

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ



Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный гуманитарный университет»
(ФГАОУ ВО «РГГУ»)

ИНСТИТУТ ИНФОРМАЦИОННЫХ НАУК И ТЕХНОЛОГИЙ БЕЗОПАСНОСТИ
ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ СИСТЕМ И БЕЗОПАСНОСТИ

Кафедра информационных технологий и систем

ПРОГРАММИРОВАНИЕ
РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

09.03.03 «Прикладная информатика»

Код и наименование направления подготовки

Прикладной искусственный интеллект

Наименование направленности (профиля)

Уровень высшего образования: *бакалавриат*

Форма обучения: *очная*

РПД адаптирована для лиц
с ограниченными возможностями
здравья и инвалидов

Москва 2025

ПРОГРАММИРОВАНИЕ

Рабочая программа дисциплины

Составитель:

к.с.-х.н., доцент, заведующая кафедрой Информационных технологий и систем

Н.Ш. Шуkenбаева

канд.филол.наук, доцент А.В. Муромцева

УТВЕРЖДЕНО:

Протокол заседания кафедры

№ 6 от 12.12. 2024 года

ОГЛАВЛЕНИЕ

1	Пояснительная записка.....	4
1.1	Цель и задачи дисциплины	4
1.2.	Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций.....	4
1.2	Место дисциплины в структуре образовательной программы.....	5
2	Структура дисциплины.....	5
3	Содержание дисциплины	5
4	Образовательные технологии	8
5	Оценка планируемых результатов обучения.....	8
5.1	Система оценивания.....	8
5.2	Критерии выставления оценки по дисциплине.....	9
5.3	Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине	11
6	Учебно-методическое и информационное обеспечение дисциплины.....	14
6.1	Список литературы.....	14
6.2	Перечень ресурсов информационно-коммуникационной среды “Интернет”.....	16
6.3	Профессиональные базы данных и информационно-справочные системы	16
7	Материально-техническое обеспечение дисциплины	16
8	Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов.....	17
9	Методические материалы.....	18
9.1	Планы практических занятий	18
9.2	Методические рекомендации по подготовке письменных работ	21
	Приложение 1	17

1 Пояснительная записка

1.1 Цель и задачи дисциплины

Цель дисциплины – приобретение знаний, навыков и умений в области программирования на языках высокого уровня.

Задачи дисциплины:

- Изучение базовых принципов программирования на языках высокого уровня;
- Дать широко применяемые технологии и методы программирования для разработки приложений на языках высокого уровня в том числе, специализированные методы обработки и хранения данных;
- Изучение главных управляющих структур языков высокого уровня;
- Познакомить со стандартами на языках высокого уровня и их отличительными особенностями;
- Дать представления о процедуре отладки разрабатываемых приложений.

1.2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций

Компетенция (код и наименование)	Индикаторы компетенций (код и наименование)	Результаты обучения
ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-7.1. Знает основные языки программирования, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий	Знать: базовые принципы программирования на языке высокого уровня. Уметь: применять методы обработки и хранения данных на языках высокого уровня. Владеть: методами использования языков высокого уровня, современных программных сред разработки информационных систем и технологий.
	ОПК-7.2. Умеет применять языки программирования, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ	Знать: главные управляющие структуры языков высокого уровня; принципы объектно-ориентированного программирования; современные программные среды разработки. Уметь: использовать языки высокого уровня, для автоматизации бизнес-процессов и решения других прикладных задач в профессиональной деятельности. Владеть: навыками программирования, отладки и тестирования на языках высокого уровня, основными управляющими структурами языков программирования.
	ОПК-7.3. Владеет навыками программирования, отладки и тестирования	Знать: алгоритмы отладки разрабатываемых приложений; Уметь: разрабатывать алгоритмы решения прикладных задач различных классов. Владеть: навыками программирования, отладки

	прототипов программно-технических комплексов задач	и тестирования на языках высокого уровня, основными управляющими структурами языков программирования.
--	--	---

1.2 Место дисциплины в структуре образовательной программы

Дисциплина «Программирование» относится к обязательной части блока дисциплин учебного плана.

Для освоения дисциплины необходимы знания, умения и владения, сформированные в ходе изучения следующих дисциплин: «Алгоритмы и структуры данных».

В результате освоения дисциплины формируются знания, умения и владения, необходимые для изучения следующих дисциплин: «Базы данных», «Пакетная и потоковая обработка данных», «Проектирование и разработка рекомендательных систем», «Программирование глубоких нейронных сетей».

2 Структура дисциплины

Структура дисциплины для очной формы обучения

Общая трудоёмкость дисциплины составляет 14 з.е., 504 ч., в том числе контактная работа обучающихся с преподавателем 154 ч., промежуточная аттестация 18 ч., самостоятельная работа обучающихся 224 ч.

