

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Петербургский государственный университет путей сообщения
Императора Александра I»
(ФГБОУ ВО ПГУПС)

Кафедра «Информационные и вычислительные системы»

РАБОЧАЯ ПРОГРАММА

дисциплины

Б1.В.6 «ПРОГРАММИРОВАНИЕ»

для направления подготовки

09.03.01 «Информатика и вычислительная техника»

по профилю

**«Программное обеспечение вычислительной техники
и автоматизированных систем»**

Форма обучения – очная

Санкт-Петербург
2025

ЛИСТ СОГЛАСОВАНИЙ

Рабочая программа рассмотрена и утверждена на заседании кафедры
«Информационные и вычислительные системы»
Протокол №4 от « 23 » декабря 2024 г.

Заведующий кафедрой
«Информационные и вычислительные системы» _____ С.Г. Ермаков
« 23 » декабря 2024 г.

СОГЛАСОВАНО

Руководитель ОПОП ВО _____ С.Г. Ермаков
« 23 » декабря 2024 г.

1. Цели и задачи дисциплины

Рабочая программа дисциплины «Программирование» (Б1.В.6) (далее – дисциплина) составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования – бакалавриат по направлению подготовки 09.03.01 «Информатика и вычислительная техника» (далее – ФГОС ВО), утвержденного 19 сентября 2017г., приказ Минобрнауки России № 917.

Целью изучения дисциплины является формирование у обучающихся компетенций, необходимых для эффективного решения задач в области программирования, включая разработку, тестирование и сопровождение программного обеспечения.

Для достижения поставленной цели решаются следующие задачи:

- изучение синтаксиса и семантики языков программирования;
- изучение основ структурного, объектно-ориентированного и функционального программирования;
- изучение методов и средств проектирования программного обеспечения;
- овладение принципами проектирования архитектуры программных систем;
- освоение методов и средств проектирования реляционных и нереляционных баз данных;
- изучение принципов разработки и согласования технических спецификаций на программные компоненты и их взаимодействие;
- развитие навыков анализа и оценки программных решений;
- изучение современных технологий и инструментов, используемых в разработке программного обеспечения;
- формирование готовности к адаптации в профессиональной среде и применению полученных знаний на практике.

2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с установленными в программе бакалавриата индикаторами достижения компетенций

Планируемыми результатами обучения по дисциплине является формирование у обучающихся компетенций (части компетенций). Сформированность компетенций (части компетенции) оценивается с помощью индикаторов достижения компетенций, приведенных в таблице 2.1.

Таблица 2.1.

Индикаторы достижения компетенций	Результаты обучения по дисциплине
УК-2. Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	
УК-2.1.1 Знает виды ресурсов и ограничений для решения профессиональных задач; основные методы оценки разных способов решения задач; действующее законодательство и правовые нормы, регулирующие профессиональную деятельность.	Обучающийся <i>знает</i> : <ul style="list-style-type: none">– синтаксис и семантику языков программирования;– основы реляционной модели данных и проектирования схемы базы данных

