Приложение 4

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Нижегородский государственный университет
им. Н.И. Лобачевского»

Арзамасский филиал ННГУ - Факультет естественных и математических наук

УТВЕРЖДЕНО

решением Ученого совета ННГУ

протокол № 15 от 24.12.2025 г.

Рабочая программа дисциплины

Алгоритмизация и программирование

Уровень высшего образования
Бакалавриат

Направление подготовки / специальность
09.03.03 - Прикладная информатика

Направленность образовательной программы
Прикладная информатика в экономике

Форма обучения
очно-заочная

г. Арзамас

2026 год начала подготовки

1. Место дисциплины в структуре ОПОП

Дисциплина Б1.О.14 Алгоритмизация и программирование относится к обязательной части образовательной программы.

2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	**Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции**		**Наименование оценочного средства**	
	Индикатор достижения компетенции (код, содержание индикатора)	**Результаты обучения по дисциплине**	**Для текущего контроля успеваемости**	**Для промежуточной аттестации**
ОПК-3: Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	*ОПК-3.1: Демонстрирует знание принципов, методов и средств решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности* *ОПК-3.2: Демонстрирует умение применять информационно-коммуникационные технологии решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с учетом основных требований информационной безопасности* *ОПК-3.3: Имеет практический опыт решения стандартных задач профессиональной деятельности с соблюдением требований информационной безопасности*	*ОПК-3.1:* *Знать принципы, методы и средства решения стандартных задач профессиональной деятельности* *Уметь выбрать принципы, методы и средства решения стандартных задач профессиональной деятельности* *Владеть навыками применения методов и средств решения стандартных задач профессиональной деятельности.* *ОПК-3.2:* *Знать принципы решения стандартных задач профессиональной деятельности* *Уметь выбрать способы решения задач профессиональной деятельности* *Владеть навыками выбора способа решения задач профессиональной деятельности.* *ОПК-3.3:* *Знать особенности подготовки обзоров, аннотаций, составления*	*Практическое задание* *Опрос*	*Зачёт:* *Контрольные вопросы* *Экзамен:* *Контрольные вопросы*

		рефератов, научных докладов, публикаций, и библиографии по научно-исследовательской работе с учетом требований информационной безопасности *Уметь подготовить обзоры, аннотации, рефераты, научные публикации, и библиографию по научно-исследовательской работе с учетом требований информационной безопасности* *Владеть навыками подготовки обзоров, аннотаций, составления рефератов, научных докладов, публикаций, и библиографии по научно-исследовательской работе с учетом требований информационной безопасности.*		
ОПК-7: Способен разрабатывать алгоритмы и программы, пригодные для практического применения	*ОПК-7.1: Демонстрирует знание основных языков программирования и работы с базами данных, операционных систем и оболочек, современных программных сред разработки информационных систем и технологий* *ОПК-7.2: Применяет языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ* *ОПК-7.3: Имеет практический опыт программирования, отладки и тестирования прототипов программно-технических комплексов задач*	*ОПК-7.1:* *Знать языки структурного и объектно-ориентированного программирования, среды разработки ПО для данных языков, современные СУБД.* *Уметь применять языки структурного и объектно-ориентированного программирования, среды разработки ПО для данных языков, современные СУБД, вести документацию и сопровождать внедренное ПО.* *Владеть языками структурного и объектно-ориентированного программирования, методами применения сред разработки ПО для данных языков, современными СУБД, техникой ведения документации и технологией сопровождения внедренного ПО.* *ОПК-7.2:* *Знать основы применения*	*Практическое задание* *Опрос*	*Зачёт:* *Контрольные вопросы* *Экзамен:* *Контрольные вопросы*

		современных языков программирования и работы с базами данных *Уметь применять современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов* *Владеть навыками решения прикладных задач различных классов, ведения баз данных и информационных хранилищ.* *ОПК-7.3:* *Знать основы программирования, отладки и тестирования прототипов программно-технических комплексов задач.* *Уметь осуществлять программирование, отладку и тестирование прототипов программно-технических комплексов задач.* *Владеть навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач.*		

3. Структура и содержание дисциплины

3.1 Трудоемкость дисциплины

	очно-заочная
Общая трудоемкость, з.е.	**7**
Часов по учебному плану	**252**
в том числе	
аудиторные занятия (контактная работа):	
- занятия лекционного типа	**32**
- занятия семинарского типа (практические занятия / лабораторные работы)	**32**
- КСР	**4**
самостоятельная работа	**148**
Промежуточная аттестация	**36** **Экзамен, Зачёт**

3.2. Содержание дисциплины

(структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий)

.