Семестр	Тип учебных занятий	Количество часов
1	Лекция	18
1	Практическая работа	24
	Всего:	42
2	Лекция	28
2	Практическая работа	42
	Всего:	70
3	Лекция	18
3	Практическая работа	24
	Всего:	42

3 Содержание дисциплины

№	Наименование раздела дисциплины	Содержание
Семестр 1		
1	Введение в среду программирования. Типы данных и составные объекты.	Интерпретаторы и компиляторы. Загрузка среды программирования с сайта разработчика. Знакомство с интерфейсом среды программирования для языков высокого уровня. Стандартные модули языков высокого уровня. Ввод и вывод. Форматированный ввод и вывод. Типизация. Специальные типы. Числа: целые, с плавающей точкой и комплексное.
2	Управляющие конструкции	Полное и неполное ветвление. Реализация

	ходом выполнения программ.	многовариантного выбора через ветвления. Параметрический цикл с вариативной и без вариативной части. Цикл с предусловием. Команды прерывания цикла и ее продолжения со следующей итерации (break, continue). Выполнение команд внутри контекста (with). Обработка исключений в программах. Встроенные типы исключений и определение новых исключений.
3	Функции и основы функционального программирования на языках высокого уровня.	Определение пользовательских функций. Передача параметров и возврат результатов. Значения аргументов функции по умолчанию. Произвольный набор аргументов. Именованные аргументы. Правила видимости. Функции, как объекты и замыкания. Оператор Lambda. Атрибуты функций.
4	Структурированные типы данных. Типовые алгоритмы обработки и их реализация.	Списки, массивы, кортежи, словари, множества. Типовые алгоритмы обработки массивов. Модуль для работы с матрицами. Сортировка (линейная, пузырьковая, вставками и др.).
5	Работа с текстом и строками. Чтение, запись данных из файла и построение графиков на языках высокого уровня.	Анализ, преобразование, поиск, замена, порождение текстовой информации. Кодировки символов на языках высокого уровня. Операции над строками. Стандартные методы обработки строк. Кодировка символов ASCII. Чтение, запись и обработка данных из файла. Построение графиков с помощью библиотек.

Семестр 2

6	Основы объектно-ориентированного программирования.	Структуры данных, её связь с обработкой данных. Уровни структур данных. Уровни данных в программировании. Классификация структур данных. Операции над структурами данных. Отличительные особенности. Объектно-ориентированное программирование. Структурность данных и структурное программирование. Принцип модульного программирования и его применение.
7	Классы. Наследование. Многопоточное программирование на языках высокого уровня.	Основы классов. Общая форма класса. Простой класс. Объявление объектов. Подробное рассмотрение оператора new. Присваивание переменным ссылок на объекты. Введение в методы. Возврат значений. Ввод метода, принимающего параметры. Конструкторы. Параметризованные конструкторы. Перегрузка методов. Перегрузка конструкторов. Применение объектов в качестве параметров. Подробное рассмотрение особенностей передачи аргументов. Возврат объектов. Рекурсия. Введение в управление доступом. Основы наследования.

		Доступ к членам класса и наследование. Создание многоуровневой иерархии. Порядок вызова конструкторов. Переопределение методов. Применение абстрактных классов. Модель потоков исполнения. Главный поток исполнения. Создание потока исполнения. Создание многих потоков исполнения.
8.	Лямбда-функции. Введение в библиотеки языков высокого уровня.	Введение в лямбда-выражения. Блочные лямбда-выражения. Обобщенные функциональные интерфейсы. Передача лямбда-выражений в качестве аргументов. Лямбда-выражения и исключения. Лямбда-выражения и захват переменных. Ссылки на методы. Ссылки на конструкторы. Конструкторы символьных строк. Длина символьной строки. Специальные строковые операции. Извлечение символов. Сравнение символьных строк. Поиск в символьных строках.
9.	Чтение и запись данных в файлы. Автоматическое закрытие файла.	Основы ввода-вывода. Потоки ввода-вывода. Потоки ввода-вывода байтов и символов. Предопределенные потоки ввода-вывода. Чтение данных, вводимых с консоли. Запись данных, выводимых на консоль. Чтение и запись данных в файлы. Автоматическое закрытие файла.
10	Основы создания апплетов.	Основы создания апплетов. Модификаторы доступа. Платформенно-ориентированные методы. Трудности, связанные с платформенно-ориентированными методами. Параметры включения и отключения режима проверки утверждений. Статический импорт.
Семестр 3		
11.	Пользовательские интерфейсы. Последовательность разработки и структура пользовательских интерфейсов	Уровни разработки интерфейса. Уровень стратегии: постановка задачи, технические требования, изучение аудитории, сегментация и профилирование, метод персонажей, пользовательские сценарии. Уровень набора возможностей, спецификация функциональности. Уровень структуры, интерфейсные шаблоны: прямого действия, сохранения контента, сокращения усилий, реализующие приглашения, шаблоны переходов. Уровень компоновки: интерфейсы программы и веб-сайта. Правила взаимодействия человек-компьютер: соответствие ментальной модели, понятность и однозначность, временные характеристики, средства осознанной работы с информацией.
12.	Использование стандартных интерфейсов	Реализация, получение ссылки, интерфейс как параметр, явная реализация, создание иерархий, наследование от нескольких базовых интерфейсов, интерфейсные индексаторы
13.	Делегаты	использование, делегаты как статические члены класса

14.	Перегрузка операторов	операторы отношения, приведения, примеры, унарные, бинарные операторы, пример использования перегрузки операторов, ограничения
15.	Компоненты	определение, IComponent, класс Component, простой компонент, использование, переопределение метода Dispose()

4 Образовательные технологии

При реализации программы дисциплины «Программирование» используются различные образовательные технологии – во время аудиторных занятий, занятия проводятся в виде лекций с использованием ПК и компьютерного проектора (лекция-визуализация), по наиболее сложным вопросам темы занятия проводятся с включением дискуссий (лекция-дискуссия). Практические занятия проводятся в компьютерном классе с использованием специальных компьютерных и промышленных автоматизированных информационных систем.

