

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

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

Институт информационных технологий, математики и механики

УТВЕРЖДЕНО
решением Ученого совета ННГУ
протокол № 10 от 02.12.2024 г.

Рабочая программа дисциплины

Python и автоматизация

Уровень высшего образования
Бакалавриат

Направление подготовки / специальность
02.03.02 - Фундаментальная информатика и информационные технологии

Направленность образовательной программы
Сопряженная разработка программного и аппаратного обеспечения

Форма обучения
очная

г. Нижний Новгород

2025 год начала подготовки

1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.ДВ.01.01 Python и автоматизация относится к части, формируемой участниками образовательных отношений 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
ПК-4: Способен проектировать программное обеспечение	<p>ПК-4.1: Знает типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения</p> <p>ПК-4.2: Знает методы и средства проектирования программного обеспечения</p> <p>ПК-4.3: Знает методы и средства проектирования баз данных</p> <p>ПК-4.4: Умеет использовать существующие типовые решения и шаблоны проектирования программного обеспечения</p> <p>ПК-4.5: Умеет применять методы и средства проектирования программного обеспечения, структур данных, баз данных</p>	<p>ПК-4.1:</p> <p>Знать:</p> <ul style="list-style-type: none"> - типовые библиотеки Python для каждой задачи: requests, pandas, re, click, argparse и др.; - основные шаблоны проектирования (Factory, Singleton, Observer, Strategy) и их реализацию в Python; - встроенные структуры данных и их характеристики. <p>Уметь:</p> <ul style="list-style-type: none"> - выбирать готовую библиотеку вместо написания собственного кода; - применять шаблон проектирования для решения повторяющейся задачи; - интегрировать сторонние модули через pip/poetry. <p>Владеть:</p> <ul style="list-style-type: none"> - навыком быстрой оценки библиотеки (документация, лицензия, звёзды на GitHub, свежесть релиза); - практикой рефакторинга «сырого» кода с введением фабрики или стратегии; - использованием typing и протоколов для описания интерфейсов. <p>ПК-4.2:</p> <p>Знать</p> <ul style="list-style-type: none"> - методологии (Agile, Waterfall), принципы SOLID, 	Тест	<p>Зачёт:</p> <p>Контрольные вопросы</p>

		<p>GRASP, KISS, DRY, YAGNI;</p> <ul style="list-style-type: none">- средства моделирования: UML (диаграммы классов, последовательности, состояний), C4 model;- понятия архитектурных стилей (монолит, микросервисы, слоистая архитектура). <p>Уметь:</p> <ul style="list-style-type: none">- разрабатывать диаграмму классов для предметной области;- декомпозировать задачу на модули и классы с учётом слабой связанности;- проектировать расширяемые системы через абстракции и внедрение зависимостей. <p>Владеть:</p> <ul style="list-style-type: none">- навыком создания UML-диаграмм (PlantUML, draw.io) в составе документации;- практикой проведения архитектурного code review;- использованием архитектурных паттернов (MVC, Repository, Unit of Work) в Python. <p>ПК-4.3:</p> <p>Знать:</p> <ul style="list-style-type: none">- нормальные формы (1НФ–3НФ), типы связей (1:1, 1:N, M:N), первичные/внешние ключи, индексы;- транзакции, уровни изоляции, ACID;- основы NoSQL (документные БД, ключ-значение, column-family);- средства проектирования: ER-диаграммы, схемы миграций. <p>Уметь:</p> <ul style="list-style-type: none">- создавать логическую и физическую модель БД по требованиям;- писать оптимизированные SQL-запросы (JOIN, GROUP BY, подзапросы, оконные функции);- выбирать между SQL и		
--	--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--

