

**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)

«Практикум по программированию»

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

Руководитель ОП

_____ А.В. Петров

«01» апреля 2025 г.

Соответствует СУОС

Утверждена протоколом заседания

высшей школы "ВШПИ"

от «01» апреля 2025 г. № 1

РПД разработал:

Старший преподаватель Т.Н. Самочадина

1. Цели и планируемые результаты изучения дисциплины

Цели освоения дисциплины

Цели дисциплины - получение практических навыков создания программ, знакомство с основными принципами процедурного программирования и алгоритмами обработки данных , освоение объектно-ориентированного программирования, умение кодировать алгоритмы на языках высокого уровня (C++, Java)

Результаты обучения выпускника

Код	Результат обучения (компетенция) выпускника ООП
ОПК-6	Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов
ИД-1 ОПК-6	Разрабатывает спецификации программного обеспечения в соответствии с поставленной задачей; разрабатывает алгоритмы решения задач в соответствии с требованиями
ИД-2 ОПК-6	Реализует программный код на языке высокого уровня и выполняет его проверку в соответствии с разработанной спецификацией
ИД-3 ОПК-6	Использует различные структуры данных и алгоритмы работы с ними для решения поставленных задач

Планируемые результаты изучения дисциплины

знания:

- Знает методы и приемы формализации поставленных задач; приемы анализа требований и формирования детальных требований к задаче; методы и приемы алгоритмизации поставленных задач
- Знает методологии структурного , объектно-ориентированного и обобщённого программирования; синтаксис выбранного языка программирования; особенности программирования на выбранном языке; стандартные библиотеки языка программирования; технологию создания программ в выбранной среде
- Знает методы анализа алгоритмов, понятие асимптотическая сложность алгоритма, роль абстрактных типов данных в процессе разработки алгоритма, структуры данных, алгоритмы поиска и сортировки, методы разработки алгоритмов

умения:

- Умеет выбирать и обосновывать выбор метода решения задачи; разрабатывать детальные требования к задаче; применять алгоритмы решения типовых задач
- Умеет реализовывать алгоритмы с использованием выбранной методологии; выявлять ошибки в программном коде; применять методы и приемы отладки программного кода
- Умеет оценивать сложность алгоритмов поиска и сортировки, реализовывать структуры данных

навыки:

- Владеет методами и приемами формализации задач; методами и приемами алгоритмизации задач
- Владеет навыками реализации алгоритмов на языке высокого уровня, навыками анализа и проверки исходного программного кода; навыками отладки программного кода
- Владеет навыками использования структур и алгоритмов для решения различных задач

2. Место дисциплины в структуре ООП

В учебном плане дисциплина «Практикум по программированию» не связана ни с одним модулем учебного плана.

Изучение дисциплины требует знания школьной программы, успешной сдачи вступительных или единых государственных экзаменов.

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
Практические занятия	28
Самостоятельная работа	36
Промежуточная аттестация (зачет)	8
Общая трудоёмкость освоения дисциплины	72, ач
	2, зет

3.2. Формы текущего контроля и промежуточной аттестации

Формы текущего контроля и промежуточной аттестации	Количество по семестрам
	Очная форма
Промежуточная аттестация	
Зачеты, шт.	2

4. Содержание и результаты обучения

4.1 Разделы дисциплины и виды учебной работы

№ раздела	Разделы дисциплины, мероприятия текущего контроля	Очная форма	
		Пр, ач	СР, ач
1.	Объектно-ориентированное программирование на языке Java. (4 семестр)		
1.1.	Основы Spring Projects. Сетевое взаимодействие. Реализация RESTfull-API.	6	6
1.2.	Основы Spring Framework. Аннотации Spring. Spring Security.	6	6
1.3.	Взаимодействие с базами данных. SQL и NoSQL решения. Использование Spring Data.	2	6
2.	Алгоритмизация и программирование (1 семестр)		