УК-2.2.1 Умеет проводить анализ поставленной цели и формулировать задачи, которые необходимо решить для ее достижения; анализировать альтернативные варианты для достижения намеченных результатов; использовать нормативно-правовую документацию в сфере профессиональной деятельности	Обучающийся <i>умеет</i> : <ul style="list-style-type: none"> – анализировать и выбирать альтернативные варианты реализации; – проводить анализ и оценку программных решений; – использовать современные инструменты разработки, такие как системы контроля версий, CI/CD; – адаптироваться в профессиональной среде и непрерывно обучаться
УК-2.3.1 Владеет методиками разработки цели и задач проекта; методами оценки потребности в ресурсах, продолжительности и стоимости проекта; навыками работы с нормативно-правовой документацией	Обучающийся <i>владеет</i> : <ul style="list-style-type: none"> – методиками разработки цели и задач проекта, UML; – методами оценки потребности в ресурсах; – навыками работы с нормативно-правовой документацией и профессиональными стандартами
ПК-2. Способен разрабатывать технические спецификации на программные компоненты и их взаимодействие	
ПК-2.1.1 Знает методы и средства проектирования программного обеспечения	Обучающийся <i>знает</i> : <ul style="list-style-type: none"> – основные методы и средства, используемые для проектирования программного обеспечения, включая принципы структурного, объектно-ориентированного и функционального программирования
ПК-2.1.2 Знает методы и средства проектирования баз данных	Обучающийся <i>знает</i> : <ul style="list-style-type: none"> – методы и средства проектирования реляционных и нереляционных баз данных
ПК-2.2.1 Умеет вырабатывать варианты реализации программного обеспечения	Обучающийся <i>умеет</i> : <ul style="list-style-type: none"> – разрабатывать различные варианты реализации программного обеспечения, анализируя их достоинства и недостатки; – выбирать наиболее подходящие варианты, исходя из конкретных требований и условий проекта
ПК-2.2.2 Умеет проводить оценку и обоснование рекомендуемых решений	Обучающийся <i>умеет</i> : <ul style="list-style-type: none"> – проводить оценку предложенных решений по различным критериям, включая производительность, устойчивость, стоимость и удобство сопровождения; – обосновывать выбранные решения, предоставляя аргументированные и объективные данные
ПК-2.3.1 Имеет навыки разработки и согласование технических спецификаций на программные компоненты и их взаимодействие с архитектором программного обеспечения	Обучающийся <i>имеет навыки</i> : <ul style="list-style-type: none"> – разработки технических спецификаций для программных компонентов, подробно описывая их функциональность, интерфейсы и взаимодействие; – согласования спецификаций с архитектором программного обеспечения, учитывая требования и ограничения проекта

3. Место дисциплины в структуре основной профессиональной образовательной программы

Дисциплина относится к части, формируемой участниками образовательных отношений блока 1 «Дисциплины (модули)».

4. Объем дисциплины и виды учебной работы

Таблица 4.1.

Вид учебной работы	Всего часов	Модуль	
		1	2
Контактная работа (по видам учебных занятий)	128	64	64
В том числе:			
– лекции (Л)	64	32	32
– лабораторные работы (ЛР)	64	32	32
Самостоятельная работа (СРС)	80	40	40
Контроль	8	4	4
Форма контроля (промежуточной аттестации)	3	3	3
Общая трудоемкость: час / з.е.	216/6	108/3	108/3

Примечание: форма контроля – зачет (3)

5. Структура и содержание дисциплины

5.1. Разделы дисциплины и содержание рассматриваемых вопросов

Таблица 5.1.

№ п/п	Наименование раздела дисциплины	Содержание раздела	Индикаторы достижения компетенций
1 модуль			
1	Основы программирования	<u>Лекция 1.</u> Введение в программирование. История и обзор языков программирования. <u>Лекция 2.</u> Синтаксис и семантика языков программирования. Основные элементы языков программирования: переменные и типы данных. <u>Лекция 3.</u> Операторы и выражения в программировании. Управляющие конструкции: условные операторы, циклы. <u>Лекция 4.</u> Функции и процедуры: объявление и вызов. Параметры функций и область видимости переменных. <u>Лекция 5.</u> Введение в алгоритмы: понятие и классификация. Основные алгоритмы: сортировка и поиск.	УК-2.1.1 ПК-2.1.1

		<p><u>Лекция 6.</u> Структуры данных: массивы и списки. Понятие и типы переменных: глобальные, локальные, статические.</p> <p><u>Лабораторная работа №1.</u> Создание и использование функций. Передача параметров в функции. Область видимости переменных.</p> <p><u>Лабораторная работа №2.</u> Реализация алгоритмов сортировки (bubble sort, selection sort). Реализация алгоритмов поиска (линейный поиск, бинарный поиск). Работа с динамическими структурами данных (списки).</p> <p><u>Самостоятельная работа (14 часов)</u> Изучение, углубление и расширение лекционного материала, дополнение текстов лекций по темам курса; оформление и подготовка к защите лабораторных работ в т.ч. повторение теоретического материала по темам курса с использованием текстов лекций, рекомендованной литературы, дополнительных источников.</p>	
2	Парадигмы программирования	<p><u>Лекция 7.</u> Структурное программирование: принципы и практика.</p> <p><u>Лекция 8.</u> Введение в объектно-ориентированное программирование (ООП).</p> <p><u>Лекция 9.</u> Основные концепции ООП: классы и объекты, наследование, полиморфизм, инкапсуляция.</p> <p><u>Лекция 10.</u> Проектирование и реализация классов и объектов.</p> <p><u>Лекция 11.</u> Основы функционального программирования. Функции высшего порядка и замыкания.</p> <p><u>Лабораторная работа №3.</u> Написание программ с использованием принципов структурного программирования. Разработка программных модулей. Создание классов и объектов. Использование основных концепций ООП (наследование, полиморфизм, инкапсуляция).</p> <p><u>Лабораторная работа №4.</u> Написание программ с использованием функций высшего порядка. Использование замыканий.</p> <p><u>Самостоятельная работа (12 часов)</u></p>	УК-2.2.1 ПК-2.1.1