Наименование разделов и тем дисциплины	Всего (часы)	в том числе			
		Контактная работа (работа во взаимодействии с преподавателем), часы из них			Самостоятельная работа обучающегося, часы
		Занятия лекционного типа	Занятия семинарского типа (практические занятия/лабораторные работы), часы	Всего	
	очно-заочно-заочно	очно-заочно-заочно	очно-заочно-заочно	очно-заочно-заочно	очно-заочно-заочно
Тема 1. Введение. Основы алгоритмизации. Стандартные алгоритмы.	13	1	2	3	10
Тема 2. Язык Python. Основные сведения. Стандартные типы данных.	13	2	2	4	9
Тема 3. Синтаксис, операторы и управляющие конструкции.	13	2	2	4	9
Тема 4. Функции.	13	2	2	4	9
Тема 5. Модули и пакеты.	13	2	2	4	9
Тема 6. Списки.	13	2	2	4	9
Тема 7. Множества, кортежи, словари.	13	2	2	4	9
Тема 8. Итерации.	13	2	2	4	9
Тема 9. Реализация алгоритмов.	13	2	2	4	9
Тема 10. Обработка исключений в Python.	12	2	2	4	8
Тема 11. Работа с файлами.	12	2	2	4	8
Тема 12. Объектно-ориентированное программирование на языке Python.	12	2	2	4	8
Тема 13. Итераторы, генераторы и декораторы.	14	2	2	4	10
Тема 14. Потоки, процессы и асинхронное программирование.	12	2	2	4	8
Тема 15. Разработка графического пользовательского интерфейса.	12	2	2	4	8
Тема 16. Сетевое программирование.	11	2	1	3	8
Тема 17.Хранение данных и обмен данными. Тестирование.	10	1	1	2	8
Аттестация	36				
КСР	4			4	
Итого	252	32	32	68	148

Содержание разделов и тем дисциплины

Тема 1. Введение. Основы алгоритмизации. Стандартные алгоритмы.
Методологии программирования. Программирование как раздел информатики. Метафоры (парадигмы) программирования. Методологии программирования. Основные понятия и определения. История и эволюция. Классификация по ядрам методологии: императивное программирование, объектно-ориентированное, функциональное, логическое. Синтаксис и семантика формального языка. Естественные и формальные языки. Понятия о синтаксисе и семантике формального языка. Алгоритмизация задачи: понятие алгоритма, свойства алгоритма и способы записи алгоритма. Программирование задачи, реализация программного продукта, технологии отладки, анализ результатов решения. Язык программирования. Классификация языков программирования. Система программирования. Основные технологические этапы создания и использования программных

продуктов. Постановка задачи, формирование математической модели решения.
Тема 2. Язык Python. Основные сведения. Стандартные типы данных.
Общее знакомство с Python: краткий обзор языка, его синтаксиса и философии. Основные понятия: переменные, операторы, ввод/вывод данных. Стандартные типы данных: целые числа (int), числа с плавающей точкой (float), строки (str), булевы значения (bool), списки (list), кортежи (tuple), словари (dict), множества (set). Описание каждого типа, его свойств и способов использования. Базовые операции с данными: арифметические операции, операции сравнения, операции конкатенации (для строк), индексация и срезы (для последовательностей).
Тема 3. Синтаксис, операторы и управляющие конструкции.
Подробное описание синтаксиса Python: отступы, зарезервированные слова, правила именования переменных. Арифметические, логические и операторы сравнения: их приоритет и ассоциативность. Управляющие конструкции: 1) Условные операторы: if, elif, else. 2) Циклы: for (с итераторами) и while. 3) Операторы управления потоком выполнения: break, continue, pass.
Тема 4. Функции.
Определение и вызов функций: синтаксис, параметры (позиционные, именованные, параметры по умолчанию), возвращаемые значения. Область видимости переменных (scope): локальные, глобальные, нелокальные переменные. Анонимные функции (lambda-функции): краткое определение простых функций. Рекурсивные функции. Документирование функций (docstrings): написание помощи для функций. Аргументы *args и kwargs: работа с переменным числом аргументов. Вложенные функции.
Тема 5. Модули и пакеты.
Модули: что такое модуль, импорт модулей (import, from...import), использование встроенных модулей. Пакеты: организация модулей в пакеты, иерархия пакетов, импорт из пакетов. Создание собственных модулей и пакетов: структура, правила именования, использование __init__.py. Стандартная библиотека Python: обзор наиболее важных модулей (например, os, sys, math, random, datetime).
Тема 6. Списки.
Создание списков: различные способы инициализации. Основные операции со списками: добавление элементов (append, extend, insert), удаление элементов (pop, remove, del), изменение элементов. Доступ к элементам: индексация, срезы (slices). Итерация по спискам. Вложенные списки. Методы списков. List comprehensions: создание списков с помощью выражений. Работа с копиями списков.
Тема 7. Множества, кортежи, словари.
Множества (sets): создание множеств, основные операции (объединение, пересечение, разность, симметрическая разность), проверка принадлежности элемента, использование множеств для удаления дубликатов. Кортежи (tuples): создание кортежей, неизменяемость кортежей, доступ к элементам, использование кортежей для представления неизменяемых данных. Словари (dictionaries): создание словарей, ключ-значение пары, доступ к значениям по ключам, добавление, удаление и изменение элементов, итерация по словарям, методы словарей.
Тема 8. Итерации.
Цикл for: использование цикла for для перебора элементов итерируемых объектов (списки, кортежи, строки, словари, файлы и т.д.). Итерируемые объекты: определение и примеры итерируемых объектов. Итераторы: понимание концепции итераторов, методы __iter__ и __next__.
Генераторы: создание генераторов с помощью функции с ключевым словом yield, преимущества использования генераторов (ленивая генерация). Цикл while: использование для итераций с условием. enumerate(): итерация с индексами. zip(): итерация по нескольким итерируемым объектам одновременно.
Тема 9. Реализация алгоритмов.
Основные алгоритмические парадигмы: последовательность, ветвление, циклы, рекурсия. Реализация базовых алгоритмов: поиск (линейный, бинарный), сортировка (пузырьком, выбором, вставками, быстрая сортировка – возможно, только упоминание), поиск в ширину/глубину. Структуры данных: использование списков, множеств, словарей, кортежей и других структур данных для эффективной реализации алгоритмов. Анализ эффективности алгоритмов: оценка временной и пространственной сложности (большое O). Примеры реализации алгоритмов на Python.