2.1.	Основные этапы решения задач на ЭВМ; критерии качества программы; жизненный цикл программы	2	2
2.2.	Интегрированная среда разработки программного обеспечения (Microsoft Visual Studio) -2	4	6
2.3.	Постановка задачи и спецификация программы; способы записи алгоритма	4	2
2.4.	Рекурсивные функции + ООП 2	4	8
Итого по видам учебной работы:		28	36
Зачеты, ач			0
Часы на контроль, ач			0
Промежуточная аттестация (зачет)		8	
Общая трудоёмкость освоения: ач / зет		72 / 2	

4.2. Содержание разделов и результаты изучения дисциплины

Раздел дисциплины	Содержание
1. Объектно-ориентированное программирование на языке Java. (4 семестр)	
1.1. Основы Spring Projects. Сетевое взаимодействие. Реализация RESTfull-API.	Реализация сетевого взаимодействия на Java. Краткий обзор средств и возможностей JavaEE. Обзор проектов входящих в Spring Projects. Принципы RESTful-API
1.2. Основы Spring Framework. Аннотации Spring. Spring Security.	Разбор аннотаций Spring Framework. Реализация REST-сервиса с использованием Spring Framework. Реализация безопасности приложения с использованием Spring Security.
1.3. Взаимодействие с базами данных. SQL и NoSQL решения. Использование Spring Data.	Общие принципы работы SQL и NoSQL баз данных. Проектирование простейшей базы данных, подключение и взаимодействие с ней при помощи Spring Data. Реализация CRUD-операций.
2. Алгоритмизация и программирование (1 семестр)	
2.1. Основные этапы решения задач на ЭВМ; критерии качества программы; жизненный цикл программы	Основные этапы решения задач на ЭВМ; критерии качества программ; жизненный цикл программ. Создание консольных приложений
2.2. Интегрированная среда разработки программного обеспечения (Microsoft Visual Studio) -2	Возможности интегрированной среды разработки IDE, разработка консольных приложений, работа с редактором исходного кода, интегрированным отладчиком, дизайнером классов.
2.3. Постановка задачи и спецификация программы; способы записи алгоритма	Понятие алгоритма. Исполнитель, система команд исполнителя. Свойства алгоритмов. Способы записи алгоритмов. Анализ, корректировка, детализация постановки задачи, разработка спецификации программы; применение различных способов записи алгоритмов. Составление спецификации, тест плана, записи алгоритмов
2.4. Рекурсивные функции + ООП 2	Определение рекурсивной функции. Шаг рекурсии. Рекурсивный вызов. Сравнение преимуществ и недостатков рекурсивного и итеративного подходов решения задачи. Разработка рекурсивных алгоритмов.

5. Образовательные технологии

1. Курс включает в себя лекции (3 семестр) , в которых сообщается весь фактический теоретический материал, практические занятия (2,3,4 семестры) на которых решаются задачи разработки алгоритмов и программ , рассматриваются все практические проблемы и на которых студенты решают индивидуальные задачи.
2. В преподавании курса используются преимущественно традиционные образовательные технологии: лекции и совместные групповые практические занятия . Практические занятия позволяют прорабатывать материал, получать навыки разработки программ, обсуждать со студентами возникающие вопросы. при необходимости в рамках практических занятий могут быть организованы индивидуальные консультации по имеющимся проблемам.

6. Лабораторный практикум

Не предусмотрено

7. Практические занятия

№ раздела	Наименование практических занятий (семинаров)	Трудоемкость, ач
		Очная форма
1.	Основные этапы решения задач на ЭВМ; критерии качества программы; жизненный цикл программы	2
2.	Интегрированная среда разработки программного обеспечения	4
3.	Постановка задачи и спецификация программы; способы записи алгоритма	4
4.	Рекурсивные функции	4
5.	Основы Spring Projects. Сетевое взаимодействие. Реализация RESTfull-API.	6
6.	Основы Spring Framework. Аннотации Spring. Spring Security.	6
7.	Взаимодействие с базами данных. SQL и NoSQL решения. Использование Spring Data.	2
Итого часов		28