		Изучение, углубление и расширение лекционного материала, дополнение текстов лекций по темам курса; оформление и подготовка к защите лабораторных работ в т.ч. повторение теоретического материала по темам курса с использованием текстов лекций, рекомендованной литературы, дополнительных источников.	
3	Проектирование программного обеспечения	<p><u>Лекция 12.</u> Основы проектирования программного обеспечения. Применение UML для моделирования программного обеспечения.</p> <p><u>Лекция 13.</u> Проектирование архитектуры программных систем: принципы и шаблоны. Модульность и повторное использование компонентов.</p> <p><u>Лабораторная работа №5.</u> Создание диаграмм классов, последовательностей и состояний. Моделирование простого программного проекта с использованием UML. Разработка и реализация архитектуры программной системы с использованием шаблонов проектирования.</p> <p><u>Самостоятельная работа (6 часов)</u> Изучение, углубление и расширение лекционного материала, дополнение текстов лекций по темам курса; оформление и подготовка к защите лабораторных работ в т.ч. повторение теоретического материала по темам курса с использованием текстов лекций, рекомендованной литературы, дополнительных источников.</p>	УК-2.3.1 ПК-2.1.1
4	Базы данных	<p><u>Лекция 14.</u> Введение в реляционные базы данных. Основы реляционной модели данных. Проектирование схемы базы данных.</p> <p><u>Лекция 15.</u> Основы языка SQL: создание и управление базами данных. Продвинутое запросы в SQL: агрегации, подзапросы, соединения.</p> <p><u>Лекция 16.</u> Введение в нереляционные базы данных. Основные типы нереляционных баз данных: документные, ключ-значение, графовые. Проектирование и использование нереляционных баз данных.</p> <p><u>Лабораторная работа №6.</u> Создание схемы базы данных. Определение связей</p>	УК-2.1.1 ПК-2.1.2

		<p>между таблицами. Заполнение таблиц данными.</p> <p><u>Лабораторная работа №7.</u> Создание таблиц и выполнение основных SQL-запросов. Управление данными (вставка, обновление, удаление). Выполнение сложных SQL-запросов. Использование подзапросов и объединений таблиц.</p> <p><u>Самостоятельная работа (8 часов)</u> Изучение, углубление и расширение лекционного материала, дополнение текстов лекций по темам курса; оформление и подготовка к защите лабораторных работ в т.ч. повторение теоретического материала по темам курса с использованием текстов лекций, рекомендованной литературы, дополнительных источников.</p>	
2 модуль			
5	Технические спецификации и взаимодействие компонентов	<p><u>Лекция 17.</u> Разработка технических спецификаций: принципы и примеры. Согласование спецификаций с архитектором программного обеспечения.</p> <p><u>Лекция 18.</u> Проектирование взаимодействия между программными компонентами. Протоколы и стандарты взаимодействия.</p> <p><u>Лабораторная работа №8.</u> Создание технической спецификации для программного компонента. Согласование спецификаций с архитектором программного обеспечения.</p> <p><u>Самостоятельная работа (6 часов)</u> Изучение, углубление и расширение лекционного материала, дополнение текстов лекций по темам курса; оформление и подготовка к защите лабораторных работ в т.ч. повторение теоретического материала по темам курса с использованием текстов лекций, рекомендованной литературы, дополнительных источников.</p>	<p>УК-2.3.1 ПК-2.3.1</p>
6	Анализ и оценка программных решений	<p><u>Лекция 19.</u> Методы анализа и оценки производительности программного обеспечения. Тестирование программного обеспечения: методы и подходы.</p> <p><u>Лекция 20.</u> Отладка программного обеспечения: инструменты и техники.</p>	<p>УК-2.2.1 ПК-2.2.2</p>