Тема 10. Обработка исключений в Python.
Что такое исключения: определение и типы исключений. Блок try...except: основной механизм обработки исключений. Обработка нескольких исключений. Блок else: код, выполняемый, если исключений не возникло. Блок finally: код, выполняемый всегда (после try и except). Raise: явный вызов исключения. Пользовательские исключения. Assert: проверка условий и генерация исключений при нарушении. Обработка исключений в функциях: как обрабатывать исключения в вызываемых функциях.
Тема 11. Работа с файлами.
Открытие файлов: различные режимы открытия (r, w, a, x, b, t, +). Чтение файлов. Запись в файлы. Закрытие файлов, использование менеджера контекста (with open(...) as f:). Работа с путями к файлам: использование модуля os или pathlib для работы с путями, создание и удаление файлов и директорий. Обработка ошибок при работе с файлами: обработка исключений (IOError, FileNotFoundError). Бинарные файлы.
Тема 12. Объектно-ориентированное программирование на языке Python.
Основные понятия ООП: классы, объекты, атрибуты, методы. Создание классов: синтаксис, конструктор (__init__), методы. Наследование: создание новых классов на основе существующих, переопределение методов. Полиморфизм: использование методов с одинаковым именем в разных классах. Инкапсуляция: скрытие данных и методов внутри класса, использование модификаторов доступа (хотя Python не имеет строгих модификаторов, принято использовать соглашения об именовании, такие как _private и __very_private).
Абстрактные классы: классы, от которых нельзя создавать объекты напрямую. Специальные методы: __str__, __repr__, __len__, __getitem__ и другие, позволяющие взаимодействовать с объектами класса на более высоком уровне.
Тема 13. Итераторы, генераторы и декораторы.
Итераторы: понятие итератора, методы __iter__ и __next__, использование итераторов для обхода коллекций. Генераторы: создание генераторов с помощью функций с yield, ленивая генерация, экономия памяти, использование генераторов в циклах for. Декораторы: определение декораторов, синтаксис, декораторы с параметрами, использование декораторов для расширения функциональности функций без изменения их исходного кода. Примеры использования декораторов (например, для логирования, измерения времени выполнения).
Тема 14. Потоки, процессы и асинхронное программирование.
Многопоточность: создание и управление потоками, понимание GIL (Global Interpreter Lock) и его ограничений в Python, применение многопоточности для задач ввода-вывода. Многопроцессорность: создание и управление процессами, преимущества многопроцессорности перед многопоточностью в Python из-за отсутствия GIL, обмен данными между процессами.
Асинхронное программирование: основы асинхронного программирования, async и await, asyncio модуль, обработка ввода-вывода без блокировки, преимущества в задачах с большим количеством операций ввода-вывода.
Тема 15. Разработка графического пользовательского интерфейса.
Выбор библиотеки GUI: обзор популярных библиотек, таких как Tkinter (стандартная библиотека), PyQt, Kivy, wxPython. Основные элементы GUI: окна, кнопки, метки, поля ввода, списки, меню. Обработка событий: реакция на действия пользователя (нажатия кнопок, ввод текста). Размещение элементов: менеджеры размещения (layout managers). Создание простых приложений: примеры создания простых оконных приложений с использованием выбранной библиотеки.
Тема 16. Сетевое программирование.
Основы сетевых протоколов: краткий обзор TCP/IP, UDP, HTTP. Модуль socket: использование socket для создания сетевых сокетов. Клиент-серверная архитектура: разработка простых клиентских и серверных приложений. Обработка запросов и ответов: передача данных между клиентом и сервером. Работа с различными протоколами: примеры работы с TCP и UDP.
Тема 17.Хранение данных и обмен данными. Тестирование.
Форматы данных: JSON, XML, CSV, форматы бинарных данных. Чтение и запись данных в этих

