

ЕЛЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ им. И.А. БУНИНА

«УТВЕРЖДАЮ»

Директор института экономики,
управления и сервисных технологий



Н.В. Осипова

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Б1.О.04.09 Языки и методы программирования

Направление подготовки: 38.03.01 Экономика с присвоением второй квалификации 09.03.02 Информационные системы и технологии

Направленность (профиль): Прикладная экономика и большие данные

Квалификация (степень): *бакалавр*

Форма обучения: *очная*

Институт: экономики, управления и сервисных технологий

Кафедра: математического моделирования, компьютерных технологий и информационной безопасности

	очная форма	очно-заочная форма	заочная форма
Курс	1		
Семестр/триместр	1,2		

Лекции	54		
Лабораторные занятия			
Практические (семинарские) занятия	90		
в т. ч. практическая подготовка			
Форма(ы) промежуточной аттестации	Экзамен — 0,3 Экзамен — 0,3		
Контроль	18		
Самостоятельная работа	53,4		

Всего часов: 216

Трудоемкость: 6 зачетных единиц.

Разработчик(и) рабочей программы: ст. преподаватель Самсонов И.Ю.

I. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЙ РАЗДЕЛ

Цель изучения дисциплины:

формирование у студентов фундаментальных знаний о структуре, классификации и принципах функционирования языков программирования, а также развитие практических навыков применения различных парадигм программирования (императивной, объектно-ориентированной, функциональной, логической и др.) для эффективного решения задач в области информационных технологий.

Задачи изучения дисциплины:

- ознакомить с классификацией языков программирования, их синтаксисом, семантикой и прагматикой, а также с историей развития и тенденциями современного программирования;
- развить навыки анализа и выбора языка программирования и метода реализации в зависимости от специфики решаемой задачи;
- освоить методы структурного и модульного программирования, принципы проектирования программ с учетом требований к надежности, читаемости и сопровождаемости кода;
- получить представления о роли и месте программирования в обществе.

Место дисциплины в структуре ОПОП: реализуется в части, формируемой участниками образовательных отношений блока Б1. Дисциплины (модули).

Планируемые результаты обучения по дисциплине:

Код компетенции	Индикаторы достижения компетенции	Планируемые результаты обучения по дисциплине
ОПК-6 (К1)	Знать: <ul style="list-style-type: none">• основные принципы функционирования современных информационных технологий (ИТ), включая аппаратное и программное обеспечение, сети, базы данных, облачные технологии и системы обработки данных;• архитектуру и компоненты информационных систем, используемых в профессиональной сфере.	Знает: <ul style="list-style-type: none">• основные понятия, принципы и архитектурные модели современных информационных технологий и систем;• методы и средства разработки, тестирования и сопровождения программного обеспечения;• принципы функционирования компьютерных сетей, баз данных, облачных платформ и веб-технологий.
	Уметь: <ul style="list-style-type: none">• анализировать задачи профессиональной деятельности с точки зрения применимости современных ИТ-решений;• выбирать и адаптировать подходящие информационные технологии для решения конкретных прикладных	Умеет: <ul style="list-style-type: none">• анализировать предметную область и формулировать требования к информационной системе;• применять современные инструменты и технологии для разработки программных