		<p><u>Лекция 21.</u> Анализ альтернативных вариантов реализации задач. Сравнение и оценка различных подходов к решению задач. Обоснование выбора оптимального решения.</p> <p><u>Лабораторная работа №9.</u> Измерение и анализ производительности программного обеспечения. Оптимизация кода для повышения производительности. Использование отладчиков для поиска и устранения ошибок. Применение различных техник отладки</p> <p><u>Самостоятельная работа (8 часов)</u> Изучение, углубление и расширение лекционного материала, дополнение текстов лекций по темам курса; оформление и подготовка к защите лабораторных работ в т.ч. повторение теоретического материала по темам курса с использованием текстов лекций, рекомендованной литературы, дополнительных источников.</p>	
7	Современные технологии и инструменты разработки ПО	<p><u>Лекция 22.</u> Среды разработки (IDE): обзор и использование.</p> <p><u>Лекция 23.</u> Системы контроля версий: основы работы с Git.</p> <p><u>Лекция 24.</u> Инструменты CI/CD: настройка и использование.</p> <p><u>Лекция 25.</u> Введение в веб-разработку: основные технологии и инструменты.</p> <p><u>Лекция 26.</u> Основы мобильной разработки: платформы и инструменты.</p> <p><u>Лекция 27.</u> Разработка облачных решений: принципы и сервисы.</p> <p><u>Лекция 28.</u> Основы DevOps: концепции и инструменты.</p> <p><u>Лекция 29.</u> Введение в разработку игр: движки и основные концепции.</p> <p><u>Лабораторная работа №10.</u> Настройка и использование различных IDE. Системы контроля версий: основы работы с Git. Настройка репозитория Git. Выполнение основных операций с Git (commit, push, pull, merge).</p> <p><u>Лабораторная работа №11.</u> Создание простого веб-приложения. Использование HTML, CSS и JavaScript.</p> <p><u>Лабораторная работа №12.</u> Разработка простого мобильного приложения.</p>	УК-2.2.1 ПК-2.1.1

		<p>Использование инструментов для Android или iOS</p> <p>Лабораторная работа №13. Создание простого игрового проекта. Использование игрового движка (например, PyGame).</p> <p>Самостоятельная работа (18 часов) Изучение, углубление и расширение лекционного материала, дополнение текстов лекций по темам курса; оформление и подготовка к защите лабораторных работ в т.ч. повторение теоретического материала по темам курса с использованием текстов лекций, рекомендованной литературы, дополнительных источников.</p>	
8	Профессиональная адаптация и практика	<p>Лекция 30. Понимание и применение профессиональных стандартов и практик.</p> <p>Лекция 31. Адаптация в профессиональной среде: рекомендации и стратегии.</p> <p>Лекция 32. Непрерывное обучение и профессиональное развитие.</p> <p>Лабораторная работа №14. Непрерывное обучение и профессиональное развитие. Создание плана непрерывного профессионального развития.</p> <p>Самостоятельная работа (8 часов) Изучение, углубление и расширение лекционного материала, дополнение текстов лекций по темам курса; оформление и подготовка к защите лабораторных работ в т.ч. повторение теоретического материала по темам курса с использованием текстов лекций, рекомендованной литературы, дополнительных источников.</p>	<p>УК-2.2.1 УК-2.3.1</p>

5.2. Разделы дисциплины и виды занятий

Таблица 5.2.

№ п/п	Наименование раздела дисциплины	Л	ЛР	СРС	Всего
1 модуль					
1	Основы программирования	12	12	14	38
2	Парадигмы программирования	10	8	12	30
3	Проектирование программного обеспечения	4	4	6	14
4	Базы данных	6	8	8	22

	Итого	32	32	40	104
Контроль					4
Всего (общая трудоемкость, час)					108
2 модуль					
5	Технические спецификации и взаимодействие компонентов	4	4	6	14
6	Анализ и оценка программных решений	6	6	8	20
7	Современные технологии и инструменты разработки ПО	16	20	18	54
8	Профессиональная адаптация и практика	6	2	8	16
	Итого	32	32	40	104
Контроль					4
Всего (общая трудоемкость, час)					108

6. Оценочные материалы для проведения текущего контроля успеваемости и промежуточной аттестации обучающихся по дисциплине

Оценочные материалы по дисциплине являются неотъемлемой частью рабочей программы и представлены отдельным документом, рассмотренным на заседании кафедры и утвержденным заведующим кафедрой.