8. Организация и учебно-методическое обеспечение самостоятельной работы

Примерное распределение времени самостоятельной работы студентов

Вид самостоятельной работы	Примерная трудоемкость, ач
	Очная форма
Текущая СР	
работа с лекционным материалом, с учебной литературой	6
опережающая самостоятельная работа (изучение нового материала до его изложения на занятиях)	0
самостоятельное изучение разделов дисциплины	0
выполнение домашних заданий, домашних контрольных работ	5
подготовка к лабораторным работам, к практическим и семинарским занятиям	12
подготовка к контрольным работам, коллоквиумам	8
Итого текущей СР:	31
Творческая проблемно-ориентированная СР	
выполнение расчётно-графических работ	0
выполнение курсового проекта или курсовой работы	0
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	0
анализ данных по заданной теме, выполнение расчётов, составление схем и моделей на основе собранных данных	0
Итого творческой СР:	5
Общая трудоемкость СР:	36

9. Учебно-методическое обеспечение дисциплины

9.1. Адрес сайта курса

<https://dl.spbstu.ru/course/view.php?id=5224>

9.2. Рекомендуемая литература

Основная литература

№	Автор, название, место издания, издательство, год (годы) издания	Год изд.	Источник
1	Павловская Т.А. С/C++. Процедурное и объектно-ориентированное программирование: Санкт-Петербург: Питер, 2015. URL: http://ibooks.ru/reading.php?short=1&productid=341427	2015	Подписное издание
2	Малыхина Г.Ф. Информационные технологии. Технологии Java: Санкт-Петербург: Изд-во Политехн. ун-та, 2008. URL: http://elib.spbstu.ru/dl/2/si20-1.pdf	2008	ЭБ СПбПУ

Дополнительная литература

№	Автор, название, место издания, издательство, год (годы) издания	Год изд.	Источник
1	Лафоре Р., Матвеев Е. Структуры данных и алгоритмы JAVA: М. [и др.]: Питер, 2011.	2011	ИБК СПбПУ
2	Лафоре Р. Объектно-ориентированное программирование в C++. Классика Computer Science (доп.тираж): Санкт-Петербург: Питер, 2011. URL: https://ibooks.ru/reading.php?short=1&productid=21961	2011	Подписное издание

Ресурсы Интернета

1. C++ Programming Style Guidelines: <http://geosoft.no/development/cppstyle.html>

9.3. Технические средства обеспечения дисциплины

Материал для практических занятий во 2 семестре представляет собой коллекцию задач и упражнений в виде файлов формата .pdf. В нем отражены варианты индивидуальных заданий, распределяемые преподавателем студентам. Требуется возможность чтения этого формата. Для проведения общих практических занятий требуется программное обеспечение - Microsoft Visual Studio 2019. Допустимо для самостоятельной работы студентов также использовать другие компиляторы языка C++ по договоренности с преподавателем.

Лекционный материал 3 семестра представлен в виде презентаций, разработанных в PowerPoint. Соответственно, на компьютере в лекционной аудитории должен быть установлен редактор подготовки презентаций, читающий этот формат. Для успешного проведения практических занятий в качестве программного обеспечения необходимо наличие Java Development Kit версии 8 и выше, а также инструмент управления проектами Maven.

Практические занятия в 4 семестре требуют того же программного обеспечения, что и предыдущем семестре.

10. Материально-техническое обеспечение дисциплины

Во 2 семестре для занятий требуются учебные классы, вмещающие 20-30 человек, оборудованные компьютерами с лицензионным программным обеспечением, с возможностью скачивания с сервера необходимых материалов в электронном виде (описание практических работ, дистрибутивы программ)

В 3 семестре для проведения лекций необходимо наличие аудитории на 100-150 посадочных мест, оборудованной компьютером и мультимедийным проектором для демонстрации лекционного материала. Для проведения практических занятий необходимо наличие компьютерного класса, вмещающего 20-30 человек, имеющего не менее 10-15 компьютеров, оснащенных необходимым лицензионным программным обеспечением

В 4 семестре для занятий требуются также учебные классы, вмещающие 20-30 человек, оборудованные компьютерами с лицензионным программным обеспечением, с возможностью скачивания с сервера необходимых материалов в электронном виде (описание практических работ, дистрибутивы программ)

11. Критерии оценивания и оценочные средства

11.1. Критерии оценивания

Для дисциплины «Практикум по программированию» формой аттестации является зачёт. 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
- 2. Форма проведения экзамена:** Портфолио, включающее практические работы.
- 3. Правила проведения промежуточной аттестации с применением электронного обучения, дистанционных образовательных технологий**

Портфолио, представляемое на промежуточную аттестацию, включает работы, выполненные в течение семестра и размещенные на сайте курса.