		<p>NoSQL для конкретной задачи.</p> <p>Владеть:</p> <ul style="list-style-type: none">- навыком создания ER-диаграмм (например, в <i>dbdiagram.io</i>, <i>draw.io</i>);- практикой написания миграций (<i>Alembic</i> для <i>SQLAlchemy</i>, <i>Flyway</i>);- использованием <i>EXPLAIN</i> для анализа и оптимизации запросов. <p>ПК-4.4:</p> <p>Знать:</p> <ul style="list-style-type: none">- когда какой шаблон применять (порождающие – для создания объектов, структурные – для интерфейсов, поведенческие – для алгоритмов);- готовые библиотеки-решения: <i>SQLAlchemy (ORM)</i>, <i>pytest (тестирование)</i>, <i>loguru (логирование)</i>, <i>pydantic (валидация)</i>. <p>Уметь:</p> <ul style="list-style-type: none">- заменить самописный велосипед на вызов стандартной библиотеки или популярного пакета;- реализовать фабрику, адаптер, декоратор, наблюдатель в рабочем коде;- настроить <i>pre-commit</i> с линтерами и форматтерами. <p>Владеть:</p> <ul style="list-style-type: none">-навыком рефакторинга <i>legasy</i>-кода с введением паттернов;-практикой интеграции готовых решений (например, подключить <i>requests-retry</i> вместо ручных повторов);- использованием шаблона <i>Repository</i> для абстракции доступа к данным. <p>ПК-4.5:</p> <p>Знать:</p> <ul style="list-style-type: none">- соответствие между задачами и структурами данных: хеш-таблица для быстрого поиска, стек для		
--	--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--

		<p>парсинга, очередь для задач;</p> <ul style="list-style-type: none"> - методы нормализации и денормализации в БД; - принципы проектирования API (ресурсы, идиоматичность, версионирование). <p>Уметь:</p> <ul style="list-style-type: none"> - проектировать классы и одновременно схему БД так, чтобы они согласовывались (ORM-маппинг, агрегаты); - выбирать структуру данных для минимизации временной сложности ($O(1)$ vs $O(n)$); - применять индексы в БД для ускорения запросов, которые генерирует приложение. <p>Владеть</p> <ul style="list-style-type: none"> - навыком создания целостного проекта: требования → модель классов → схема БД → API endpoints → тесты; - практикой оптимизации запросов с использованием EXPLAIN и профилирования кода; - использованием диаграмм (классов, ER, последовательности) для передачи проектных решений команде. 		
--	--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--

3. Структура и содержание дисциплины

3.1 Трудоемкость дисциплины

	очная
Общая трудоемкость, з.е.	2
Часов по учебному плану	72
в том числе	
аудиторные занятия (контактная работа):	
- занятия лекционного типа	16
- занятия семинарского типа (практические занятия / лабораторные работы)	16
- КСР	1
самостоятельная работа	39
Промежуточная аттестация	0
	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	
0 Ф 0	0 Ф 0	0 Ф 0	0 Ф 0	0 Ф 0	
Тема 1. Основы Python	8	2	2	4	4
Тема 2. Объектно-ориентированное программирование	8	2	2	4	4
Тема 3. Стандартная библиотека Python	8	2	2	4	4
Тема 4. Обработка текстов и регулярные выражения	8	2	2	4	4
Тема 5. Работа с API и веб-сервисами	8	2	2	4	4
Тема 6. Автоматизация задач разработки	8	2	2	4	4
Тема 7. Командная строка и CLI-инструменты	8	2	2	4	4
Тема 8. Работа с данными: подготовка к ML	7	1	1	2	5
Тема 9. Best practices	8	1	1	2	6
Аттестация	0				
КСР	1			1	
Итого	72	16	16	33	39

Содержание разделов и тем дисциплины

1. Основы Python

Тема 1.1. Введение в Python

- Интерпретатор Python: CPython, PyPy
- Виртуальные окружения: venv, virtualenv
- Управление пакетами: pip, requirements.txt
- Основные типы данных: числа, строки, списки, словари, множества
- Особенности Python: динамическая типизация, управление памятью
- Практическая работа: настройка окружения, решение базовых задач

Тема 1.2. Управляющие конструкции и функции

- Условия и циклы: if, for, while
- List comprehensions и generator expressions
- Функции: определение, аргументы, *args и **kwargs
- Lambda-функции
- Декораторы: синтаксис и применение
- Практическая работа: написание функций для обработки данных

2. Объектно-ориентированное программирование

Тема 2.1. Классы и объекты

- Определение классов и создание объектов
- Атрибуты и методы
- Конструктор `__init__` и специальные методы
- Инкапсуляция: приватные атрибуты
- Наследование и полиморфизм
- Практическая работа: создание иерархии классов

Тема 2.2. Продвинутое возможности

- Property и дескрипторы
- Статические методы и методы класса
- Магические методы: `__str__`, `__repr__`, `__eq__`
- Контекстные менеджеры: `with statement`
- Практическая работа: реализация классов с расширенной функциональностью