С целью активизировать работу студентов при освоении теоретического материала, изложенного на лекциях, при проведении практических работ проводится устный экспресс-опрос студентов по вопросам предыдущих практических работ и тем, изложенных на лекциях.

Самостоятельная работа студентов направлена на закрепление полученных навыков и для приобретения новых теоретических и фактических знаний, выполняется в читальном зале библиотеки и в домашних условиях, подкрепляется учебно-методическим и информационным обеспечением (учебники, учебно-методические пособия, конспекты лекций).

15 % – интерактивных занятий от объема аудиторных занятий.

В период временного приостановления посещения обучающимися помещений и территории РГГУ для организации учебного процесса с применением электронного обучения и дистанционных образовательных технологий могут быть использованы следующие образовательные технологии:

- видео-лекции;
- онлайн-лекции в режиме реального времени;
- электронные учебники, учебные пособия, научные издания в электронном виде и доступ к иным электронным образовательным ресурсам;
- системы для электронного тестирования;
- консультации с использованием телекоммуникационных средств.

5 Оценка планируемых результатов обучения

5.1 Система оценивания

Форма контроля	Макс. количество баллов	
	За одну работу	Всего
Текущий контроль: - защита отчета по практической работе	12 баллов	60 баллов
Промежуточная аттестация: зачёт с оценкой		40 баллов
Итого за 1 семестр		100 баллов

Форма контроля	Макс. количество баллов	
	За одну работу	Всего
Текущий контроль: - защита отчета по практической работе	15 баллов	60 баллов
Промежуточная аттестация: зачёт с оценкой		40 баллов
Итого за 2 семестр		100 баллов

Форма контроля	Макс. количество баллов	
	За одну работу	Всего
Текущий контроль: - защита отчета по практической работе	15 баллов	60 баллов
Промежуточная аттестация: экзамен		40 баллов
Итого за 3 семестр		100 баллов

Полученный совокупный результат конвертируется в традиционную шкалу оценок и в шкалу оценок Европейской системы переноса и накопления кредитов (European Credit Transfer System; далее – ECTS) в соответствии с таблицей:

100-балльная шкала	Традиционная шкала	Шкала ECTS
95 – 100	отлично	зачтено
83 – 94		
68 – 82		
56 – 67		
50 – 55		
20 – 49		
0 – 19	удовлетворительно	не зачтено
	неудовлетворительно	

5.2 Критерии выставления оценки по дисциплине

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
100-83/ A,B	«отлично»/ «зачтено (отлично)»/ «зачтено»	Выставляется обучающемуся, если он глубоко и прочно усвоил теоретический и практический материал, может продемонстрировать это на занятиях и в ходе промежуточной аттестации. Обучающийся исчерпывающе и логически стройно излагает учебный материал, умеет увязывать теорию с практикой, справляется с решением задач профессиональной направленности высокого уровня сложности, правильно обосновывает принятые решения. Свободно ориентируется в учебной и профессиональной литературе.

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
		<p>Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «высокий».</p>
82-68/ C	«хорошо»/ «зачтено (хорошо)»/ «зачтено»	<p>Выставляется обучающемуся, если он знает теоретический и практический материал, грамотно и по существу излагает его на занятиях и в ходе промежуточной аттестации, не допуская существенных неточностей.</p> <p>Обучающийся правильно применяет теоретические положения при решении практических задач профессиональной направленности разного уровня сложности, владеет необходимыми для этого навыками и приёмами.</p> <p>Достаточно хорошо ориентируется в учебной и профессиональной литературе.</p> <p>Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «хороший».</p>
67-50/ D,E	«удовлетвори- тельно»/ «зачтено (удовлетвори- тельно)»/ «зачтено»	<p>Выставляется обучающемуся, если он знает на базовом уровне теоретический и практический материал, допускает отдельные ошибки при его изложении на занятиях и в ходе промежуточной аттестации.</p> <p>Обучающийся испытывает определённые затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, владеет необходимыми для этого базовыми навыками и приёмами.</p> <p>Демонстрирует достаточный уровень знания учебной литературы по дисциплине.</p> <p>Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «достаточный».</p>
49-0/ F,FX	«неудовлетворите- льно»/ не зачтено	<p>Выставляется обучающемуся, если он не знает на базовом уровне теоретический и практический материал, допускает грубые ошибки при его изложении на занятиях и в ходе промежуточной аттестации.</p> <p>Обучающийся испытывает серьёзные затруднения в применении теоретических положений при решении практических задач профессиональной</p>

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
		<p>направленности стандартного уровня сложности, не владеет необходимыми для этого навыками и приёмами.</p> <p>Демонстрирует фрагментарные знания учебной литературы по дисциплине.</p> <p>Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции на уровне «достаточный», закреплённые за дисциплиной, не сформированы.</p>

Критерии оценивания практических работ:

