно выполнивший не менее 50% заданий из возможных;
- Оценки **«неудовлетворительно»** заслуживает студент, выполнивший правильно менее 50% заданий из возможных.

Критерии оценивания полученных умений и навыков во время выполнения практических занятий следующие:

- Оценки **«отлично»** заслуживает студент, проделавший все практические задания, активно участвующий в обсуждениях результатов выполненной работы, понимающий процесс

выполнения работы, готовый на основе проделанной работы выполнить самостоятельно предложенное индивидуальное задание по теме;

- Оценки «**хорошо**» заслуживает студент, проделавший от 90% до 100% практических заданий, участвующий в обсуждениях результатов выполненной работы, понимающий процесс выполнения работы, готовый выполнить подобное задание;
- Оценки «**удовлетворительно**» заслуживает студент, проделавший от 70% до 100% практических заданий, участвующий в обсуждениях результатов выполненной работы, понимающий процесс выполнения работы, готовый повторить выполненное задание;
- Оценки «**неудовлетворительно**» заслуживает студент, не выполнивший учебную программу по данной дисциплине (количество невыполненных лабораторных работ меньше 70%).

Формой промежуточной аттестации по данной дисциплине является зачет в 5 семестре, курсовой проект и экзамен в 6 семестре.

Критерии оценивания устного ответа:

- Оценки «**отлично**» заслуживает студент, обнаруживший всестороннее, систематическое и глубокое знание материалов изученной дисциплины, умение свободно выполнять задания, предусмотренные программой, усвоивший основную и знакомый с дополнительной литературой, рекомендованной рабочей программой дисциплины; проявивший творческие способности в понимании, изложении и использовании материалов изученной дисциплины, безупречно ответившему не только на вопросы билета, но и на дополнительные вопросы в рамках рабочей программы дисциплины, правильно выполнившему практическое задание.
- Оценки «**хорошо**» заслуживает студент, обнаруживший полное знание материала изученной дисциплины, успешно выполняющий предусмотренные задания, усвоивший основную литературу, рекомендованную рабочей программой дисциплины; показавшему систематический характер знаний по дисциплине, ответившему на все вопросы билета, правильно выполнившему практическое задание, но допустившему при этом не принципиальные ошибки.
- Оценки «**удовлетворительно**» заслуживает студент, обнаруживший знание материала изученной дисциплины в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии, справляющийся с выполнением заданий, знакомы с основной литературой, рекомендованной рабочей программой дисциплины; допустившим погрешность в ответе на теоретические вопросы и/или при выполнении практических заданий, но обладающий необходимыми знаниями для их устранения под руководством преподавателя, либо неправильно выполнившему практическое задание, но по указанию преподавателя выполнившим другие практические задания из того же раздела дисциплины.

- Оценка «**неудовлетворительно**» выставляется студенту, обнаружившему серьезные пробелы в знаниях основного материала изученной дисциплины, допустившему принципиальные ошибки в выполнении заданий, не ответившему на все вопросы билета и дополнительные вопросы и неправильно выполнившему практическое задание (неправильное выполнение только практического задания не является однозначной причиной для выставления оценки «неудовлетворительно»). Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение по образовательной программе без дополнительных занятий по соответствующей дисциплине (формирования и развития компетенций, закреплённых за данной дисциплиной). Оценка «неудовлетворительно» выставляется также, если студент после начала экзамена отказался его сдавать или нарушил правила сдачи экзамена

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 СПбГУ на портале [etk.spbstu.ru/](http://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 и лабораторных работ, ознакомления с основной и дополнительной литературой.

Разбор наиболее часто встречающихся ошибок при выполнении самостоятельных заданий происходит с помощью преподавателя в часы аудиторных занятий.

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 обратиться за помощью к преподавателю на ближайшей 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 курсовая работа составляют важную часть профессиональной подготовки студентов. Они направлены на подтверждение теоретических положений и формирование учебных и профессиональных практических умений.

Для более хорошего усваивания материала, особенно плохо успевающими студентами, предлагается добавить дополнительные часы (1 час в неделю) для факультативных лабораторных занятий. На этих занятиях можно ускорить освоение материала по использованию современных технологий программирования вычислительных процессов.

### **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.