Промежуточная аттестация студентов проводится преподавателями, ведущими практические работы.

Для сдачи работ не позднее чем за 2 дня до даты зачета (по расписанию экзаменационной сессии) обучающийся должен:

- продемонстрировать выполнение проектов,
- загрузить отчеты по работам на сайт курса.

Собеседование по темам работ и получение зачета может выполняться в соответствии с расписанием экзаменационной сессии.

Баллы за работы ставятся преподавателем, ведущим лабораторные занятия.

4. Критерии выставления оценок по результатам промежуточной аттестации по дисциплине

Критерии оценки работ.

Оценка «зачтено» по дисциплине «Практикум по программированию» выставляется в случае успешной сдачи работ, входящих в состав портфолио.

Оценка «зачтено» за лабораторную работу выставляется в случае выполнения следующих условий:

- проект выполнен в соответствии с индивидуальным вариантом задания и представлен вовремя (не позднее чем за 2 дня до даты зачета по расписанию экзаменационной сессии);
- представлен отчет по работе;
- успешно пройдено собеседование по теме работы.

Если, что-то из перечисленного выше не выполнено, по дисциплине «Практикум по программированию» выставляется оценка «не зачтено».

Качество выполненных работ оценивается баллами (5, 4, 3), и может учитываться в экзаменационной оценке по дисциплинам «Алгоритмизация и программирование» и "Алгоритмы и структуры данных".

Максимальная оценка *зачтенной* работы - 5 баллов.ю

Минимальная оценка *зачтенной* работы – 3 балла.

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

Баллы по приведенной шкале в рамках применения СИД (ПСРО+ ПА)	Оценка по результатам промежуточной аттестации
	Экзамен/диф.зачет/зачет
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

Трудности начального этапа изучения программирования в значительной степени определяются большим разбросом начального уровня подготовки. Первая рекомендация - иметь по возможности все необходимые материалы в электронном виде и передавать их студентам и исходить из того, что если даже студент что-то может сделать (решить легкую задачу), то надо всегда убедиться, что он понимает то, что делает. Объяснения должны быть предельно понятными и хорошо проиллюстрированными. На практических занятиях надо требовать, чтобы все заданные на дом задачи выполнялись последовательно и в соответствии с логикой построения курса. Рекомендуется объяснять общие подходы к решению задач, пути решения возникающих трудностей, а также обсуждать интересные решения, предлагаемые способными студентами. Очень важен текущий контроль в семестре.

На этапе третьего семестра, на начальном этапе обучения новому языку, идёт существенный упор на знания, полученных за 2 предыдущих семестра, рекомендуется проверить действительный уровень студентов в необходимых областях. Сгладить переход на новый язык помогут примеры аналогичных конструкций из уже изученного языка. На лекционных занятиях хорошо зарекомендовали себя периодические вопросы к аудитории по только что рассказанному материалу – они способствуют вовлечению студентов в ход лекции. На практических занятиях обязательно задавать дополнительные вопросы по использованным конструкциям, для проверки уровня понимания того, как работает код.

На этапе четвёртого семестра, ввиду отсутствия лекционных занятий, вся теоретическая подготовка переходит на плечи студентов, что неминуемо ведёт к большому числу вопросов, следовательно необходимо обязательно выделять на практических занятиях время для ответов на них и разбор фрагментов студенческого кода. Обязательно демонстрировать с подробными комментариями код, решающий задачи сходные с задачами студентов, а также рассматривать нестандартные подходы найденные обучающимися.

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.