форматах. Базы данных: краткий обзор различных типов баз данных (SQL, NoSQL), введение в работу с базами данных через Python (например, с помощью библиотеки sqlite3 для SQLite или psycopg2 для PostgreSQL). Обмен данными через API: понимание концепции API, работа с HTTP-запросами (например, с помощью библиотеки requests), обработка ответов от API. Сериализация и десериализация: преобразование объектов Python в формат данных и обратно.

Практические занятия /лабораторные работы организуются, в том числе, в форме практической подготовки, которая предусматривает участие обучающихся в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью.
На проведение практических занятий / лабораторных работ в форме практической подготовки отводится: очно-заочная форма обучения - 10 ч.

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа обучающихся включает в себя подготовку к контрольным вопросам и заданиям для текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведенным в п. 5.

Для обеспечения самостоятельной работы обучающихся используются:
Электронные курсы, созданные в системе электронного обучения ННГУ:

Алгоритмизация и программирование, https://e-learning.unn.ru/course/view.php?id=2371.

Иные учебно-методические материалы:
Учебно-методические документы, регламентирующие самостоятельную работу, адреса доступа к документам:
https://arz.unn.ru/sveden/document/
https://arz.unn.ru/pdf/Metod_all_all.pdf

5. Фонд оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)

5.1 Типовые задания, необходимые для оценки результатов обучения при проведении текущего контроля успеваемости с указанием критериев их оценивания:

5.1.1 Типовые задания (оценочное средство - Практическое задание) для оценки сформированности компетенции ОПК-3:

1. "Привет, мир!" с переменной:

- Задача: Создайте программу, которая запрашивает у пользователя его имя. Затем программа должна поздороваться с пользователем, используя введенное имя.

- Цель: Закрепить навыки работы с вводом/выводом данных и переменными.

- Пример (псевдокод):

Ввод: имя_пользователя

Вывод: "Привет, " + имя_пользователя + "!"

2. Калькулятор простых арифметических операций:

- Задача: Разработайте программу, которая запрашивает у пользователя два числа и операцию (+, -, *, /). Выполните указанную операцию над числами и выведите результат. Обработайте случай деления на ноль.

- Цель: Закрепить навыки работы с арифметическими операторами, условными операторами (if/else) и обработкой исключений (деление на ноль).

- Пример (псевдокод):

Ввод: число1, число2, операция

Если операция == "+":

Результат = число1 + число2

Иначе если операция == "-":

Результат = число1 - число2

Иначе если операция == "×":

Результат = число1 × число2

Иначе если операция == "/":

Если число2 == 0:

Вывод: "Ошибка: деление на ноль!"

Иначе:

Результат = число1 / число2

Вывод: Результат

3. Определение большего из двух чисел:

- Задача: Напишите программу, которая запрашивает у пользователя два числа и определяет, какое из них больше. Выведите сообщение с указанием большего числа. Если числа равны, выведите соответствующее сообщение.

- Цель: Закрепить навыки работы с условными операторами (if/else) и операторами сравнения.

- Пример (псевдокод):

```
Ввод: число1, число2
Если число1 > число2:
  Вывод: "Число " + число1 + " больше, чем число " + число2
Иначе если число2 > число1:
  Вывод: "Число " + число2 + " больше, чем число " + число1
Иначе:
  Вывод: "Числа равны"
```

4. Вывод чисел от 1 до N:

- Задача: Создайте программу, которая запрашивает у пользователя число N. Программа должна вывести все числа от 1 до N (включительно) в столбик.

- Цель: Закрепить навыки работы с циклами (for или while).

- Пример (псевдокод):

```
Ввод: N
Для i от 1 до N:
  Вывод: i
```

5. Игра "Угадай число":

- Задача: Разработайте программу, которая генерирует случайное число в диапазоне от 1 до 100. Пользователь должен угадать это число. Программа должна давать подсказки "Больше" или "Меньше" после каждой попытки. Когда пользователь угадывает число, программа должна сообщить, сколько попыток ему потребовалось.

- Цель: Закрепить навыки работы с циклами (while), условными операторами (if/else), случайными числами и вводом/выводом данных.

- Пример (псевдокод):

```
Случайное число = Сгенерировать случайное число от 1 до 100
Попытки = 0
Пока Не угадал:
  Ввод: Предположение
  Попытки = Попытки + 1
  Если Предположение < Случайное число:
    Вывод: "Больше"
  Иначе если Предположение > Случайное число:
    Вывод: "Меньше"
  Иначе:
    Вывод: "Вы угадали! Число попыток: " + Попытки
    Угадал = Истина
```

5.1.2 Типовые задания (оценочное средство - Практическое задание) для оценки сформированности компетенции ОПК-7:

1. Конвертер температуры:

Задача: Напишите программу на Python, которая запрашивает у пользователя температуру в градусах Цельсия и преобразует ее в градусы Фаренгейта. Выведите результат с пояснением.

Формула: Фаренгейт = (Цельсий × 9/5) + 32

Пример:

```
celsius = float(input("Введите температуру в градусах Цельсия: "))
fahrenheit = (celsius * 9/5) + 32
print(f"{celsius} градусов Цельсия равны {fahrenheit} градусам Фаренгейта.")
```

2. Проверка четности числа:

Задача: Создайте программу на Python, которая запрашивает у пользователя целое число и определяет, является ли оно четным или нечетным. Выведите соответствующее сообщение.

Использование оператора % (modulo): Число четное, если остаток от деления на 2 равен 0.

Пример:

```
number = int(input("Введите целое число: "))
if number % 2 == 0:
    print(f"{number} - четное число.")
else:
    print(f"{number} - нечетное число.")
```

3. Подсчет гласных в строке:

Задача: Напишите программу на Python, которая запрашивает у пользователя строку и подсчитывает количество гласных букв (a, e, i, o, u) в этой строке. Буквы могут быть как в верхнем, так и в нижнем регистре.

Использование циклов и строковых методов:

Пример:

```
string = input("Введите строку: ").lower() # Преобразуем строку в нижний регистр
vowels = "aeiou"
count = 0
```

```
for char in string:
    if char in vowels:
        count += 1
print(f"Количество гласных букв в строке: {count}")
```

4. Факториал числа:

Задача: Разработайте программу на Python, которая запрашивает у пользователя неотрицательное целое число и вычисляет его факториал.

Факториал (n!): Произведение всех целых чисел от 1 до n. Например, 5! = 1 × 2 × 3 × 4 × 5 = 120.

Обработка крайнего случая (0! = 1):

Пример (с использованием цикла):

```
number = int(input("Введите неотрицательное целое число: "))
if number < 0:
    print("Факториал определен только для неотрицательных чисел.")
elif number == 0:
    print("Факториал 0 равен 1")
else:
    factorial = 1
    for i in range(1, number + 1):
        factorial *= i
    print(f"Факториал числа {number} равен {factorial}")
```

5. Простая игра "Камень, ножницы, бумага":

Задача: Реализуйте простую версию игры "Камень, ножницы, бумага" против компьютера.

Шаги:

1. Компьютер случайным образом выбирает один из вариантов ("камень", "ножницы", "бумага").

2. Пользователь вводит свой выбор.

3. Программа определяет победителя по правилам игры и выводит результат.

Использование модуля random для выбора компьютером:

Пример (фрагмент):

```
import random

choices = ["камень", "ножницы", "бумага"]
 computer_choice = random.choice(choices)

user_choice = input("Выберите: камень, ножницы или бумага: ").lower()

print(f"Компьютер выбрал: {computer_choice}")
print(f"Вы выбрали: {user_choice}")

if user_choice == computer_choice:
    print("Ничья!")
elif (user_choice == "камень" and computer_choice == "ножницы") or \
     (user_choice == "ножницы" and computer_choice == "бумага") or \
     (user_choice == "бумага" and computer_choice == "камень"):
   print("Вы победили!")
else:
```

```
print("Компьютер победил!")
```

Критерии оценивания (оценочное средство - Практическое задание)

Оценка	Критерии оценивания
зачтено	Задание выполнено полностью и правильно на основании изученной теории; теоретический материал и решение поставленных задач изложены в необходимой логической последовательности, грамотный научный язык; задание выполнено самостоятельно. Могут быть допущены две–три несущественные ошибки, исправленные по требованию преподавателя.
не зачтено	Задание обнаруживает непонимание студентом основного содержания учебного материала или допущены существенные ошибки, которые не могут быть исправлены при наводящих вопросах преподавателя.