	<p>задач;</p> <ul style="list-style-type: none"> использовать ИТ-инструменты и платформы в ходе проектной, аналитической и разработческой деятельности. 	<p>решений;</p> <ul style="list-style-type: none"> использовать системы контроля версий, среды разработки, тестовые фреймворки и другие средства автоматизации.
	<p>Владеть:</p> <ul style="list-style-type: none"> навыками применения современных информационных технологий для обработки данных, автоматизации процессов и реализации программных решений; методами интеграции различных ИТ-компонентов в единую систему для достижения поставленных целей; средствами оценки эффективности и безопасности используемых ИТ-решений. 	<p>Владеет:</p> <ul style="list-style-type: none"> навыками проектирования и реализации программных модулей с использованием актуальных технологий и языков программирования; методами эффективной коммуникации в профессиональной среде (в том числе технической документации, презентаций, отчётов); способами самостоятельного освоения новых технологий и адаптации к меняющейся ИТ-среде.
<p>ОПК-3 (К2)</p>	<p>Знать:</p> <ul style="list-style-type: none"> основы информационной и библиографической культуры: принципы поиска, оценки, систематизации и использования научной и профессиональной информации; базовые требования и нормы информационной безопасности (защита данных, авторские права, цифровая идентичность, безопасность в интернете); современные информационно-коммуникационные технологии (ИКТ), применяемые в профессиональной деятельности. 	<p>Знает:</p> <ul style="list-style-type: none"> принципы информационной грамотности, источники научной информации, основы ИКТ и информационной безопасности.
	<p>Уметь:</p> <ul style="list-style-type: none"> эффективно искать, анализировать, критически оценивать и использовать информацию из различных источников для решения профессиональных задач; применять ИКТ-средства (текстовые редакторы, системы презентаций, облачные сервисы, системы управления проектами и др.) в учебной и профессиональной деятельности оформлять научные и технические документы в соответствии с установленными стандартами. 	<p>Умеет:</p> <ul style="list-style-type: none"> эффективно искать, обрабатывать и использовать информацию, применять цифровые инструменты для решения профессиональных задач с соблюдением этических и правовых норм.

	<p>Владеть:</p> <ul style="list-style-type: none"> • навыками работы с электронными библиотеками, базами данных, поисковыми системами и цифровыми образовательными ресурсами; • практическими приёмами безопасного поведения в цифровой среде, включая защиту персональных и служебных данных; • методами критического анализа достоверности и релевантности информации. 	<p>Владеет:</p> <ul style="list-style-type: none"> • навыками информационно-поисковой деятельности, работы с цифровыми платформами и обеспечения информационной безопасности в профессиональной среде.
ОПК-5 (К2)	<p>Знать:</p> <ul style="list-style-type: none"> • основные типы программного обеспечения; • архитектуру и компоненты аппаратного обеспечения; • требования совместимости между аппаратным и программным обеспечением. 	<p>Знает:</p> <ul style="list-style-type: none"> • принципы совместимости и функционирования аппаратно-программного комплекса, этапы установки и настройки ПО и оборудования.
	<p>Уметь:</p> <ul style="list-style-type: none"> • производить установку и настройку операционных систем и прикладного программного обеспечения; • проверять работоспособность установленного программного и аппаратного обеспечения. 	<p>Умеет:</p> <ul style="list-style-type: none"> • самостоятельно устанавливать, настраивать и тестировать программное и аппаратное обеспечение информационных систем.
	<p>Владеть:</p> <ul style="list-style-type: none"> • навыками безопасной и корректной установки программного обеспечения с проверкой источников и лицензий; • инструментами диагностики и тестирования оборудования (программные и аппаратные утилиты). 	<p>Владеет:</p> <ul style="list-style-type: none"> • практическими навыками сборки, диагностики и инсталляции компонентов компьютерных систем, необходимыми для решения стандартных задач технической поддержки и сопровождения.
ОПК-6 (К2)	<p>Знать:</p> <ul style="list-style-type: none"> • основные принципы алгоритмизации и программирования; • типы алгоритмов; • базовые структуры данных; • современные языки программирования и среды разработки, применяемые в информационных системах. 	<p>Знает:</p> <ul style="list-style-type: none"> • основы алгоритмизации, принципы построения программ и требования к их качеству в условиях практического применения.
	<p>Уметь:</p> <ul style="list-style-type: none"> • разрабатывать алгоритмы для решения прикладных задач в области информационных технологий; • применять алгоритмы и программы для обработки данных, автоматизации 	<p>Умеет:</p> <ul style="list-style-type: none"> • разрабатывать, реализовывать и тестировать алгоритмы и программы, ориентированные на решение задач в сфере информационных систем и

<p>процессов и интеграции в информационные системы;</p> <ul style="list-style-type: none"> документировать код и разрабатывать техническую документацию к программным модулям. 	технологий.
<p>Владеть:</p> <ul style="list-style-type: none"> навыками проектирования и реализации эффективных, надёжных и масштабируемых программных решений; методами анализа сложности алгоритмов (O-нотация) и выбора оптимальных структур данных; способами адаптации программных решений к реальным условиям эксплуатации в информационных системах. 	<p>Владеет:</p> <ul style="list-style-type: none"> практическими навыками программирования, анализа алгоритмов и создания программных решений, пригодных для внедрения в реальные ИТ-среды.