Критерии оценивания / Уровень требований к обучающемуся	Макс. кол- во баллов
Текущий контроль, всего в т.ч.:	60
Практическая работа 1 семестр	12
Практическая работа 2 семестр	15
Практическая работа 3 семестр	15
Задания выполнены не полностью и (или) допущены две и более ошибки или три и более недочета	1-6 / 1-8
Задания выполнены полностью, но допущены два-три недочета, в т. ч. при ответе на контрольные вопросы	7-9 / 9-12
Задания выполнены полностью, возможна одна неточность, ответы на контрольные вопросы правильные	10-12 / 13-15

5.3 Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине

Вопросы для подготовки к экзамену

1. Ввод и вывод на языках высокого уровня. Осуществление форматированного ввода и вывода.
2. В чем особенности цикла for.
3. Для чего применяется функция range().
4. Когда и для чего используется цикл while.
5. Как можно организовать досрочный выход из цикла. В каких случаях это используется.
6. Для чего используется оператор continue.
7. Как можно проверить цикл на экстренный выход.
8. В чем особенности типовых алгоритмов циклической структуры: вычисления суммы n слагаемых, вычисления произведения n сомножителей, табулирования функции.
9. Что такое рекуррентное соотношение. Как вычисляется сумма ряда с использованием рекуррентных соотношений.
10. Когда используются вложенные циклы?

11. Что такое разветвление. Какой оператор используется для организации разветвлений.
12. Как реализуется типовая структура «Обход». В чем ее особенность?
13. Что такое логические переменные, логические операторы, логические выражения. Для чего их используют.
14. Назначение библиотеки Math.
15. Как организуется обработка потока данных, количество которых заранее не известно?
16. В чем заключаются особенности организации потокового ввода, если каждая порция данных включает несколько значений?
17. Что такое массив. Элемент массива. Индекс элемента массива. Как можно инициализировать массив. Как организуется вывод массива в строку, в столбец.
18. Каков алгоритм суммирования элементов массива. Суммирования элементов массива, удовлетворяющих условию.
19. Как сформировать другой массива из элементов заданного массива, удовлетворяющих условию. Как осуществляется перестановка элементов массива.
20. В чем заключается удаление элемента массива. Включение элемента в массив.
21. Как найти максимальный (минимальный) элемент массива.
22. Как осуществляется упорядочение элементов массива.
23. Что такое матрица. Как задается матрица. В чем особенности ввода матрицы по строкам, по столбцам? Как осуществить вывод матрицы в наглядной форме.
24. Каков алгоритм суммирования элементов матрицы. Суммирования элементов матрицы, удовлетворяющих условию. Нахождения максимального (минимального) элемента матрицы.
25. Как осуществляются операции со строками (столбцами) матрицы (поиск максимального элемента, включение, удаление элемента, перестановка элементов и т.п.).
26. Как реализовать удаление строки (столбца) матрицы. Включение одномерного массива в качестве строки (столбца) в матрицу.
27. Модуль для работы с матрицами NumPy.
28. В чем особенности операций с главной диагональю, с побочной диагональю.
29. Как осуществляется обработка фрагмента матрицы (верхнего, нижнего треугольника; верхней, нижней, правой, левой четверти; периметра и т.п.).
30. В чем особенности обработки матрицы, заданной в виде одномерной последовательности.
31. Как осуществляется вывод матрицы, заданной в виде одномерной последовательности?
32. Чем заключается принцип модульного программирования?
33. Что такое функция. Как выглядит общий синтаксис описания функции.
34. Каковы разновидности функций по характеру возвращаемого значения. В чем особенности их оформления.
35. Как осуществляется вызов функции. В чем специфика обращения к функции в зависимости от количества ее параметров: функция без параметров, с фиксированным количеством параметров, с произвольным количеством параметров.
36. Оператор Lambda.
37. В чем заключаются особенности задания собственных функций.
38. Как происходит использование параметров по умолчанию при работе с функциями.
39. Что такое локальные и глобальные переменные. В чем особенности работы с указанными видами переменных.
40. Назначение библиотеки Matplotlib.

41. Объектно-ориентированное программирование.
42. Примитивные типы. Числа с плавающей точкой. Символы.
43. Логические значения. Переменные. Преобразование и приведение типов. Автоматическое продвижение типов в выражениях.
44. Массивы. Арифметические операции. Поразрядные операции. Операции отношения.
45. Операция присваивания. Тернарная операция ?. Операторы выбора.
46. Условный оператор if. Оператор switch.
47. Операторы цикла. Цикл while. Цикл do...while. Цикл for.
48. Вложенные циклы. Операторы перехода. Применение оператора break.
49. Применение оператора continue. Оператор return.
50. Основы классов. Общая форма класса. Простой класс. Объявление объектов.
51. Подробное рассмотрение оператора new. Присваивание переменным ссылок на объекты.
52. Введение в методы. Возврат значений. Ввод метода, принимающего параметры.
53. Конструкторы. Параметризованные конструкторы.
54. Ключевое слово this. Сокрытие переменной экземпляра. Сборка "мусора".
55. Перегрузка методов. Перегрузка конструкторов.
56. Применение объектов в качестве параметров.
57. Возврат объектов. Рекурсия. Введение в управление доступом.
58. Ключевое слово static. Ключевое слово final.
59. Основы наследования. Доступ к членам класса и наследование.
60. Создание многоуровневой иерархии.
61. Порядок вызова конструкторов. Переопределение методов.
62. Применение абстрактных классов.
63. Модель потоков исполнения в C#. Главный поток исполнения.
64. Создание потока исполнения. Создание многих потоков исполнения.
65. Введение в лямбда-выражения. Блочные лямбда-выражения.
66. Обобщенные функциональные интерфейсы. Передача лямбда-выражений в качестве аргументов.
67. Лямбда-выражения и исключения. Лямбда-выражения и захват переменных.
68. Ссылки на методы. Ссылки на конструкторы.
69. Конструкторы символьных строк. Длина символьной строки.
70. Специальные строковые операции. Извлечение символов.
71. Сравнение символьных строк. Поиск в символьных строках.
72. Основы ввода-вывода. Потоки ввода-вывода.
73. Потоки ввода-вывода байтов и символов. Предопределенные потоки ввода-вывода.
74. Чтение данных, вводимых с консоли. Запись данных, выводимых на консоль.
75. Чтение и запись данных в файлы.
76. Автоматическое закрытие файла.
77. Основы создания апплетов. Модификаторы доступа.
78. Платформенно-ориентированные методы.
79. Трудности, связанные с платформенно-ориентированными методами.
80. Параметры включения и отключения режима проверки утверждений.
81. Статический импорт. Вызов перегружаемых конструкторов по ссылке this().
82. Пользовательские интерфейсы готовые формы.
83. Графический интерфейс Tkinter.
84. Подходы к разработке пользовательских интерфейсов.
85. Уровень стратегии: постановка задачи, определение технических требований.
86. Интерфейсные шаблоны: прямого действия, сохранения контента на странице.
87. Интерфейсные шаблоны: сокращения усилий, приглашения.

88. Интерфейсные шаблоны: переходы, мгновенная реакция интерфейса.
89. Интерфейсы программных продуктов и веб-сайтов.
90. Правила взаимодействия человек-компьютер: соответствие ментальной модели, понятность, однозначность.
91. Правила взаимодействия человек-компьютер: временные характеристики, осознанной работы с информацией.
92. Документальное оформление пользовательского интерфейса, справочная система.
93. Определение, значения понятия: Дизайн.
94. Эстетическая и функциональная стороны дизайна.
95. Роль цвета в дизайне интерфейсов.
96. Форма объектов в дизайне интерфейсов.
97. Принципы композиции в дизайне интерфейсов.
98. Текст, анимация, звук в дизайне интерфейсов.
99. Эргономика, ее применение, сенсомоторные факторы в дизайне интерфейсов.
100. Антропометрические факторы при работе на компьютере.

6 Учебно-методическое и информационное обеспечение дисциплины

6.1 Список литературы

Основная

1. Федоров, Д. Ю. Программирование на Python : учебное пособие для вузов / Д. Ю. Федоров. — 6-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 187 с. — (Высшее образование). — ISBN 978-5-534-19666-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://www.urait.ru/bcode/556864>
2. Гниденко И. Г. Технологии и методы программирования : учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 248 с. — URL: <https://urait.ru/bcode/536903>.
3. Чернышев, С. А. Основы программирования на Python : учебное пособие для вузов / С. А. Чернышев. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 349 с. — (Высшее образование). — ISBN 978-5-534-17139-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://www.urait.ru/bcode/544190>
4. Функциональное программирование. Теоретические и практические основы для разных языков : учебник для вузов / А. Ю. Анисимов [и др.]; под общей редакцией А. Ю. Анисимова, А. Е. Трубина, Ф. А. Маstryева. — Москва : Издательство Юрайт, 2025. — 135 с. — (Высшее образование). — ISBN 978-5-534-20518-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://www.urait.ru/bcode/558300>
5. Зыков С. В. Программирование. Функциональный подход : учебник и практикум для вузов / С. В. Зыков. — 2-е изд. — Москва : Издательство Юрайт, 2024. — 150 с. — (Высшее образование). — ISBN 978-5-534-16942-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://www.urait.ru/bcode/537721>
6. Жуков Р. А. Язык программирования Python: практикум : учебное пособие / Р.А. Жуков. — Москва : ИНФРА-М, 2023. — 216 с. + Доп. материалы [Электронный ресурс]. — (Высшее образование: Бакалавриат). — DOI 10.12737/textbook_5cb5ca35aaa7f5.89424805. - ISBN 978-5-16-016971-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1915716>
7. Яворски М. Python. Лучшие практики и инструменты : практическое руководство / М. Яворски, Т. Зиаде. - Санкт-Петербург : Питер, 2021. - 560 с. - (Серия «Библиотека

- программиста»). - ISBN 978-5-4461-1589-1. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1739603>
8. Титов А. Н. Python. Обработка данных : учебно-методическое пособие / А. Н. Титов, Р. Ф. Тазиева ; Минобрнауки России, Казан. нац. исслед. технол. ун-т. - Казань : Изд-во КНИТУ, 2022. - 104 с. - ISBN 978-5-7882-3171-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/2069264>
 9. Тузовский, А. Ф. Объектно-ориентированное программирование : учебное пособие для вузов / А. Ф. Тузовский ; Томский политехнический университет. - Москва : Юрайт, 2022. - URL: <https://urait.ru/bcode/414163>
 10. Ткаченко, О. Н. Взаимодействие пользователей с интерфейсами информационных систем для мобильных устройств: исследование опыта : учебное пособие / О.Н. Ткаченко. — Москва : Магистр : ИНФРА-М, 2022. — 152 с. - ISBN 978-5-9776-0457-4. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1859029>
 - 11.

Дополнительная

1. Основы алгоритмизации и программирования на языке Python : учебник / О.В. Андреева, О.И. Ремизова. – Москва : Издательский Дом НИТУ «МИСиС», 2022. – 149 с.
2. Груздев А. В. Предварительная подготовка данных в Python: Том 1. Инструменты и валидация : практическое руководство / А. В. Груздев. - Москва : ДМК Пресс, 2023. - 816 с. - ISBN 978-5-93700-156-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/2109509>
3. Чан Д. Python: быстрый старт : практическое руководство / Д. Чан. - Санкт-Петербург: Питер, 2021. - 224 с. - (Серия «Библиотека программиста»). - ISBN 978-5-4461-1800-7. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1739605>
4. Седер, Н. Python. Экспресс-курс : практическое руководство / Н. Седер. - Санкт-Петербург : Питер, 2021. - 480 с. - (Серия «Библиотека программиста»). - ISBN 978-5-4461-0908-1. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1756159>
5. Шевченко Л. Г. Программирование на PYTHON в среде IDLE : учебное пособие / Л. Г. Шевченко, Т. В. Дружинина. - Новосибирск : Изд-во НГТУ, 2020. - 195 с. - ISBN 978-5-7782-4215-9. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1866915>
6. Плас Дж. В. Python для сложных задач: наука о данных и машинное обучение : практическое руководство / Дж. В. Плас. - Санкт-Петербург : Питер, 2021. - 576 с. - (Серия «Бестселлеры O'Reilly»). - ISBN 978-5-4461-0914-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1739601>
7. Matplotlib. Урок 4.2. Визуализация данных. Ступенчатый, стековый, точечный график и другие. - URL: <https://devpractice.ru/matplotlib-lesson-4-2-stack-stem-scatter/> (дата доступа 15.03.24)
8. Магазанник, В. Д. Человеко-компьютерное взаимодействие : учебное пособие / В. Д. Магазанник. - 2-е изд., доп. - Москва : Университетская книга, 2020. - 408 с. - ISBN 978-5-98699-181-8. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1214481>
9. Impact Mapping. Как повысить эффективность программных продуктов и проектов по их разработке: Практическое руководство / Аджич Г. - М.:Альпина Паблишер, 2017. - 86 с.: 60x90 1/8 (Обложка) ISBN 978-5-9614-6448-1 - Режим доступа: <http://znanium.com/catalog/product/1003514>
10. Разработка пользовательского интерфейса: принципы – URL: <https://texterra.ru/blog/razrabotka-polzovatelskogo-interfeysa-kak-sozdat-gui.html> (дата доступа 24.02.24)

6.2 Перечень ресурсов информационно-коммуникационной среды “Интернет»

1. Форум программистов и сисадминов Киберфорум <http://www.cyberforum.ru/>
2. Клуб программистов <https://programmersforum.ru/>
3. Форум программистов <https://programmersforum.ru/>

6.3 Профессиональные базы данных и информационно-справочные системы

Доступ к профессиональным базам данных: <https://liber.rsuh.ru/ru/bases>

Информационные справочные системы:

1. Консультант Плюс
2. Гарант

7 Материально-техническое обеспечение дисциплины

Для материально-технического обеспечения дисциплины необходимы:

- для лекций:

- учебная аудитория,
- доска,
- проектор (стационарный или переносной),
- компьютер или ноутбук,
- программное обеспечение (ПО).

- для практических занятий:

- лаборатория,
- доска,
- проектор (стационарный или переносной),
- компьютер или ноутбук для преподавателя,
- компьютеры для обучающихся,
- выход в Интернет,
- программное обеспечение (ПО).

Перечень программного обеспечения (ПО)

- для лекций:

№п/п	Наименование ПО	Способ распространения
1	Microsoft Office 2010 Pro	лицензионное
2	Windows 10 и современное	лицензионное
3	Kaspersky Endpoint Security	лицензионное

- для практических занятий:

Наименование ПО	Способ распространения
Windows 10 и современное	лицензионное
Microsoft Office 2010 Pro или аналоги	лицензионное
Mozilla Firefox или др. браузер	свободно распространяемое
Kaspersky Endpoint Security	лицензионное
Python 3.11	свободно распространяемое
PyCharm	лицензионное
Microsoft Visual Professional 2019	лицензионное

8 Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов

В ходе реализации дисциплины используются следующие дополнительные методы обучения, текущего контроля успеваемости и промежуточной аттестации обучающихся в зависимости от их индивидуальных особенностей:

- для слепых и слабовидящих:
 - лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
 - письменные задания выполняются на компьютере со специализированным программным обеспечением, или могут быть заменены устным ответом;
 - обеспечивается индивидуальное равномерное освещение не менее 300 люкс;
 - для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств;
 - письменные задания оформляются увеличенным шрифтом;
 - экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.
- для глухих и слабослышащих:
 - лекции оформляются в виде электронного документа, либо предоставляется звукоусиливающая аппаратура индивидуального пользования;
 - письменные задания выполняются на компьютере в письменной форме;
 - экзамен и зачёт проводятся в письменной форме на компьютере; возможно проведение в форме тестирования.
- для лиц с нарушениями опорно-двигательного аппарата:
 - лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
 - письменные задания выполняются на компьютере со специализированным программным обеспечением;
 - экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

При необходимости предусматривается увеличение времени для подготовки ответа.

Процедура проведения промежуточной аттестации для обучающихся устанавливается с учётом их индивидуальных психофизических особенностей. Промежуточная аттестация может проводиться в несколько этапов.

При проведении процедуры оценивания результатов обучения предусматривается использование технических средств, необходимых в связи с индивидуальными особенностями обучающихся. Эти средства могут быть предоставлены университетом, или могут использоваться собственные технические средства.

Проведение процедуры оценивания результатов обучения допускается с использованием дистанционных образовательных технологий.

Обеспечивается доступ к информационным и библиографическим ресурсам в сети Интернет для каждого обучающегося в формах, адаптированных к ограничениям их здоровья и восприятия информации:

- для слепых и слабовидящих:
 - в печатной форме увеличенным шрифтом;
 - в форме электронного документа;
 - в форме аудиофайла.
- для глухих и слабослышащих:
 - в печатной форме;
 - в форме электронного документа.
- для обучающихся с нарушениями опорно-двигательного аппарата:
 - в печатной форме;

- в форме электронного документа;
- в форме аудиофайла.

Учебные аудитории для всех видов контактной и самостоятельной работы, научная библиотека и иные помещения для обучения оснащены специальным оборудованием и учебными местами с техническими средствами обучения:

- для слепых и слабовидящих:
 - устройством для сканирования и чтения с камерой SARA CE;
 - дисплеем Брайля PAC Mate 20;
 - принтером Брайля EmBraille ViewPlus;
- для глухих и слабослышащих:
 - автоматизированным рабочим местом для людей с нарушением слуха и слабослышащих;
 - акустический усилитель и колонки;
- для обучающихся с нарушениями опорно-двигательного аппарата:
 - передвижными, регулируемыми эргономическими партами СИ-1;
 - компьютерной техникой со специальным программным обеспечением.

9 Методические материалы

9.1 Планы практических занятий

1 семестр

Практическая работа №1. Первая программа на Python.

Целью работы является формирование умений работать в интегрированной среде разработки приложений, составлять и отлаживать программы на языках высокого уровня, освоить простые операции ввода-вывода. Приложение должно обеспечивать функционал, способный производить определённые действия, в зависимости от условия и введенных данных пользователем с консоли, и корректно выводить результат.

Вопросы для защиты практической работы №1:

1. Встроенные типы чисел. Машинное представление чисел с плавающей точкой и целых.
2. Ввод и вывод с консоли.
3. Преобразование типов при сравнении чисел.
5. Инструкции и синтаксис. Составные конструкции. Инструкции `if/else/elif`, логические операторы и выражения сравнения.

Практическая работа №2. Разработка программ циклической структуры.

Целью работы является формирование умений разрабатывать программы на языках высокого уровня с использованием циклических структур. Приложение должно обеспечивать решение ряда задач по заданному алгоритму, включающему в себя повторение вычислений и выбор данных.

Вопросы для защиты практической работы №2:

1. Типы циклов.
2. Как можно организовать досрочный выход из цикла. В каких случаях это используется?
3. Когда используются вложенные циклы.

Практическая работа №3. Принцип модульного программирования.

Целью работы является формирование умений разрабатывать программы на языках высокого уровня с использованием функций, являющихся элементом модульного программирования. Приложение должно обеспечивать решение ряда задач с использованием функций способных как самостоятельно проводить вычисления, так и передавать данные другим структурам.

Вопросы для защиты практической работы №3:

1. В чем заключается принцип модульного программирования.
2. Как осуществляется вызов функции.
3. В чем специфика обращения к функции в зависимости от количества ее параметров: функция без параметров, с фиксированным количеством параметров, с произвольным количеством параметров.
4. Каковы разновидности функций по характеру возвращаемого значения. В чем особенности их оформления.

Практическая работа №4. Структурированные типы данных.

Целью работы является формирование умений разрабатывать программы на языках высокого уровня с использованием структурированных типов данных (списков, кортежей, словарей). Приложение должно обеспечивать решение ряда задач с использованием структурированных типов данных для формирования их, нахождения заданных параметров, сортировки или замены данных.

Вопросы для защиты практической работы №4:

1. Упорядочение элементов списка.
2. Операции со строками (столбцами) матрицы (поиск максимального элемента, включение, удаление элемента, перестановка элементов и т.п.).
3. Корректный вывод двумерной матрицы.
4. Операции над строками. Стандартные методы обработки строк.
5. Кодировка символов ASCII.

Практическая работа №5. Обработка электронной таблицы и построение графика.

Целью работы является формирование умений работать с электронными таблицами на языке высокого уровня, осуществлять их считывание и сохранение в файл CSV. Приложение должно обеспечивать считывание и преобразование электронной таблицы по заданному алгоритму. Построение графика по выбранным данным.

Вопросы для защиты практической работы №5:

1. Библиотека Matplotlib.
2. Встроенный объект dict и его методы. Множества.
3. Встроенные типы set и frozenset.
4. Работа с файлами. Чтение и запись файла.

2 семестр

Практическая работа №1. Классы. Наследование. Многопоточное программирование.

Задание:

1. Согласно заранее определенному варианту выполнить объявление многомерного статического массива;
2. Выполнить поиск и перестановку минимального и максимального элемента.
3. Подготовить развернутый отчет о проделанной работе.

Практическая работа №2. Лямбда-функции. Введение в библиотеку.

Задание:

1. Согласно заранее определенному варианту выполнить программирование алгоритмов сортировки с использованием лямбда-функции;
2. Методы сортировки должны быть реализованы с использованием библиотек lang ;
3. Подготовить развернутый отчет о проделанной работе.

Практическая работа №3. Апплеты. Основы создания апплетов.

Задание:

1. Согласно заранее определенному варианту и используя принципы объектно-ориентированного программирования организовать создание объекта типа **class** в состав,

- которого входят методы сортировки динамически инициализированного двумерного массива, а также определены конструктор и деструктор класса;
- Предусмотреть возможность создания дочерних классов на базе созданного родительского класса;
 - Подготовить развернутый отчет о проделанной работе.

Практическая работа №4. Использование стандартных интерфейсов, делегатов, компонентов

Задание:

- Согласно заранее определенному варианту и используя принципы объектно-ориентированного программирования организовать создание стандартного интерфейса (делегатов / компонентов / перегруженных операторов класса);
- Предусмотреть возможность создания дочерних классов на базе созданного родительского класса;
- Подготовить развернутый отчет о проделанной работе.

3 семестр

Практическая работа №1. Интерфейс. Начало работы с Tkinter

Задание:

- Создание проекта пользовательского элемента управления, согласно заранее определенному варианту.
- Разработка макета элемента управления.
- Заполнение элемента управления.
- Использование окна структуры документа.
- Добавление обработчиков событий.
- Тестирование элемента управления.
- Использование элемента управления в форме.

Практическая работа №2. Введение в разработку графического интерфейса.

Задание:

- Знакомство со средством QT5.
- Ключевые понятия.
- Действия со средством QT5..
- Обзор изученного.
- Предпросмотр графического интерфейса, согласно выданному варианту.
- Развёртывание приложения с графическим интерфейсом.

Практическая работа №3. Обработка изображений в приложении с графическим интерфейсом..

Задание:

- Создание формы приложения, согласно заранее определенному варианту.
- Добавление пакета для изображения.
- Вывод изображения на метке.
- Сборка и запуск приложения.
- Показ изображения как фона в кадре.
- Создание пользовательского кода.

Практическая работа №4. Интернационализация формы графического интерфейса.

- Принцип интернационализации.
- Как настроить автоматическую интернационализацию на этапе проектирования.
- Интернационализация всего проекта.

4. Интернационализация одной формы.

9.2 Методические рекомендации по подготовке письменных работ

Отчет по проделанной работе должен быть изложен с соблюдением правил грамматики русского и английского языков (в случаях необходимости). При этом отражаемые результаты работы должны быть информативными, тезисного порядка. В отчет входят следующие обязательные разделы:

1. Титульный лист с полным указанием ведомственной принадлежности, названия ВУЗа, института, факультета, кафедры. Кроме того, полное точное название лабораторной работы, Ф.И.О. студента, подготовившего отчет о результатах проделанной работы и Ф.И.О., должность, название кафедры преподавателя осуществляющего проверку и оценивание полученных результатов.
2. Содержание.
3. Введение.
4. Цели и задачи практической работы.
5. Методы и технологии, применяемые для решения поставленных задач оформленные в виде отдельных этапов работы.
6. Выводы по работе.
7. Приложения.

Оформление отчета выполняется с использованием компьютерной верстки LaTeX. Отчет сохраняется и представляется для проверки в виде отдельного файла. В имени файла показывается фамилия студента и номер выполненной работы.

АННОТАЦИЯ ДИСЦИПЛИНЫ

Цель дисциплины – приобретение знаний, навыков и умений в области программирования на языках высокого уровня.

Задачи дисциплины:

- Изучение базовых принципов программирования на языках высокого уровня;
- Дать широко применяемые технологии и методы программирования для разработки приложений на языках высокого уровня в том числе, специализированные методы обработки и хранения данных;
- Изучение главных управляющих структур языков высокого уровня;
- Познакомить со стандартами на языках высокого уровня и их отличительными особенностями;
- Дать представления о процедуре отладки разрабатываемых приложений.

В результате освоения дисциплины обучающийся должен:

Знать: базовые принципы программирования на языке высокого уровня; главные управляющие структуры языков; принципы объектно-ориентированного программирования; современные программные среды разработки; алгоритмы отладки разрабатываемых приложений.

Уметь: применять методы обработки и хранения данных на языках высокого уровня; использовать языки высокого уровня, для автоматизации бизнес-процессов и решения других прикладных задач в профессиональной деятельности; разрабатывать алгоритмы решения прикладных задач различных классов.

Владеть: методами использования языков высокого уровня, современных программных сред разработки информационных систем и технологий; навыками программирования, отладки и тестирования на языках высокого уровня, основными управляющими структурами языков программирования.