7. Методические указания для обучающихся по освоению дисциплины

Порядок изучения дисциплины, следующий:

1. Освоение разделов дисциплины производится в порядке, приведенном в разделе 5 «Содержание и структура дисциплины». Обучающийся должен освоить все разделы дисциплины, используя методические материалы дисциплины, а также учебно-методическое обеспечение, приведенное в разделе 8 рабочей программы.

2. Для формирования компетенций обучающийся должен представить выполненные типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, предусмотренные текущим контролем успеваемости (см. оценочные материалы по дисциплине).

3. По итогам текущего контроля успеваемости по дисциплине, обучающийся должен пройти промежуточную аттестацию (см. оценочные материалы по дисциплине).

8. Описание материально-технического и учебно-методического обеспечения, необходимого для реализации программы магистратуры по дисциплине

8.1. Помещения представляют собой учебные аудитории для проведения учебных занятий, предусмотренных программой магистратуры, укомплектованные специализированной учебной мебелью и оснащенные оборудованием и техническими средствами обучения, служащими для представления учебной информации большой аудитории: стационарным или переносным экраном, маркерной доской и (или) меловой доской, стационарным или переносным мультимедийным проектором, персональными компьютерами.

Все помещения, используемые для проведения учебных занятий и самостоятельной работы, соответствуют действующим санитарным и противопожарным нормам и правилам.

Для проведения лабораторных работ используются дисплейные классы кафедры «Информационные и вычислительные системы». Классы оборудованы персональными компьютерами, включая сервер локальной сети для размещения методических материалов и результатов выполнения лабораторных работ. На компьютерах установлен комплект необходимого программного обеспечения, приведенного в п. 8.2.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета.

8.2. Университет обеспечен необходимым комплектом лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства:

- операционная система Windows, MS Office, Антивирус Касперского;
- среда программирования IDE: PyCharm (Python) (бесплатное, свободно распространяемое программное обеспечение; режим доступа <https://python.org/downloads/>)
- NetBeans IDE 8.2 (бесплатное, свободно распространяемое программное обеспечение; режим доступа <https://netbeans.org/downloads/>)
- StarUML V 6.2.2 (бесплатное, свободно распространяемое программное обеспечение; режим доступа <https://staruml.softonic.ru/>)

8.3. Обучающимся обеспечен доступ (удаленный доступ) к современным профессиональным базам данных:

- Научная электронная библиотека [eLIBRARY.RU](http://elibrary.ru) [Электронный ресурс] – Режим доступа: <https://www.elibrary.ru> (свободный доступ);
- Электронно-библиотечная система «Лань» [Электронный ресурс] – Режим доступа: <https://e.lanbook.com> (свободный доступ);
- Электронная библиотечная система «Юрайт» [Электронный ресурс]– Режим доступа: <https://urait.ru> (свободный доступ).

8.4. Обучающимся обеспечен доступ (удаленный доступ) к информационным справочным системам:

8.5. Перечень печатных и электронных изданий, используемых в образовательном процессе:

1. Прохоренок Н. А. Python 3 и PyQt5. Разработка приложений. – 2-е изд., перераб. и доп. /Н.А. Прохоренок, В.А. Дронов. – СПб.: БХВ-Петербург, 2019. – 832с.
2. Вандерплас Дж. Python для сложных задач.: Издательский дом «Питер», 2019, 572 с.
3. Орлов С. А. Теория и практика языков программирования. – 2-е изд. – СПб.: Издательский дом «Питер», 2017г. – 688с.

8.6. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», используемых в образовательном процессе:

1. Личный кабинет обучающегося и электронная информационно-образовательная среда [Электронный ресурс]. – Режим доступа: <http://sdo.pgups.ru/> (для доступа к полнотекстовым документам требуется авторизация);
2. Научно-техническая библиотека университета [Электронный ресурс]. – Режим доступа: <http://library.pgups.ru/> (свободный доступ).

Разработчик рабочей программы,
доцент

В.Е. Петров

« 23 » декабря 2024 г.