II. СОДЕРЖАНИЕ И ОБЪЕМ ДИСЦИПЛИНЫ

с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу

Очная форма обучения

№ п/п	Наименование разделов и тем	Всего	Аудиторные занятия			Сам. раб.
			ЛК	ПЗ	ЛБ	
Раздел 1. Введение в языки программирования.		14,2	6	6	0	2,2
1.	Тема 1. Понятие языка программирования. Классификация: машинные, ассемблеры, языки высокого уровня.	4,5	2	2	0	0,5
2.	Тема 2. Эволюция языков программирования. Поколения языков.	4,5	2	2	0	0,5
3.	Тема 3. Трансляция программ: компиляция и интерпретация.	4,5	2	2	0	0,5
4.	Тема 4. Обзор современных языков и областей применения.	0,7	0	0	0	0,7
Раздел 2. Основы синтаксиса и семантики.		14,5	4	8	0	2,5
5.	Тема 1. Лексический анализ. Служебные слова, идентификаторы, литералы.	4,5	2	2	0	0,5
6.	Тема 2. Формальные грамматики: БНФ, EBNF.	7	2	4	0	1
7.	Тема 3. Синтаксические диаграммы. Деревья разбора.	2,5	0	2	0	0,5
8.	Тема 4. Семантика: статическая и динамическая.	0,5	0	0	0	0,5
Раздел 3. Императивное и структурное программирование.		14	4	8	0	2
9.	Тема 1. Основные конструкции: присваивание, ветвление, циклы.	4,5	2	2	0	0,5
10.	Тема 2. Структурное программирование. Теорема о структурировании.	6,5	2	4	0	0,5

11.	Тема 3. Подпрограммы: функции, параметры, область видимости.	2,5	0	2	0	0,5
12.	Тема 4. Модульность. Принципы проектирования.	0,5	0	0	0	0,5
Раздел 4. Методы программирования.		12	4	6	0	2
13.	Тема 1. Проектирование "сверху вниз" и "снизу вверх".	4,5	2	2	0	0,5
14.	Тема 2. Обработка ошибок и исключений.	4,5	2	2	0	0,5
15.	Тема 3. Реиспользование кода: библиотеки, паттерны.	2,5	0	2	0	0,5
16.	Тема 4. Документирование и стиль кода.	0,5	0	0	0	0,5
Раздел 5. Объектно-ориентированное программирование.		21	8	10	0	3
17.	Тема 1. Основные понятия ООП: класс, объект, экземпляр.	4,5	2	2	0	0,5
18.	Тема 2. Инкапсуляция и сокрытие данных.	4,5	2	2	0	0,5
19.	Тема 3. Наследование и полиморфизм.	4,5	2	2	0	0,5
20.	Тема 4. Абстрактные классы, интерфейсы.	4,5	2	2	0	0,5
21.	Тема 5. UML-диаграммы классов.	3	0	2	0	1
Раздел 6. Функциональное программирование.		17	6	8	0	3
22.	Тема 1. Основные концепции: чистые функции, неизменяемость.	4,5	2	2	0	0,5
23.	Тема 2. Функции высшего порядка, лямбды.	4,5	2	2	0	0,5
24.	Тема 3. Языки: Haskell, Lisp, Scala.	4,5	2	2	0	1
25.	Тема 4. Применение в Python/JavaScript.	3	0	2	0	1
Раздел 7. Логическое программирование.		14	4	8	0	2
26.	Тема 1. Основы логического программирования.	4,5	2	2	0	0,5
27.	Тема 2. Пролог: факты, правила, запросы.	4,5	2	2	0	0,5
28.	Тема 3. Унификация и поиск с возвратом.	2,5	0	2	0	0,5
29.	Тема 4. Примеры: экспертные системы, головоломки.	2,5	0	2	0	0,5
	<i>В т.ч. с практической подготовкой</i>			0	0	
	<i>Экзамен</i>	0,3				
	<i>Контроль</i>	9				
	<i>Итого за 1 семестр</i>	98,7	36	54	0	8,7
Раздел 8. Современные языки программирования.		14	4	6	0	4
30.	Тема 1. Сравнение Python, Java, C++, JavaScript, Rust, Go.	5	2	2	0	1
27.	Тема 2. Системы типизации: статическая, динамическая, вывод типов.	5	2	2	0	1
28.	Тема 3. Управление памятью: GC, RAII, владение (Rust).	3	0	2	0	1
29.	Тема 4. Языки для веб, мобильной и системной разработки.	1	0	0	0	1
Раздел 9. Параллельное и асинхронное программирование.		16	6	6	0	4
30.	Тема 1. Понятия параллелизма и асинхронности.	5	2	2	0	1
27.	Тема 2. Поток, процессы, синхронизация.	5	2	2	0	1
28.	Тема 3. Асинхронные вызовы: async/await, промисы.	3	2	2	0	1
29.	Тема 4. Примеры на Python, Java, JS.	1	0	0	0	1
Раздел 10. Реализация языков программирования.		12	4	4	0	4
30.	Тема 1. Этапы компиляции: лексический, синтаксический, семантический анализ.	5	2	2	0	1
27.	Тема 2. Интерпретаторы и виртуальные машины (JVM, CLR).	5	2	2	0	1
28.	Тема 3. JIT-компиляция.	3	0	0	0	1