3. Стандартная библиотека Python

Тема 3.1. Работа с файловой системой

- Модули `os` и `os.path`
- `pathlib` для современной работы с путями
- Чтение и запись файлов: открытие, режимы, кодировки
- Работа с директориями: создание, удаление, обход
- `glob` для поиска файлов по шаблону
- `shutil` для копирования и перемещения
- Практическая работа: скрипты для массовой обработки файлов

Тема 3.2. Работа с внешними процессами

- `subprocess`: запуск внешних программ
- Перехват вывода: `stdout`, `stderr`
- Передача данных в процессы через `stdin`
- Управление кодами возврата
- `concurrent.futures` для параллельного выполнения
- Практическая работа: автоматизация запуска утилит, обработка результатов

Тема 3.3. Полезные модули стандартной библиотеки

- `json` для работы с JSON
- `csv` для табличных данных
- `datetime` для работы с датой и временем
- `argparse` для разбора аргументов командной строки
- `logging` для ведения логов
- `configparser` для конфигурационных файлов

4. Обработка текстов и регулярные выражения

Тема 4.1. Работа со строками

- Методы строк: split, join, strip, replace
- Форматирование строк: f-strings, format, %
- Unicode и кодировки
- Практическая работа: парсинг и преобразование текстовых данных

Тема 4.2. Регулярные выражения

- Модуль re: search, match, findall, sub
- Синтаксис регулярных выражений
- Группы и захват подстрок
- Жадные и ленивые квантификаторы
- Практическая работа: извлечение данных из логов, валидация форматов

5. Работа с API и веб-сервисами

Тема 5.1. HTTP-запросы с requests

- Установка и основы библиотеки requests
- GET, POST, PUT, DELETE запросы
- Параметры запросов и заголовки
- Обработка JSON-ответов
- Аутентификация: Basic, Bearer tokens
- Обработка ошибок и таймаутов
- Практическая работа: взаимодействие с REST API (GitHub, публичные API)

Тема 5.2. Парсинг HTML

- BeautifulSoup для извлечения данных из HTML
- Поиск элементов: find, find_all, CSS selectors
- lxml как альтернатива
- Этика веб-скрапинга
- Практика: сбор данных с веб-страниц

6. Автоматизация задач разработки

Тема 6.1. Скрипты для сборки и развёртывания

- Автоматизация сборки проектов
- Управление зависимостями
- Создание архивов и пакетов
- Fabric для удалённого развёртывания
- invoke для задач автоматизации
- Практическая работа: скрипт для автоматизированного развёртывания

Тема 6.2. Автоматизация тестирования

- unittest: структура тестов, assertions
- pytest: fixtures, параметризация
- Mock-объекты для изоляции тестов
- Coverage для измерения покрытия

- Практическая работа: написание тестов для скриптов

Тема 6.3. Обработка логов и мониторинг

- Парсинг лог-файлов различных форматов
- Агрегация и статистика
- Поиск аномалий и ошибок
- Генерация отчётов
- Отправка уведомлений: email, Slack, Telegram
- Практическая работа: анализ логов веб-сервера, создание dashboard

7. Командная строка и CLI-инструменты

Тема 7.1. Создание CLI-приложений

- argparse для сложных CLI
- click как альтернатива
- Интерактивные prompt с input
- Цветной вывод: colorama, termcolor
- Практика: создание утилиты командной строки

Тема 7.2. Интеграция в shell

- Создание исполняемых скриптов: shebang
- Коды возврата и обработка ошибок
- Работа со стандартными потоками: stdin, stdout, stderr
- Pipes и цепочки команд

8. Работа с данными: подготовка к ML

Тема 8.1. NumPy для численных вычислений

- Массивы NumPy и их преимущества
- Векторизованные операции
- Индексирование и срезы
- Математические функции
- Broadcasting
- Практическая работа: обработка числовых данных

Тема 8.2. Pandas для анализа данных

- DataFrame и Series
- Загрузка данных: CSV, JSON, Excel
- Фильтрация и группировка
- Агрегация и pivot tables
- Обработка пропущенных значений
- Объединение датафреймов: merge, join
- Практическая работа: анализ реальных датасетов, подготовка данных

9. Best practices

Тема 9.1. Качество кода

- PEP 8 и style guides
- Type hints и статическая типизация: mypy
- Docstrings и документация
- Code review practices
- Практическая работа: рефакторинг кода с улучшением качества

Тема 9.2. Обработка ошибок и отладка

- Try-except-finally для обработки исключений
- Создание пользовательских исключений
- Отладка: pdb, ipdb
- Профилирование: cProfile, line_profiler
- Практическая работа: отладка и оптимизация скриптов

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.

