

**Федеральное государственное автономное образовательное
учреждение высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»**

УТВЕРЖДЕНО

**Директор физтех-школы
прикладной математики и
информатики**

А.М. Райгородский

	Рабочая программа дисциплины (модуля)
по дисциплине:	Python и основы анализа данных
по направлению:	Прикладная математика и информатика
профиль подготовки:	А1360: Передовые методы искусственного интеллекта Физтех-школа Прикладной Математики и Информатики центр практик и стажировок ФПМИ
курс:	1
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 1 (осенний) - Дифференцированный зачет

Аудиторных часов: 30 всего, в том числе:

лекции: 0 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 60 час.

Всего часов: 90, всего зач. ед.: 2

Программу составил: А.Ю. Ширяев, заведующий кафедрой

Программа обсуждена на заседании центра практик и стажировок ФПМИ 05.08.2025

Аннотация

Дисциплина "Python и основы анализа данных" предназначена для студентов, желающих овладеть основами программирования на языке Python и основными методами анализа данных. Дисциплина направлена на формирование у студентов базовых и прикладных компетенций в области программирования на языке Python и анализа данных как основы для дальнейшего изучения машинного обучения и технологий искусственного интеллекта.

В рамках курса рассматриваются синтаксис и парадигмы языка Python, принципы организации данных, методы их обработки, визуализации и интерпретации. Особое внимание уделяется работе с современными библиотеками и инструментами (NumPy, Pandas, Matplotlib, Seaborn, Jupyter) и решению реальных задач анализа данных.

Обучение строится на сочетании теоретических лекций, интерактивных семинаров, лабораторных и проектных занятий. Значительная часть учебной нагрузки реализуется при участии ИТ-практиков и промышленных партнёров (Яндекс, Сбер) в формате кейсов, мастер-классов, хакатонов и командных проектов.

Изучение дисциплины позволяет студентам:

- овладеть навыками программирования и аналитического мышления;
- научиться применять методы анализа данных для решения практических и исследовательских задач;
- подготовиться к освоению курсов «Машинное обучение», «Глубинное обучение», «Методы оптимизации» и других дисциплин ИИ-трека программы AI360.

Курс базируется на знаниях, полученных при изучении дисциплины «Программирование на C++», и является предшествующей для дисциплины «Технологии программирования» и «Машинное обучение».

Дисциплина «Python и основы анализа данных» входит в состав образовательных программ по треку «Искусственный интеллект».

1. Цели и задачи

Цель дисциплины

Обучить студентов основам программирования на языке Python. Сформировать у студентов практические и исследовательские компетенции в области программирования на языке Python и анализа данных, обеспечивающие способность решать реальные задачи искусственного интеллекта. Особый акцент делается на кейсы промышленных партнёров (Яндекс, Сбер и др.) и проектную работу.

Задачи дисциплины

- развить умение писать простой код на Python для решения задач анализа данных;
 - развитие логического мышления и способности к решению проблем.
- А также:
- Освоение синтаксиса и парадигм программирования на Python.
 - Приобретение навыков работы с основными библиотеками для анализа данных (NumPy, Pandas, Matplotlib, Seaborn).
 - Формирование компетенций по обработке, очистке и визуализации данных.
 - Решение прикладных кейсов и проектных задач с использованием Python.
 - Работа в команде над мини-проектами под наставничеством ИТ-практиков.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-2 Способен использовать современные	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности

информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области
	ОПК-2.3 Знает основные требования информационной безопасности
ОПК-3 Способен составлять и оформлять научные и (или) технические (технологические, инновационные) отчеты (публикации, проекты)	ОПК-3.2 Владеет на практике методологией составления научно-технических отчетов (проектов)
	ОПК-3.1 Знает основные правила оформления научных публикаций и научно-технической документации, в том числе с использованием прикладного программного обеспечения
	ОПК-3.3 Владеет методами визуального и графического представления результатов научной (научно-технической, инновационной технологической) деятельности в виде отчетов, научных публикаций
PL-1 Способен применять язык программирования Python для решения задач в области ИИ	PL-1.1 Разрабатывает и отлаживает прикладные решения разной сложности и для разного круга конечных пользователей с использованием языка программирования Python, тестирует, испытывает и оценивает качество таких решений
	PL-1.2 Осуществляет выбор инструментов разработки на Python, приемлимых для создания прикладной системы обработки научных данных, машинного обучения и визуализации с заданными требованиями
	PL-1.3 Разрабатывает и поддерживает системы обработки больших данных различной степени сложности
ПК-2 Способен самостоятельно или в качестве члена (руководителя) малого коллектива организовывать и проводить научные исследования и их апробацию	ПК-2.1 Знает принципы построения научной работы, методы сбора и анализа полученного материала, способы аргументации
	ПК-2.2 Способен планировать и проводить научные исследования самостоятельно или в качестве члена (руководителя) малого научного коллектива
	ПК-2.3 Способен проводить апробацию результатов научно-исследовательской работы посредством публикации научных статей и участия в конференциях

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- основные концепции программирования на Python;
- ключевые функции библиотеки Pandas для анализа данных.

уметь:

- писать простой код на Python для работы с данными;
- анализировать данные с помощью Pandas и других инструментов;
- визуализировать данные с помощью Matplotlib или Seaborn.

владеть:

- базовыми навыками программирования на Python;
- навыками анализа данных с помощью Pandas;
- способностью применять полученные знания к решению реальных задач анализа данных.

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Основы Python		8		6
2	Структуры данных в Python		3		8
3	Работа с файлами и библиотеками		5		12
4	Анализ данных с использованием Python		3		10
5	Проект по анализу данных		11		24
Итого часов			30		60
Подготовка к экзамену		0 час.			
Общая трудоёмкость		90 час., 2 зач.ед.			

4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 1 (Осенний)

1. Основы Python

Введение в Python: история, особенности, преимущества. Синтаксис Python: переменные, типы данных, операторы. Управляющие конструкции: условные операторы, циклы. Функции и модули в Python.

Форматы обучения: Лекции, интерактивные семинары, практические занятия с живым кодингом, мастер-классы от ИТ-практиков, мини-проекты, групповые обсуждения.

Основные вопросы темы: Установка и настройка рабочей среды (Python, IDE, Jupyter Notebook), базовый синтаксис и типы данных, операции, ввод-вывод, условные операторы и циклы, функции и обработка ошибок, основы структурирования кода и написания документации.

Варианты для конкурса проектов: "CLI-калькулятор с историей операций", "Интерактивный конвертер единиц измерения", "Мини-чат с вводом/выводом и базовой валидацией", "Игрушечный REPL для арифметических выражений".

Варианты для групповых проектов: "Учебный интерактивный квиз по синтаксису Python", "Консольный менеджер заметок", "Приложение «Дневник кода»: учёт изученных тем и выполненных задач".

Варианты мастер-классов от индустриального партнёра: "Python как язык быстрого прототипирования сервисов" (Яндекс), "Инструменты разработчика: PyCharm, venv, pip, poetry и управление зависимостями" (Сбер).

Практические кейсы от индустриальных партнёров: "Автоматизация рутинных операций в команде аналитиков: написание скриптов для обработки CSV-файлов" (Авито), "Сценарии начальной автоматизации DevOps-процессов с помощью Python" (Сбер).

Варианты хакатона: "Bootcamp-хакатон: собрать полезный CLI-инструмент за 8 часов", "Квиз-хакатон по базовому синтаксису Python".

2. Структуры данных в Python

Списки, кортежи, множества и словари. Работа с коллекциями данных. Индексация и срезы. Применение структур данных в решении задач.

Форматы обучения: Лекции, интерактивные семинары с живым кодингом, практические занятия, командные мини-проекты, мастер-классы от ИТ-практиков, групповые обсуждения и работа с кейсами.

Основные вопросы темы: Встроенные структуры данных Python (списки, кортежи, множества, словари), операции добавления, удаления, сортировки и поиска элементов, генераторы списков и словарей, итераторы и диапазоны, особенности ссылочной модели данных, вложенные структуры и работа с ними, сравнение эффективности различных структур хранения информации.

Варианты для конкурса проектов: "Система учёта задач с использованием словарей и списков", "Интерактивный телефонный справочник", "Мини-таблица лидеров игр с сортировкой и фильтрацией", "Анализ частоты слов в тексте с использованием словарей", "Модель корзины интернет-магазина с подсчётом стоимости и скидок".

Варианты для групповых проектов: "Консольное приложение для управления списком студентов и оценок", "Учебный трекер привычек с сохранением статистики", "Интерактивный менеджер расходов с категориями и отчётами", "Программа анализа логов с группировкой событий по типам".

Варианты мастер-классов от индустриального партнёра: "Эффективные структуры данных Python для задач аналитики" (Сбер), "Контейнеры и коллекции Python в разработке реальных сервисов" (Яндекс).

Практические кейсы от индустриальных партнёров: "Оптимизация структуры хранения данных о клиентах и транзакциях для анализа" (Сбер), "Построение мини-системы инвентаризации объявлений с помощью встроенных коллекций Python" (Авито).

Варианты хакатона: "DataStruct-хакатон: реализовать мини-приложение с использованием трёх разных структур данных", "Квиз-хакатон по коллекциям Python".

3. Работа с файлами и библиотеками

Основы работы с файлами в Python. Использование стандартных библиотек для чтения и записи данных. Знакомство с популярными библиотеками для анализа данных (например, NumPy, Pandas).

Форматы обучения: Лекции, интерактивные семинары, лабораторные работы, мастер-классы от ИТ-практиков, проектные занятия с обработкой данных, мини-хакатоны, групповые обсуждения.

Основные вопросы темы: Работа с файловой системой и путями, чтение и запись текстовых и бинарных файлов, обработка CSV и JSON, сериализация данных (модуль pickle), организация структуры проекта, импорт и использование стандартных библиотек Python, установка и управление внешними пакетами с помощью pip и venv, обзор популярных библиотек для анализа данных — NumPy, Pandas, Matplotlib, Seaborn.

Варианты для конкурса проектов: "Консольный файловый менеджер с экспортом данных в CSV/JSON", "Программа анализа текстовых файлов и подсчёта статистики", "Скрипт для объединения и очистки данных из нескольких файлов", "Мини-поисковик по локальным файлам", "Архиватор логов с настройкой параметров сжатия".

Варианты для групповых проектов: "Система учёта и импорта данных из файлов разного формата", "Интерактивный парсер данных из открытых источников", "Учебный репозиторий библиотек Python с демонстрационными примерами", "Приложение для визуализации данных из CSV с помощью Matplotlib".

Варианты мастер-классов от индустриального партнёра: "Инструменты разработчика: работа с внешними библиотеками, `venv` и `pip`" (Сбер), "Практика управления зависимостями в проектах Python" (Яндекс).

Практические кейсы от индустриальных партнёров: "Скрипты начальной автоматизации загрузки и предобработки данных" (Авито), "Парсер технических логов с сохранением результатов в JSON" (Сбер).

Варианты хакатона: "FileData-хакатон: собрать утилиту для работы с файлами CSV и JSON за 8 часов", "Квиз-хакатон по библиотекам стандартной библиотеки Python".

4. Анализ данных с использованием Python

Обработка и анализ данных с помощью библиотеки Pandas. Визуализация данных с помощью библиотеки Matplotlib или Seaborn. Работа с большими наборами данных. Применение методов анализа данных для получения информации из наборов данных.

Форматы обучения: Лекции, практические занятия в Jupyter Notebook, лабораторные работы с реальными наборами данных, мастер-классы от ИТ-практиков, командные проекты, кейс-семинары и мини-хакатоны.

Основные вопросы темы: Основные этапы анализа данных, загрузка и очистка данных с использованием библиотек NumPy и Pandas, обработка пропусков и выбросов, агрегация и группировка данных, построение сводных таблиц, визуализация данных с помощью Matplotlib и Seaborn, базовые методы статистического анализа, формирование гипотез и их проверка, работа с реальными датасетами.

Варианты для конкурса проектов: "Анализ данных о продажах и визуализация динамики", "Исследование погодных данных и выявление сезонных трендов", "Анализ успеваемости студентов и визуализация факторов влияния", "Исследование пользовательских отзывов и построение распределений", "Анализ данных о транспорте и построение графиков интенсивности".

Варианты для групповых проектов: "Исследование открытого датасета (Kaggle, Rosstat, OpenData) с отчётом и визуализацией", "Создание интерактивного дашборда на Python", "Проект по анализу данных социального поведения пользователей", "Мини-исследование рынка с применением статистических методов".

Варианты мастер-классов от индустриального партнёра: "Практика анализа данных в проектах ИИ: Pandas и визуализация" (Яндекс), "Data-driven подходы: от анализа к гипотезам" (Сбер).

Практические кейсы от индустриальных партнёров: "Исследование клиентских данных и визуализация метрик удержания" (Сбер), "Предобработка и анализ объявлений для выявления закономерностей спроса" (Авито).

Варианты хакатона: "DataAnalysis-хакатон: исследовать и визуализировать открытый датасет", "Квиз-хакатон по библиотекам Pandas и Matplotlib".

5. Проект по анализу данных

Построение проекта по анализу данных от начала до конца. Загрузка и предобработка данных. Проведение анализа данных и выявление закономерностей. Визуализация результатов анализа. Подготовка отчета или презентации на основе результатов анализа.

"Форматы обучения: Командные проектные занятия, консультации с наставниками, мастер-классы от ИТ-практиков, защита проектов, участие в хакатонах и кейс-чемпионатах, работа в средах GitHub и Jupyter Notebook.

Основные вопросы темы: Формулировка задачи анализа данных, определение целей и гипотез, поиск и подготовка данных, выбор инструментов и библиотек Python, проведение анализа и визуализация результатов, подготовка отчёта и презентации проекта, оформление и защита итоговых результатов.

Варианты для конкурса проектов: "Исследование пользовательских данных сервиса с построением отчёта", "Анализ динамики продаж по регионам", "Выявление факторов, влияющих на успешность продукта", "Исследование паттернов поведения пользователей на основе открытых данных", "Создание аналитического отчёта с визуализацией и рекомендациями".

Варианты для групповых проектов: "Проект по анализу данных реального кейса от индустриального партнёра", "Разработка интерактивного дашборда для визуализации ключевых метрик", "Исследование данных компании и формулировка бизнес-гипотез", "Сравнительный анализ открытых источников данных и собственных измерений".

Варианты мастер-классов от индустриального партнёра: "Как оформлять аналитический проект: от данных к презентации" (Сбер), "Опыт внедрения аналитических решений на Python в продакшн" (Яндекс).

Практические кейсы от индустриальных партнёров: "Проект по анализу пользовательских метрик в цифровом сервисе" (Яндекс), "Исследование клиентской базы и построение отчётности для бизнес-подразделений" (Сбер).

Варианты хакатона: "AI360 Data Project Hack: реализовать полный цикл анализа данных за 48 часов", "DemoDay-хакатон: защита итоговых проектов по анализу данных".

5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Для проведения занятий используются специализированные компьютерные классы, оснащённые современными персональными компьютерами (не менее одного рабочего места на студента), мультимедийным оборудованием, включая проектор, интерактивную панель и акустическую систему, а также высокоскоростным доступом в сеть Интернет. Применяется программное обеспечение для демонстрации кода и интерактивных занятий, включая Jupyter Notebook, PyCharm и VS Code.

Программная среда включает язык программирования Python версии 3.10 и выше, библиотеки и инструменты анализа данных NumPy, Pandas, Matplotlib, Seaborn, SciPy, а также системы управления пакетами и виртуальными средами pip, venv и poetry. Для работы используются среды разработки и ноутбук-платформы Jupyter Notebook и Google Colab, инструменты командной работы и хранения кода Git и GitHub, в том числе образовательная среда GitHub Classroom. При необходимости студенты и преподаватели получают доступ к корпоративным вычислительным ресурсам и облачным средам промышленных партнёров, включая Яндекс.Cloud и СберCloud.

Для проектной и кейсовой работы используются открытые и корпоративные наборы данных (Kaggle, Rosstat, OpenData, Яндекс.Датасеты, Сбер AI Hub), а также онлайн-платформы для хакатонов, совместных проектов и защиты кейсов (Miro, Notion, Zoom, MS Teams).

Дополнительно дисциплина обеспечивается печатными и электронными учебными материалами, методическими указаниями и демонстрационными ноутбуками с примерами кода. Предусмотрена возможность проведения гостевых лекций и мастер-классов с участием ИТ-практиков в очном и онлайн-формате, а также доступ студентов к внутренней образовательной платформе программы AI360 и системе управления обучением университета.

Такая материально-техническая база обеспечивает реализацию дисциплины на современном уровне, включая выполнение лабораторных и проектных работ, проведение кейсов и хакатонов, а также поддержку индивидуальной и командной исследовательской деятельности студентов.

Опционально, по согласованию с промышленными партнёрами, студенты могут проходить обучение и выполнять проектные задания на площадках партнёров в дистанционном формате, с использованием специализированных облачных сервисов, корпоративных платформ совместной разработки и хранилищ данных. Такой формат обеспечивает доступ к актуальной промышленной инфраструктуре и позволяет студентам закрепить навыки работы с инструментами, применяемыми в реальной профессиональной среде.

6.Перечень рекомендуемой литературы

Основная литература

1. Программирование на C++ [Электронный ресурс], Электрон. версия печ. публикации / Н. Дейл, Ч. Уимз, М. Хедингтон. — М., ДМК Пресс, 2007

Дополнительная литература

1. Лутц М. Изучаем Python. — СПб.: Питер, 2022.
2. Рамалхо Л. Fluent Python. Пишем на современном Python. — М.: ДМК Пресс, 2021.
3. Свейгарт Э. Автоматизация рутинных задач с помощью Python. — СПб.: Питер, 2021.
4. Бейдер Д. Python. Трюки для профи. Эффективные рецепты программирования. — СПб.: Питер, 2020.
5. Доуни А. Think Python: Как думать как программист. — СПб.: Питер, 2019.
6. Гуттаг Дж. Введение в вычисления и программирование с использованием Python. — М.: Вильямс, 2018.

Литература по треку «Искусственный интеллект»:

7. Мюллер А., Гвидо С. Введение в машинное обучение с помощью Python. — М.: ДМК Пресс, 2020.
8. Чоллет Ф. Глубокое обучение на Python. — СПб.: Питер, 2019.
9. Грин Х. Глубокое обучение с использованием Python и Keras. — М.: ДМК Пресс, 2021.
10. Расмуссен С., Блюменсток Дж. Python и машинное обучение. — СПб.: Питер, 2021.

7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

- Официальная документация Python: <https://docs.python.org/3/>
- NumPy: <https://numpy.org/doc/>
- pandas: <https://pandas.pydata.org/docs/>
- matplotlib: <https://matplotlib.org/stable/contents.html>
- scikit-learn: <https://scikit-learn.org/stable/>
- PyTorch: <https://pytorch.org/docs/>
- TensorFlow: https://www.tensorflow.org/api_docs
- Ресурсы промышленных партнёров: открытые API, исследовательские публикации и образовательные платформы."

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)

Для реализации дисциплины используются:

- интерпретатор Python 3.x, среды разработки (PyCharm, VS Code, Jupyter Notebook, Google Colab);
- библиотеки Python (NumPy, pandas, matplotlib, scikit-learn, Flask, requests, BeautifulSoup);
- системы управления данными и версиями (SQLite, PostgreSQL, GitHub/GitLab);
- облачные и SaaS-сервисы (Google Colab, Kaggle, Яндекс.Cloud, СберCloud);
- SaaS-сервисы для совместной проектной работы (Miro, Notion, Trello, Slack, GitHub Projects);
- программное обеспечение для организации видеоконференций (Zoom, MS Teams, Google Meet) для дистанционного участия специалистов промышленных партнёров в мастер-классах, семинарах и консультациях;
- системы дистанционного обучения (Moodle, Canvas, Blackboard);
- специализированные облачные сервисы и корпоративные платформы промышленных партнёров для проведения дистанционных практик и работы с кейсами.

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Студент, изучающий дисциплину, должен с одной стороны, овладеть общим понятийным аппаратом, а с другой стороны, должен научиться применять теоретические знания на практике. Успешное освоение дисциплины требует:

- посещения студентом всех видов аудиторных занятий;
- качественной самостоятельной подготовки к практическим занятиям, активной работы на них;
- активной самостоятельной и аудиторной работы студента;

- своевременной сдачи преподавателю заданий по аудиторным видам работ.

1. При изучении дисциплины рекомендуется сочетать лекционные занятия с выполнением практических заданий, так как освоение языка Python требует регулярной практики программирования.
2. Особое внимание следует уделить выполнению кейсов и проектных заданий, предоставляемых индустриальными партнёрами. Эти задания позволяют закрепить теоретические знания и сформировать практические навыки, применимые в профессиональной деятельности в области искусственного интеллекта.
3. При подготовке к занятиям рекомендуется использовать не только учебную литературу, но и официальную документацию Python и библиотек (NumPy, pandas, scikit-learn и др.), а также справочные материалы, размещённые в системах дистанционного обучения.
4. Для успешного освоения дисциплины необходимо регулярно использовать системы контроля версий (GitHub/GitLab), облачные и SaaS-сервисы (Google Colab, Kaggle, Яндекс.Cloud, СберCloud) и сервисы совместной работы (Miro, Notion, Trello, Slack или их российские аналоги).
5. Важно активно участвовать в интерактивных формах обучения — мастер-классах, групповых обсуждениях, мини-хакатонах и дистанционных практиках с индустриальными партнёрами. Эти форматы дают возможность получить опыт командной работы и взаимодействия с практиками ИТ-индустрии.
6. При подготовке к зачётам и защите проектов рекомендуется ориентироваться на критерии оценивания, установленные в фонде оценочных средств, уделяя внимание не только правильности выполнения заданий, но и качеству кода, его оптимизации и соответствию требованиям кейсов.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению: Прикладная математика и информатика
профиль подготовки: АІ360: Передовые методы искусственного интеллекта
Физтех-школа Прикладной Математики и Информатики
центр практик и стажировок ФПМИ
курс: 1
квалификация: бакалавр

Семестр, формы промежуточной аттестации: 1 (осенний) - Дифференцированный зачет

Разработчик: А.Ю. Ширяев, заведующий кафедрой

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности
	ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области
	ОПК-2.3 Знает основные требования информационной безопасности
ОПК-3 Способен составлять и оформлять научные и (или) технические (технологические, инновационные) отчеты (публикации, проекты)	ОПК-3.2 Владеет на практике методологией составления научно-технических отчетов (проектов)
	ОПК-3.1 Знает основные правила оформления научных публикаций и научно-технической документации, в том числе с использованием прикладного программного обеспечения
	ОПК-3.3 Владеет методами визуального и графического представления результатов научной (научно-технической, инновационной технологической) деятельности в виде отчетов, научных публикаций
PL-1 Способен применять язык программирования Python для решения задач в области ИИ	PL-1.1 Разрабатывает и отлаживает прикладные решения разной сложности и для разного круга конечных пользователей с использованием языка программирования Python, тестирует, испытывает и оценивает качество таких решений
	PL-1.2 Осуществляет выбор инструментов разработки на Python, приемлимых для создания прикладной системы обработки научных данных, машинного обучения и визуализации с заданными требованиями
	PL-1.3 Разрабатывает и поддерживает системы обработки больших данных различной степени сложности
ПК-2 Способен самостоятельно или в качестве члена (руководителя) малого коллектива организовывать и проводить научные исследования и их апробацию	ПК-2.1 Знает принципы построения научной работы, методы сбора и анализа полученного материала, способы аргументации
	ПК-2.2 Способен планировать и проводить научные исследования самостоятельно или в качестве члена (руководителя) малого научного коллектива
	ПК-2.3 Способен проводить апробацию результатов научно-исследовательской работы посредством публикации научных статей и участия в конференциях

2. Показатели оценивания компетенций

В результате изучения дисциплины «Python и основы анализа данных» обучающийся должен:

знать:

- основные концепции программирования на Python;
- ключевые функции библиотеки Pandas для анализа данных.

уметь:

- писать простой код на Python для работы с данными;
- анализировать данные с помощью Pandas и других инструментов;
- визуализировать данные с помощью Matplotlib или Seaborn.

владеть:

- базовыми навыками программирования на Python;
- навыками анализа данных с помощью Pandas;
- способностью применять полученные знания к решению реальных задач анализа данных.

3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

1. Что такое Python? Каковы его основные особенности?
2. Какие типы данных поддерживает Python? Приведите примеры.
3. Чем отличаются списки и кортежи в Python?
4. Какие управляющие конструкции используются в Python? Приведите примеры.
5. Что такое функция в Python? Как объявить функцию и передать параметры?
6. Как создать словарь в Python? Как получить доступ к элементам словаря?
7. В чем разница между множеством и списком в Python?
8. Как открыть файл в Python для чтения и записи данных?
9. Какие библиотеки используются для анализа данных в Python? Укажите их основные возможности.
10. Что такое библиотека Pandas? Какие структуры данных она предоставляет?

4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

1. Как провести загрузку и предобработку данных с использованием Pandas?
2. Как создать график с использованием библиотеки Matplotlib?
3. В чем отличие между диаграммой рассеяния и гистограммой?
4. Какие шаги включает процесс создания проекта по анализу данных с использованием Python?
5. Какие инструменты можно использовать для подготовки отчета или презентации на основе результатов анализа данных?
6. Что такое индексация и срезы в Python? Как их использовать?
7. Какие методы сортировки списка существуют в Python?
8. Что такое модуль в Python? Как его импортировать?
9. Что такое объектно-ориентированное программирование (ООП) в Python?
10. В чем разница между локальными и глобальными переменными в Python?

Критерии оценивания

Оценка "Отлично" (10) - полностью и вовремя решены все задачи без ошибок. Продемонстрирован грамотный подход к решению задач, реализованы оптимальные алгоритмы, код оформлен в едином удобочитаемом стиле.

Оценка "Отлично" (9) - полностью и вовремя решены все задачи без ошибок. Продемонстрирован грамотный подход к решению задач, реализованы оптимальные алгоритмы.

Оценка "Отлично" (8) - полностью и вовремя решены все задачи без ошибок. Продемонстрирован грамотный подход к решению задач.

Оценка "Хорошо" (7) - полностью решены все задачи. Допущены несущественные ошибки.

Оценка "Хорошо" (6) - полностью решено большинство задач. В некоторых задачах допущены и не исправлены ошибки, либо некоторые задачи решены частично.

Оценка "Хорошо" (5) - полностью решено две трети задач. В некоторых задачах допущены и не исправлены ошибки, либо некоторые задачи решены частично.

Оценка "Удовлетворительно" (4) - полностью решено более половины задач. В остальных задачах допущены и не исправлены ошибки, либо некоторые задачи решены частично.

Оценка "Удовлетворительно" (3) - полностью решено более половины задач.

Оценка "Неудовлетворительно" (2) - решено менее половины задач.

Оценка "Неудовлетворительно" (1) - не решено ни одной задачи.

5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Дифференцированный зачет может проводиться по итогам текущей успеваемости и сдачи заданий и других видов работ, предусмотренных программой дисциплины и (или) путем организации специального опроса, проводимого в устной и (или) письменной форме.

При проведении устного дифференцированного зачета обучающемуся предоставляется 30 минут на подготовку. Опрос обучающегося не должен превышать одного астрономического часа.

Во время проведения дифференцированного зачета обучающиеся могут пользоваться программой дисциплины, а также справочной литературой, конспектами лекций или другими материалами.