29.	Тема 4. Языки домена (DSL).	1	0	0	0	1
Раздел 11. Среды разработки и инструменты.		12	0	8	0	4
30.	Тема 1. IDE: отладка, рефакторинг, профилирование.	3	0	2	0	1
27.	Тема 2. Git: основные команды, ветвление, workflow.	5	0	4	0	1
28.	Тема 3. Тестирование: unit-тесты, интеграционные тесты.	3	0	2	0	1
29.	Тема 4. CI/CD: непрерывная интеграция и доставка.	1	0	0	0	1
Раздел 12. Мультипарадигменное программирование.		10	0	6	0	4
30.	Тема 1. Комбинирование парадигм в одной задаче.	3	0	2	0	1
27.	Тема 2. Проектирование гибридных решений.	3	0	2	0	1
28.	Тема 3. Выбор языка и парадигмы под задачу.	3	0	2	0	2
Раздел 13. Современные тенденции.		15,7	0	8	0	7,7
30.	Тема 1. Low-code / no-code платформы.	3	0	2	0	1
27.	Тема 2. Квантовые языки: Q#, Qiskit.	3	0	2	0	1
28.	Тема 3. Безопасность языков программирования.	3	0	2	0	1
29.	Тема 4. ИИ в программировании: Copilot, генерация кода.	6,7	0	2	0	4,7
	<i>В т.ч. с практической подготовкой</i>			0	0	
	<i>Экзамен</i>	0,3				
	<i>Контроль</i>	9				
	<i>Итого за 2 семестр</i>	98,7	18	36	0	44,7
	Итого:	216	54	90	0	53,4

Очно-заочная форма обучения (не реализуется)

Заочная форма обучения (не реализуется)

III. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕЙ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

Текущая аттестация проводится в форме тестирования, реферата.

Типовой вариант тестового задания.

1. По уровню абстракции языки программирования разделяются на (выберите несколько правильных ответов):

1. параллельные
2. машинные
3. машинно-ориентированные
4. машинно-независимые (языки высокого уровня)
5. цифровые
6. арифметические

2. Языки низкого уровня характеризуются (выберите несколько правильных ответов):

1. отсутствием понятия типа данных

2. зависимостью от архитектуры конкретного компьютера
3. примитивными управляющими структурами и средствами описания структур данных.
4. наличием понятия типа данных
5. независимостью от архитектуры конкретного компьютера
6. развитыми управляющими структурами и средствами описания структур данных.