5.1.3 Типовые задания (оценочное средство - Опрос) для оценки сформированности компетенции ОПК-3:

1. Что такое интерпретируемый язык программирования, и является ли Python таким?

2. Объясните разницу между == и is в Python.

3. Перечислите основные типы данных в Python.

4. Как создать список в Python и добавить в него элемент?

5. Что такое индексация в Python и как она работает со списками?

6. Как использовать цикл for для перебора элементов списка?

7. Объясните разницу между списками и кортежами.

8. Как определить функцию в Python и передать ей аргументы?

9. Что такое область видимости переменных (scope) в Python?

10. Что делает оператор in в Python?

5.1.4 Типовые задания (оценочное средство - Опрос) для оценки сформированности компетенции ОПК-7:

11. Объясните, что такое исключения и как их обрабатывать в Python с помощью try...except.

12. Как открыть и закрыть файл в Python, используя менеджер контекста with?

13. Что такое модуль в Python и как импортировать модуль?

14. Как создать и использовать словарь в Python?

15. Что такое list comprehension и как он используется?

16. Объясните концепцию итераторов и генераторов в Python.

17. Что такое декораторы и как они используются в Python?

18. Объясните разницу между многопоточностью и многопроцессорностью в Python.

19. Что такое объектно-ориентированное программирование, и какие основные принципы ООП вы знаете?

20. Напишите короткий фрагмент кода, демонстрирующий наследование классов в Python.

Критерии оценивания (оценочное средство - Опрос)

Оценка	Критерии оценивания
зачтено	Ответ полный и правильный на основании изученной теории; теоретический материал и решение поставленных задач изложены в необходимой логической последовательности, грамотный научный язык; ответ самостоятельный. Могут быть допущены две-три несущественные ошибки, исправленные по требованию преподавателя.
не зачтено	Ответ обнаруживает непонимание студентом основного содержания учебного материала или допущены существенные ошибки, которые не могут быть исправлены при наводящих вопросах преподавателя.

5.2. Описание шкал оценивания результатов обучения по дисциплине при промежуточной аттестации

Шкала оценивания сформированности компетенций

Уровень сформированности компетенций (индикатора достижения компетенций)	**неудовлетворительно**	**удовлетворительно**	**хорошо**	**отлично**
	не зачтено	**зачтено**		
Знания	Уровень знаний ниже минимальных требований. Имели место грубые ошибки	Минимально допустимый уровень знаний. Допущено много негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки, без ошибок
Умения	При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки	Продемонстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме	Продемонстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания в полном объеме, но некоторые с недочетами	Продемонстрированы все основные умения. Решены все основные задачи с отдельными несущественными недочетами, выполнены все задания в полном объеме
Навыки	При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые ошибки	Имеется минимальный набор навыков для решения стандартных задач с некоторыми недочетами	Продемонстрированы базовые навыки при решении стандартных задач с некоторыми недочетами	Продемонстрированы навыки при решении нестандартных задач без ошибок и недочетов

Шкала оценивания при промежуточной аттестации

Оценка		**Уровень подготовки**
зачтено	**отлично**	Все компетенции (части компетенций), на формирование которых направлена

		дисциплина, сформированы на уровне не ниже «отлично», при этом хотя бы одна компетенция сформирована на уровне «отлично»
	хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо», при этом хотя бы одна компетенция сформирована на уровне «хорошо»
	удовлетворительно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно»
не зачтено	**неудовлетворительно**	Хотя бы одна компетенция сформирована на уровне «неудовлетворительно».

5.3 Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения на промежуточной аттестации с указанием критериев их оценивания:

5.3.1 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ОПК-3

1. Операторы ввода–вывода. Проектирование ввода–вывода информации.
2. Условный оператор и оператор варианта. Пример с использованием блок-схемы.
3. Составной оператор. Пример с использованием блок-схемы.
4. Организация циклов. Блок-схемы. Вложенные циклы; правила работы с вложенными циклами.
5. Списки.
6. Типы данных.