Самостоятельная работа студентов при изучении дисциплины проводится с целью углубления знаний по дисциплине и предусматривает:

- повторение пройденного учебного материала, чтение рекомендованной литературы;
- подготовку к практическим занятиям;
- выполнение общих и индивидуальных домашних заданий;
- работу с электронными источниками;
- подготовку к сдаче формы промежуточной аттестации.

Цель самостоятельной работы - формирование навыков непрерывного самообразования и профессионального совершенствования.

Самостоятельная работа способствует формированию аналитического и творческого мышления, совершенствует способы организации исследовательской деятельности, воспитывает целеустремленность, системность и последовательность в работе студентов, развивает у них навык завершать начатую работу.

Основные виды самостоятельной работы студентов:

- работа с основной и дополнительной литературой;
- изучение категориального аппарата дисциплины;
- самостоятельное изучение тем дисциплины;
- подготовка к зачёту;
- работа в библиотеке;
- изучение сайтов по темам дисциплины в сети Интернет.

Работа с основной и дополнительной литературой

Изучение рекомендованной литературы следует начинать с учебников и учебных пособий, затем переходить к научным монографиям и материалам периодических изданий. Работа с литературой предусматривает конспектирование наиболее актуальных и познавательных

материалов. Это не только мобилизует внимание, но и способствует более глубокому осмыслению материала, его лучшему запоминанию, а также позволяет студентам проводить систематизацию и сравнительный анализ изучаемой информации. Таким образом, конспектирование – одна из основных форм самостоятельного труда, которая требует от студента активно работать с учебной литературой и не ограничиваться конспектом лекций. Студент должен уметь самостоятельно подбирать необходимую литературу для учебной и научной работы, уметь обращаться с предметными каталогами и библиографическим справочником библиотеки.

Изучение категориального аппарата дисциплины

Изучение и осмысление экономических категорий требует проработки лекционного материала, выполнения практических заданий, изучение словарей, энциклопедий, справочников.

Индивидуальная самостоятельная работа студента направлена на овладение и грамотное применение экономической терминологии в области компьютерного моделирования.

Самостоятельное изучение тем дисциплины

Особое место отводится самостоятельной проработке студентами отдельных разделов и тем изучаемой дисциплины. Такой подход вырабатывает у студентов инициативу, стремление к увеличению объема знаний, умений и навыков, всестороннего овладения способами и приемами профессиональной деятельности.

Изучение вопросов определенной темы направлено на более глубокое усвоение основных категорий экономической теории, понимание экономических процессов, происходящих в обществе, совершенствование навыка анализа теоретического и эмпирического материала.

Подготовка к зачёту

Промежуточная аттестация студентов по дисциплине проходит в виде зачёта. Условием успешного прохождения промежуточной аттестации является систематическая работа студента в течение семестра. В этом случае подготовка к зачёту является систематизацией всех полученных знаний по данной дисциплине.

Рекомендуется внимательно изучить перечень вопросов к зачёту также использовать в процессе обучения программу, учебно-методический комплекс, другие методические материалы.

Желательно спланировать троекратный просмотр материала перед зачётом. Во-первых, внимательное чтение с осмыслением, подчеркиванием и составлением краткого плана ответа. Во-вторых, повторная проработка наиболее сложных вопросов. В-третьих, быстрый просмотр материала или планов ответов для его систематизации в памяти.

Изучение сайтов по темам дисциплины в сети Интернет

Ресурсы Интернет являются одним из альтернативных источников быстрого поиска требуемой информации. Их использование возможно для получения основных и дополнительных сведений по изучаемым материалам. Необходимо помнить об оформлении ссылок на Интернет-источники.

Для повышения эффективности самостоятельной работы студентов преподавателю целесообразно использовать следующие виды деятельности:

- консультации,
- выдача заданий на самостоятельную работу,
- информационное обеспечение обучения,
- контроль качества самостоятельной работы студентов.

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 - Тест) для оценки сформированности компетенции ПК-4:

1. Какой оператор используется для возведения в степень?

- a) ^
- b) **
- c) //
- d) %

2. Как создать пустой словарь?

- a) dict()
- b) {}
- c) []
- d) set()

Что выведет `print(3 * 'abc')`?

- a) abcabcabc
- b) 3abc
- c) Ошибку
- d) abc

4. Как получить последний элемент списка?

- a) `list[-1]`
- b) `list.last()`
- c) `list.end()`
- d) `list[last]`

5. Какой метод добавляет элемент в список?

a) add()

b) append()

c) insert()

d) push()

6. Что делает функция range(5)?

a) Создает список [0,1,2,3,4]

b) Создает генератор чисел от 0 до 4

c) Возвращает 5

d) Ошибку

7. Как открыть файл для чтения?

a) open(file, 'r')

b) open(file, 'read')

c) open(file, 'w')

d) open(file)

8. Как создать класс?

a) class MyClass

b) def MyClass()

c) new class MyClass

d) class MyClass()

9. Как импортировать модуль math?

a) include math

b) import math

c) using math

d) require math

10. Что делает метод pandas.read_csv()?

a) Читает Excel файл

- b) Читает CSV файл
- c) Создает CSV файл
- d) Записывает в CSV

11. Как создать виртуальное окружение?

- a) `python -m venv env`
- b) `python create env`
- c) `virtual env`
- d) `new venv env`

12. Какой тип данных неизменяем?

- a) Список
- b) Словарь
- c) Кортеж
- d) Множество

13. Что выведет `print(bool([]))`?

- a) True
- b) False
- c) []
- d) Ошибку

14. Как получить текущую дату?

- a) `datetime.now()`
- b) `time.now()`
- c) `date.current()`
- d) `now()`

15. Какой оператор проверяет идентичность объектов?

- a) `==`
- b) `=`

c) is

d) in

Критерии оценивания (оценочное средство - Тест)

Оценка	Критерии оценивания
превосходно	96-100% правильных ответов
отлично	81-95% правильных ответов
очень хорошо	76-80% правильных ответов
хорошо	61-75% правильных ответов
удовлетворительно	46-60% правильных ответов
неудовлетворительно	31-45% правильных ответов
плохо	30% и меньше правильных ответов

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

Шкала оценивания сформированности компетенций

Уровень сформированности компетенций (индикатора достижения компетенций)	плохо	неудовлетворительно	удовлетворительно	хорошо	очень хорошо	отлично	превосходно
	не зачтено			зачтено			
<u>Знания</u>	Отсутствие знаний теоретического материала. Невозможность оценить полноту знаний вследствие отказа обучающегося от ответа	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. Допущено несколько несущественных ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Ошибок нет.	Уровень знаний в объеме, превышающем программу подготовки.
<u>Умения</u>	Отсутствие минимальных умений. Невозможность оценить наличие умений вследствие	При решении стандартных задач не продемонстрированы основные умения. Имели	Продемонстрированы основные умения. Решены типовые задачи с	Продемонстрированы все основные умения. Решены все основные	Продемонстрированы все основные умения. Решены все основные	Продемонстрированы все основные умения. Решены все	Продемонстрированы все основные умения. Решены все основные задачи.

	отказа обучающегося от ответа	место грубые ошибки	негрубыми ошибками. Выполнены все задания, но не в полном объеме	задачи с негрубыми ошибками. Выполнены все задания в полном объеме, но некоторые с недочетами	задачи. Выполнены все задания в полном объеме, но некоторые с недочетами	основные задачи с отдельным и несущественными недочетами, выполнены все задания в полном объеме	Выполнены все задания, в полном объеме без недочетов
<u>Навыки</u>	Отсутствие базовых навыков. Невозможность оценить наличие навыков вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые ошибки	Имеется минимальный набор навыков для решения стандартных задач с некоторым и недочетами	Продемонстрированы базовые навыки при решении стандартных задач с некоторым и недочетами	Продемонстрированы базовые навыки при решении стандартных задач без ошибок и недочетов	Продемонстрированы навыки при решении нестандартных задач без ошибок и недочетов	Продемонстрирован творческий подход к решению нестандартных задач

Шкала оценивания при промежуточной аттестации

Оценка		Уровень подготовки
зачтено	превосходно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно», продемонстрированы знания, умения, владения по соответствующим компетенциям на уровне выше предусмотренного программой
	отлично	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично».
	очень хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо»
	хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «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».
	плохо	Хотя бы одна компетенция сформирована на уровне «плохо»

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 ПК-4

Основные особенности Python

2. Динамическая типизация в Python
3. Основные структуры данных
4. Операторы ветвления
5. Циклы в Python
6. Создание функций
7. Лямбда-функции
8. Работа со списками
9. Методы словарей
10. Обработка исключений
11. Контекстные менеджеры
12. Принципы ООП
13. Создание классов
14. Наследование в Python
15. Модули и пакеты
16. Установка библиотек
17. Работа с NumPy
18. Pandas DataFrame
19. Визуализация данных
20. Работа с SQLite
21. List comprehensions
22. Генераторы
23. Декораторы
24. Работа с датами
25. Регулярные выражения
26. Многопоточность

27.Работа с API

28.Виртуальные окружения

29.PEP 8

30.Отладка кода

Критерии оценивания (оценочное средство - Контрольные вопросы)

Оценка	Критерии оценивания
зачтено	Вся компетенция (части компетенции), на формирование которой направлена дисциплина, сформированы на уровне не ниже «удовлетворительно»
не зачтено	Хотя бы одна часть компетенции сформирована на уровне «неудовлетворительно»

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

Основная литература:

1. 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. - Текст : электронный., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=976447&idb=0>.
2. GPT-3: программирование на Python в примерах : практическое руководство / Аймен Эль Амри. - Москва : ДМК-пресс, 2023. - 218 с. - ISBN 978-5-93700-221-1. - Текст : электронный., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=1012137&idb=0>.
3. Борзунов С. В. Языки программирования. Python: решение сложных задач : учебное пособие для вузов / Борзунов С. В.,Кургалин С. Д. - 2-е изд., стер. - Санкт-Петербург : Лань, 2026. - 192 с. - Библиогр.: доступна в карточке книги, на сайте ЭБС Лань. - Книга из коллекции Лань - Информатика. - ISBN 978-5-507-51226-3. - Текст : электронный., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=1004082&idb=0>.

Дополнительная литература:

1. Черпаков Игорь Владимирович. Алгоритмизация и программирование на Python : учебник для вузов / И. В. Черпаков. - Москва : Юрайт, 2026. - 159 с. - (Высшее образование). - URL: <https://urait.ru/bcode/582412> (дата обращения: 24.01.2026). - ISBN 978-5-534-21910-4 : 909.00. - Текст : электронный., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=996256&idb=0>.
2. Никитина Т. П. Программирование. Основы Python для инженеров : учебное пособие для вузов / Никитина Т. П.,Королев Л. В. - 3-е изд., стер. - Санкт-Петербург : Лань, 2026. - 156 с. - Библиогр.: доступна в карточке книги, на сайте ЭБС Лань. - Книга из коллекции Лань - Информатика. - ISBN 978-5-507-51280-5. - Текст : электронный., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=1004628&idb=0>.

3. Борзунов С. В. Языки программирования. Python: решение сложных задач : учебное пособие для вузов / Борзунов С. В., Кургалин С. Д. - 2-е изд., стер. - Санкт-Петербург : Лань, 2026. - 192 с. - Библиогр.: доступна в карточке книги, на сайте ЭБС Лань. - Книга из коллекции Лань - Информатика. - ISBN 978-5-507-51226-3. - Текст : электронный., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=1004082&idb=0>.

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

1. Операционная система Microsoft Windows
2. Пакет прикладных программ Microsoft Office
3. Правовая система «Консультант плюс»
4. Правовая система «Гарант».
5. Интернет браузеры (Mozilla Firefox, Google Chrome)

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 02.03.02 - Фундаментальная информатика и информационные технологии.

Автор(ы): Золотых Николай Юрьевич, доктор физико-математических наук, доцент.

Заведующий кафедрой: Мееров Иосиф Борисович, кандидат технических наук.

Программа одобрена на заседании методической комиссии от 02.12.2024, протокол № 5.