3. Среда программирования — это:

1. система обозначений, используемая для представления в более удобной форме программ, записанных в машинном коде
2. совокупность команд, которая отличается количеством адресов в команде, назначением информации, задаваемой в адресах, набором операций, которые может выполнить машина
3. система для разработки новых программ на конкретном языке программирования
4. совокупность идей и понятий, определяющих стиль написания компьютерных программ

4. Трансляторы программ разделяются на (выберите несколько вариантов ответа):

1. компиляторы
2. интерпретаторы
3. симуляторы
4. эмуляторы
5. роботы

Типовые темы рефератов.

Темы рефератов для 1 семестра.

1. Эволюция языков программирования: от машинных кодов до C++.
2. Особенности языка C++ как мультипарадигменного языка.
3. Сравнение компиляции и интерпретации на примере C++ и Python.
4. Базовые конструкции структурного программирования в C++.
5. Роль БНФ в описании синтаксиса языков программирования.
6. Принципы проектирования "сверху вниз" на примере решения задачи на C++.
7. Модульность программ: использование заголовочных файлов в C++.
8. Обработка ошибок в C++: механизмы try-catch и исключения.
9. Классы и объекты в C++: основы объектно-ориентированного программирования.
10. Инкапсуляция и управление доступом в C++ (public, private, protected).
11. Наследование в C++: виды и примеры реализации.

- 12.Полиморфизм в C++: виртуальные функции и таблицы виртуальных методов.
- 13.Управление памятью в C++: new/delete и потенциальные утечки памяти.
- 14.Конструкторы и деструкторы: их роль в жизненном цикле объекта.
- 15.Перегрузка функций и операций в C++.
- 16.Шаблоны функций и классов в C++.
- 17.Алгоритмы обработки массивов: сортировка и поиск в C++.
- 18.Реализация стека и очереди на языке C++.
- 19.Работа с файлами в C++: текстовые и бинарные потоки.
- 20.Использование стандартной библиотеки (STL) в C++: контейнеры и итераторы.

Темы рефератов для 2 семестра.

1. Сравнение языков C++, Java и Python: производительность, безопасность, применение.
2. Управление памятью: сборка мусора (GC) vs система владения в Rust.
3. Параллельное программирование в C++: потоки и синхронизация.
4. Асинхронное программирование в Python: asyncio и async/await.
5. Архитектура JVM: как работает виртуальная машина Java.
6. JIT-компиляция: принципы и применение в .NET и Java.
7. Средства отладки в современных IDE: сравнение Visual Studio, CLion, IntelliJ IDEA.
8. Git как инструмент командной разработки: ветвление, merge, pull request.
9. CI/CD в разработке ПО: примеры на GitHub Actions или GitLab CI.
10. Unit-тестирование в C++: использование фреймворков (Google Test, Catch2).
11. Функциональное программирование в Python: map, filter, lambda, functools.
12. Логическое программирование: сравнение Пролога и современных логических DSL.
13. Язык Rust: безопасность, производительность и будущее системного программирования.
14. Low-code платформы: преимущества и ограничения (на примере OutSystems, Mendix).
15. Квантовое программирование: основы Q# и Qiskit.
16. Применение GitHub Copilot и других ИИ-помощников в разработке кода.
17. Мультипарадигменный подход: комбинирование ООП, функционального и императивного стилей.
18. DSL (Domain-Specific Languages): примеры и применение в промышленной разработке.
19. Архитектура компилятора: от исходного кода до машинных инструкций.
20. Современные тенденции в языках программирования: безопасность, производительность, простота.

Промежуточная аттестация обучающихся осуществляется в форме экзамена с использованием следующих оценочных материалов: вопросы к экзамену.