5.3.2 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ОПК-7

1. Линейный алгоритм
2. Разветвляющийся алгоритм
3. Циклический алгоритм
4. Работа с кортежами и списками
5. Работа со строками
6. Обработка вложенных последовательностей
7. Работа с функциями. Создание модулей
8. Работа с файлами
9. Объектно-ориентированное программирование

Критерии оценивания (оценочное средство - Контрольные вопросы)

Оценка	Критерии оценивания
зачтено	ответ полный и правильный на основании изученной теории; теоретический материал и решение поставленных задач изложены в необходимой логической последовательности, грамотный научный язык; ответ самостоятельный. Могут быть допущены две–три несущественные ошибки, исправленные по требованию преподавателя.
не зачтено	ответ полный и правильный на основании изученной теории; теоретический материал и решение поставленных задач изложены в необходимой логической последовательности, грамотный научный язык; ответ самостоятельный. Могут быть допущены две–три несущественные ошибки, исправленные по требованию преподавателя.

5.3.3 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ОПК-3

1. Типы данных в Python (int, float, str, bool, list, tuple, dict, set).

2. Операторы присваивания, арифметические, сравнения, логические, битовые.

3. Условные операторы (if, elif, else).

4. Циклы (for, while).

5. Ввод и вывод данных (input(), print()).

6. Индексация и срезы (slices) в строках и списках.

7. Функции: определение, параметры, возвращаемые значения, область видимости.

8. Модули и импорт модулей.

5.3.4 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ОПК-7

1. Обработка исключений (try, except, finally).
2. Работа с файлами (чтение, запись, открытие/закрытие).
3. List comprehensions.
4. Lambda-функции.
5. Работа со словарями (добавление, удаление, изменение элементов, итерация).
6. Объектно-ориентированное программирование (классы, объекты, наследование, полиморфизм, инкапсуляция).
7. Итераторы и генераторы.
8. Декораторы.
9. Многопоточность или многопроцессорность (основы).
10. Работа с базами данных (например, SQLite).
11. Сериализация и десериализация данных (JSON, Pickle).

Критерии оценивания (оценочное средство - Контрольные вопросы)

Оценка	Критерии оценивания
отлично	выставляется, когда студент глубоко и прочно усвоил весь программный материал, исчерпывающе, последовательно, грамотно и логически стройно его излагает, не затрудняется с ответом при видоизменении задания, свободно справляется с ситуационными заданиями, правильно обосновывает принятые решения, умеет самостоятельно обобщать и излагать материал, не допуская ошибок.
хорошо	выставляется, если студент твердо знает программный материал, грамотно и по существу излагает его, не допускает существенных неточностей в ответе на вопрос, может правильно применять теоретические положения и владеет необходимыми умениями и навыками при анализе информации.
удовлетворительно	выставляется в том случае, при котором студент освоил только основной материал, но не знает отдельных деталей, допускает неточности,

Оценка	Критерии оценивания
	недостаточно правильные формулировки, нарушает последовательность в изложении программного материала и испытывает затруднения в выполнении анализа информации.
неудовлетворительно	выставляется студенту, в ответе которого обнаружились существенные пробелы в знании основного содержания учебной программы дисциплины и / или неумение использовать полученные знания.

6. Учебно-методическое и информационное обеспечение дисциплины (модуля)

Основная литература:

1. Василекина О. М. Учебно-методическое пособие по дисциплине «Алгоритмизация и программирование»: Структурное и процедурное программирование на языке Python направление подготовки 09.03.03 Прикладная информатика профиль «Прикладная информатика в экономике» / Василекина О. М. - Великие Луки : Великолукская ГСХА, 2024. - 104 с. - Книга из коллекции Великолукская ГСХА - Информатика., https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=918825&idb=0.
2. Гуриков Сергей Ростиславович (Московский технический университет связи и информатики). Основы алгоритмизации и программирования на Python : Учебное пособие / Московский технический университет связи и информатики. - 1. - Москва : ООО "Научно-издательский центр ИНФРА-М", 2025. - 343 с. - (Высшее образование). - ВО - Бакалавриат. - ISBN 978-5-16-020255-6. - ISBN 978-5-16-102278-8 (электр. издание)., https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=933203&idb=0.
3. Федоров Дмитрий Юрьевич. Программирование на python : учебное пособие для вузов / Д. Ю. Федоров. - 6-е изд. - Москва : Юрайт, 2025. - 187 с. - (Высшее образование). - URL: https://urait.ru/bcode/556864 (дата обращения: 15.08.2024). - ISBN 978-5-534-19666-5 : 719.00. - Текст : электронный // ЭБС "Юрайт"., https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=908071&idb=0.
4. Басев И. Н. Введение в язык программирования Python : электронное учебно-методическое пособие / Басев И. Н.,Голунова Л. В.,Функ А. В. - Новосибирск : СГУПС, 2024. - 65 с. - Книга из коллекции СГУПС - Информатика. - ISBN 978-5-00148-441-7., https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=976447&idb=0.
5. Канева О. Н. Введение в программирование на языке Python : учеб. пособие / Канева О. Н.,Финк Т. Ю. - Омск : ОмГТУ, 2024. - 149 с. - Книга из коллекции ОмГТУ - Информатика. - ISBN 978 5 8149 3864 0., https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=978571&idb=0.

Дополнительная литература:

1. Щербаков А.Г. Практикум изучения языка программирования PYTHON. Начальный уровень : Учебное пособие / А.Г. Щербаков. - Москва : Русайнс, 2024. - 116 с. - Режим доступа: book.ru. - ISBN 978-5-466-07049-1., https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=901134&idb=0.

2. Чернышев Станислав Андреевич. Основы программирования на Python : учебное пособие для вузов / С. А. Чернышев. - 2-е изд. - Москва : Юрайт, 2024. - 349 с. - (Высшее образование). - URL: https://urait.ru/bcode/544190 (дата обращения: 15.08.2024). - ISBN 978-5-534-17139-6 : 1499.00. - Текст : электронный // ЭБС "Юрайт"., https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=901582&idb=0.
3. Апанасевич С. А. Структуры и алгоритмы обработки данных. Линейные структуры : учебное пособие для вузов / Апанасевич С. А. - 2-е изд., стер. - Санкт-Петербург : Лань, 2025. - 136 с. - Книга из коллекции Лань - Информатика. - ISBN 978-5-507-53508-8., https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=975818&idb=0.

Программное обеспечение и Интернет-ресурсы (в соответствии с содержанием дисциплины):

в) программное обеспечение и Интернет-ресурсы:

Лицензионное программное обеспечение: Операционная система Windows.
Лицензионное программное обеспечение: Microsoft Office.

Профессиональные базы данных и информационные справочные системы

Российский индекс научного цитирования (РИНЦ), платформа Elibrary: национальная информационно-аналитическая система. Адрес доступа: http://elibrary.ru/project_risc.asp

MathSciNet: информационно-библиографическая и реферативная база данных по математике, в т.ч. прикладной математике и статистике. Электронная версия Mathematical Reviews. Адрес доступа: http://www.ams.org/mathscinet

Свободно распространяемое программное обеспечение:
программное обеспечение 1С:
* "Бухгалтерия предприятия", редакция 3.0, см. http://v8.1c.ru/buhv8/ ,
* "Управление торговлей", редакция 11.1, см. http://v8.1c.ru/trade/ ,
* "Зарплата и управление персоналом", редакция 3.0, см. http://v8.1c.ru/hrm/ ,
* "Управление небольшой фирмой", редакция 1.5, см. http://v8.1c.ru/small.biz/ ,
* "ERP Управление предприятием 2.0", см. http://v8.1c.ru/erp/ .
* "Бухгалтерия государственного учреждения", редакция 1.0, см. http://v8.1c.ru/stateacc/ ,
* "Зарплата и кадры государственного учреждения", редакция 1.0, http://v8.1c.ru/statehrm/ .

программное обеспечение Python IDLE:

Электронные библиотечные системы и библиотеки:
Электронная библиотечная система "Лань" https://e.lanbook.com/
Электронная библиотечная система "Консультант студента" http://www.studentlibrary.ru/
Электронная библиотечная система "Юрайт"http://www.urait.ru/ebs
Электронная библиотечная система "Znanium" http://znanium.com/
Электронно-библиотечная система Университетская библиотека ONLINE http://biblioclub.ru/
Фундаментальная библиотека ННГУ www.lib.unn.ru/
Сайт библиотеки Арзамасского филиала ННГУ. – Адрес доступа: lib.arz.unn.ru

Ресурс «Массовые открытые онлайн-курсы Нижегородского университета им. Н.И. Лобачевского» https://mooc.unn.ru/
Портал «Современная цифровая образовательная среда Российской Федерации» https://online.edu.ru/public/promo

7. Материально-техническое обеспечение дисциплины (модуля)

Учебные аудитории для проведения учебных занятий, предусмотренных образовательной программой, оснащены мультимедийным оборудованием (проектор, экран), техническими средствами обучения, компьютерами.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду.

Программа составлена в соответствии с требованиями ОС ННГУ по направлению подготовки/специальности 09.03.03 - Прикладная информатика.

Автор(ы): Первушкина Елена Александровна, кандидат педагогических наук, доцент.

Рецензент(ы): Статуев Алексей Анатольевич, кандидат педагогических наук.

Заведующий кафедрой: Нестерова Лариса Юрьевна, кандидат педагогических наук.

Программа одобрена на заседании методической комиссии от 10.12.2025, протокол № 10.