**Вопросы к экзамену
(1 семестр, очная форма обучения)**

1. Что такое язык программирования? Приведите примеры языков низкого и высокого уровня.
2. Чем отличается компилятор от интерпретатора?
3. Какие поколения языков программирования вы знаете? Приведите примеры.
4. Что такое формальная грамматика? Как записывается БНФ?
5. Опишите этапы трансляции программы.
6. Что такое лексический анализ? Какие единицы он выделяет?
7. Что такое синтаксическое дерево? Приведите пример.
8. В чём разница между синтаксисом и семантикой языка?
9. Какие основные управляющие конструкции есть в императивных языках?
10. Сформулируйте теорему о структурировании.
11. Что такое подпрограмма? Чем отличаются функции и процедуры?
12. Что такое область видимости переменных? Приведите пример.
13. Что такое модульность? Какие преимущества она даёт?
14. Что такое метод проектирования "сверху вниз"?
15. Как обрабатываются ошибки и исключения в языках программирования?
16. Что такое стиль программирования? Почему он важен?
17. Что такое ООП? Назовите его основные принципы.
18. Объясните понятия класса и объекта. В чём их разница?
19. Что такое инкапсуляция? Как она реализуется в языках программирования?
20. Что такое наследование? Приведите пример.
21. Что такое полиморфизм? Какие виды полиморфизма вы знаете?
22. Что такое абстрактный класс и интерфейс? В чём разница?
23. Что такое виртуальные методы?
24. Что такое UML-диаграмма классов? Какие элементы она содержит?
25. Что такое функциональное программирование? Назовите его ключевые черты.
26. Что такое чистая функция? Приведите пример.
27. Что такое неизменяемость данных? Почему она важна в функциональном программировании?
28. Что такое рекурсия? Приведите пример вычисления факториала.
29. Что такое логическое программирование? Какой язык ему соответствует?
30. Как устроен запрос в Прологе? Приведите пример правила и факта.

**Вопросы к экзамену
(2 семестр, очная форма обучения)**

1. Сравните языки Python, Java и C++ по типизации, производительности и применению.
2. В чём разница между статической и динамической типизацией? Приведите примеры.

3. Что такое сильная и слабая типизация? Приведите примеры языков.
4. Как устроена система управления памятью в Java?
5. Что такое сборка мусора (garbage collection)? В каких языках она используется?
6. Как работает система владения в языке Rust?
7. Что такое RAII? В каких языках применяется?
8. В чём разница между параллелизмом и асинхронностью?
9. Что такое поток (thread)? Какие проблемы могут возникнуть при работе с потоками?
10. Что такое блокировки (mutex) и гонки данных (race condition)?
11. Что такое async/await? В каких языках используется?
12. Что такое промисы (promises) и futures?
13. Какие этапы включает архитектура компилятора?
14. Что такое промежуточный код?
15. Что такое виртуальная машина? Приведите пример (JVM, CLR).
16. Что такое JIT-компиляция? В каких средах она используется?
17. Что такое DSL (язык предметной области)? Приведите пример.
18. Какие основные функции современных IDE?
19. Что такое отладка? Какие виды отладки вы знаете?
20. Что такое система контроля версий? Зачем нужен Git?
21. Объясните понятия репозиторий, ветка, коммит, merge.
22. Что такое CI/CD? Какие этапы включает непрерывная интеграция?
23. Что такое unit-тест? Зачем он нужен?
24. Что такое интеграционный тест?
25. Какие принципы лежат в основе мультипарадигменного программирования?
26. Приведите пример задачи, где целесообразно использовать несколько парадигм.
27. Что такое low-code и no-code платформы? Приведите примеры.
28. Какие языки используются в квантовом программировании? Кратко опишите Q# или Qiskit.
29. Как ИИ (например, GitHub Copilot) помогает в написании кода?
30. Какие современные тенденции в языках программирования вы считаете наиболее перспективными?

IV. ПЕРЕЧЕНЬ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

4.1. Основная литература

1. Огнева, М. В. Программирование на языке C++: практический курс: учебное пособие для вузов / М. В. Огнева, Е. В. Кудрина. — Москва: Издательство Юрайт, 2022. — 335 с. — (Высшее образование). — ISBN 978-5-534-05123-0. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/492984> (дата обращения: 18.04.2025).

2. Федоров, Д. Ю. Программирование на языке высокого уровня Python: учебное пособие для вузов / Д. Ю. Федоров. — 4-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2022. — 214 с. — (Высшее образование). — ISBN 978-5-534-15733-8. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/509562> (дата обращения: 18.04.2025).

4.2. Дополнительная литература

1. Зыков, С. В. Программирование: учебник и практикум для вузов / С. В. Зыков. — Москва: Издательство Юрайт, 2022. — 320 с. — (Высшее образование). — ISBN 978-5-534-02444-9. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/489754> (дата обращения: 01.01.2025).

2. Зыков, С. В. Программирование. Объектно-ориентированный подход: учебник и практикум для вузов / С. В. Зыков. — Москва: Издательство Юрайт, 2022. — 155 с. — (Высшее образование). — ISBN 978-5-534-00850-0. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/490423> (дата обращения: 18.04.2025).

3. Зыков, С. В. Программирование. Функциональный подход: учебник и практикум для вузов / С. В. Зыков. — Москва: Издательство Юрайт, 2022. — 164 с. — (Высшее образование). — ISBN 978-5-534-00844-9. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/490870> (дата обращения: 18.04.2025).

У. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

№ пп	Ссылка на информационный ресурс	Наименование разработки в электронной форме	Доступность
1.	https://infourok.ru/	Инфоурок: образовательный интернет-проект России. Включает: конспекты уроков, презентации, тесты, видеоуроки и другие материалы по предметам школьной программы.	Свободный доступ
2.	http://edu.ru/	Российское образование: Федеральный портал. Включает ссылки на порталы и сайты образовательных учреждений; государственные образовательные стандарты; нормативные документы; каталог экскурсий и обучающих программ.	Свободный доступ
3.	https://docs.microsoft.com/ru-ru/learn/	Виртуальная академия Microsoft	Свободный доступ
4.	http://www.ict.edu.ru	Федеральный образовательный портал "Информационно-	Свободный доступ

VI. СОВРЕМЕННЫЕ ПРОФЕССИОНАЛЬНЫЕ БАЗЫ ДАННЫХ И ИНФОРМАЦИОННЫЕ СПРАВОЧНЫЕ СИСТЕМЫ

1	http://www.biblioclub.ru	Электронно-библиотечная система (ЭБС) Университетская библиотека онлайн	Регистрация через любой университетский компьютер. В дальнейшем предоставляется неограниченный индивидуальный доступ из любой точки, в которой имеется доступ к сети Интернет
2.	https://urait.ru/	Образовательная платформа Юрайт — образовательный ресурс, электронная библиотека и интернет-магазин, где читают и покупают электронные и печатные учебники авторов — преподавателей ведущих университетов для всех уровней профессионального образования, а также пользуются видео- и аудиоматериалами, тестированием и сервисами для преподавателей, доступными 24 часа 7 дней в неделю.	Регистрация через любой университетский компьютер. В дальнейшем предоставляется неограниченный индивидуальный доступ из любой точки, в которой имеется доступ к сети Интернет
3	www.garant.ru	Информационно-правовой портал	Свободный доступ
4	www.elibrary.ru	Российский информационный портал в области науки, технологии, медицины и образования	Свободный доступ
5	www.consultant.ru	Российская компьютерная справочно-правовая система	Свободный доступ

VII. ЛИЦЕНЗИОННОЕ И СВОБОДНО РАСПРОСТРАНЯЕМОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

При реализации учебной дисциплины применяется следующее лицензионное и свободно распространяемое программное обеспечение:

- Microsoft Windows;
- Microsoft Office;
- Oracle VirtualBox;
- Microsoft Visual Studio Code
- Libre Office и др.

VIII. ОБОРУДОВАНИЕ И ТЕХНИЧЕСКИЕ СРЕДСТВА ОБУЧЕНИЯ, НЕОБХОДИМЫЕ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

Учебные занятия проводятся в аудиториях, укомплектованных специализированной мебелью, в том числе стационарными или переносными техническими средствами обучения (проектор, экран, компьютер/ноутбук).

Лабораторные занятия, групповые и индивидуальные консультации, текущая и промежуточная аттестации проводятся в специализированных компьютерных классах. Перечень основного оборудования: автоматизированные рабочие места с компьютерами, программное обеспечение общего и профессионального назначения.

Самостоятельная работа проводится в кабинетах, оснащенных компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета.