МИНОБРНАУКИ РОССИИ

ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

НАУЧНО-ОБРАЗОВАТЕЛЬНЫЙ ЦЕНТР "ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ"

Рабочая программа дисциплины (модуля)

ОСНОВЫ PYTHON ДЛЯ АНАЛИЗА ДАННЫХ

Направление и направленность (профиль)
09.04.03 Прикладная информатика. Интеллектуальный анализ данных

Год набора на ОПОП
2026

Форма обучения
очная

Владивосток 2026

Рабочая программа дисциплины (модуля) «Основы Python для анализа данных» составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 09.04.03 Прикладная информатика (утв. приказом Минобрнауки России от 19.09.2017г. №916) и Порядком организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры (утв. приказом Минобрнауки России от 06.04.2021 г. N245).

Составитель(и):

Вишневский А.А.

Кригер А.Б.

Утверждена на заседании научно-образовательный центр "искусственный интеллект" от 27.05.2026 , протокол № 5

СОГЛАСОВАНО:

Заведующий кафедрой (разработчика)

Кригер А.Б.

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ	
Сертификат	1582918206
Номер транзакции	0000000000F7D824
Владелец	Кригер А.Б.

1 Цель, планируемые результаты обучения по дисциплине (модулю)

Цель освоения дисциплины заключается в формировании у обучающихся базовых знаний и практических навыков программирования на Python, необходимых для обработки, анализа и визуализации данных с использованием современных библиотек и инструментов. Для достижения этой цели решаются следующие задачи:

1. изучение основ Python, включая синтаксис, структуры данных (списки, словари, кортежи, множества) и управляющие конструкции, а также разработку простых скриптов и функций для автоматизации задач;
2. освоение работы с библиотеками для анализа данных, такими как NumPy и pandas, для эффективной обработки и манипуляции данными.

Планируемыми результатами обучения по дисциплине (модулю), являются знания, умения, навыки. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы, представлен в таблице 1.

Таблица 1 – Компетенции, формируемые в результате изучения дисциплины (модуля)

Название ОПОП ВО, сокращенное	Код и формулировка компетенции	Код и формулировка индикатора достижения компетенции	Результаты обучения по дисциплине		
			Код результата	Формулировка результата	
09.04.03 «Прикладная информатика» (М-ПИ)	ПКВ-1 : Способен использовать и развивать методы научных исследований и инструментария в области интеллектуального анализа данных и интеллектуальных систем	ПКВ-1.2к : Разрабатывает принципиально новые инструментальные средства (модели, алгоритмы, технологии) в области интеллектуального анализа данных	РД1	Знание	основ объектно-ориненированнного программирования
			РД2	Умение	умение настраивать среду для создания программного кода
			РД3	Навык	использования классов, объектов, методов

В процессе освоения дисциплины решаются задачи воспитания гармонично развитой, патриотичной и социально ответственной личности на основе традиционных российских духовно-нравственных и культурно-исторических ценностей, представленные в таблице 1.2.

Таблица 1.2 – Целевые ориентиры воспитания

Воспитательные задачи	**Формирование ценностей**	**Целевые ориентиры**
Формирование научного мировоззрения и культуры мышления		
Формирование осознания ценности научного мировоззрения и критического мышления	Гуманизм	Системное мышление
Формирование коммуникативных навыков и культуры общения		
Формирование навыков публичного выступления и презентации своих идей	Взаимопомощь и взаимоуважение	Умение работать в команде и взаимопомощь

2 Место дисциплины (модуля) в структуре ОПОП

дисциплина относится к базовой части, раздел 1 Дисциплины (модули)

3. Объем дисциплины (модуля)

Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу с обучающимися (по видам учебных занятий) и на самостоятельную работу, приведен в таблице 2.

Таблица 2 – Общая трудоемкость дисциплины

Название ОПОП ВО	Форма обучения	Часть УП	Семестр (ОФО) или курс (ЗФО, ОЗФО)	Трудоемкость	Объем контактной работы (час)						СРС	Форма аттестации
				(З.Е.)	Всего	Аудиторная			Внеаудиторная			
						лек.	прак.	лаб.	ПА	КСР		
09.04.03 Прикладная информатика	ОФО	М01.В	1	3	33	8	24	0	1	0	75	Э

4 Структура и содержание дисциплины (модуля)

4.1 Структура дисциплины (модуля) для ОФО

Тематический план, отражающий содержание дисциплины (перечень разделов и тем), структурированное по видам учебных занятий с указанием их объемов в соответствии с учебным планом, приведен в таблице 3.1

Таблица 3.1 – Разделы дисциплины (модуля), виды учебной деятельности и формы текущего контроля для ОФО

№	Название темы	Код результата обучения	Кол-во часов, отведенное на				Форма текущего контроля
			Лек	Практ	Лаб	СРС	
1	Введение объектно-ориентированное программирование.	РД1	2	0	0	0	Практическая работа
2	Язык Python	РД1	2	2	0	5	Практическая работа
3	Python: модули, пакеты, библиотеки	РД1, РД2	4	0	0	10	Практическая работа
4	Переменные, константы, типы данных	РД1	0	2	0	5	Практическая работа
5	Циклы, условия, функции	РД1	0	2	0	10	Практическая работа
6	Математические вычисления	РД1	0	2	0	5	
7	Составные типы данных: строки, списки, кортежи, словари, множества.	РД1	0	6	0	10	Практическая работа
8	Работа с файлами	РД2	0	2	0	10	Практическая работа
9	Библиотека numpy	РД3	0	4	0	10	Практическая работа
10	Библиотека pandas	РД3	0	4	0	10	Практическая работа
Итого по таблице			**8**	**24**	**0**	**75**	

4.2 Содержание разделов и тем дисциплины (модуля) для ОФО

Тема 1 Введение объектно-ориентированное программирование.

Содержание темы: Основные принципы и методы объектно-ориентированного программирования.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: удаленная.

Виды самостоятельной подготовки студентов по теме: .

Тема 2 Язык Python.

Содержание темы: Обзор языка программирования Python и инструментов работы с ним (Anaconda, PyCharm, Colab, GitHub).

Формы и методы проведения занятий по теме, применяемые образовательные технологии: удаленная.

Виды самостоятельной подготовки студентов по теме: .

Тема 3 Python: модули, пакеты, библиотеки.

Содержание темы: Обзор основных модулей, пакетов и библиотек языка. Стратегии работы с ними.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: удаленная.

Виды самостоятельной подготовки студентов по теме: .

Тема 4 Переменные, константы, типы данных.

Содержание темы: Изучение механизмов инициализации переменных и констант, обзор основных типов данных в Python. Чтение данных, простейшая арифметика, вывод результатов.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: удаленная.

Виды самостоятельной подготовки студентов по теме: .

Тема 5 Циклы, условия, функции.

Содержание темы: Изучение циклов, методов обработки условий и создания функций, ввод данных с помощью функции.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: удаленная.

Виды самостоятельной подготовки студентов по теме: .

Тема 6 Математические вычисления.

Содержание темы: Использование всех известных математических функций и методов, в том числе ряды, округления, факториалы.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: .

Виды самостоятельной подготовки студентов по теме: .

Тема 7 Составные типы данных: строки, списки, кортежи, словари, множества.

Содержание темы: Методы типов, включая понятия «метода». Сравнение типов данных, варианты использования (включая условия и циклы).

Формы и методы проведения занятий по теме, применяемые образовательные технологии: удаленная.

Виды самостоятельной подготовки студентов по теме: .

Тема 8 Работа с файлами.

Содержание темы: Загрузка, просмотр данных, оценка качества данных (info, counts).

Формы и методы проведения занятий по теме, применяемые образовательные технологии: удаленная.

Виды самостоятельной подготовки студентов по теме: .

Тема 9 Библиотека numpy.

Содержание темы: Работа (создание, манипуляции, преобразование) с матрицами и векторами.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: удаленная.

Виды самостоятельной подготовки студентов по теме: .

Тема 10 Библиотека pandas.

Содержание темы: Массивы данных (DataFrames) и работа с ними.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: удаленная.

Виды самостоятельной подготовки студентов по теме: .

5 Методические указания для обучающихся по изучению и реализации дисциплины (модуля)

5.1 Методические рекомендации обучающимся по изучению дисциплины и по обеспечению самостоятельной работы

Методические рекомендации по организации самостоятельной работы

В ходе изучения дисциплины студенты должны посещать аудиторные занятия (лекции, практические занятия, консультации). Особое место в овладении частью тем данной дисциплины отводится самостоятельной работе, при этом во время аудиторных занятий могут быть рассмотрены и проработаны наиболее важные и трудные вопросы по той или иной теме дисциплины, а применение уже освоенные навыков в смежных технологиях вынесены на самостоятельное обучение.

В соответствии с учебным планом направления подготовки процесс изучения дисциплины предусматривает проведение лекций, практических занятий, консультаций, а также самостоятельную работу студентов.

Ниже перечислены предназначенные для самостоятельного изучения студентами те вопросы, которые во время проведения аудиторных занятий изучаются недостаточно или изучение которых носит обзорный характер.

Перечень и тематика самостоятельных работ студентов по дисциплине

1. Статистические методы анализа данных

5.2 Особенности организации обучения для лиц с ограниченными возможностями здоровья и инвалидов

При необходимости обучающимся из числа лиц с ограниченными возможностями здоровья и инвалидов (по заявлению обучающегося) предоставляется учебная информация в доступных формах с учетом их индивидуальных психофизических особенностей:

- для лиц с нарушениями зрения: в печатной форме увеличенным шрифтом; в форме электронного документа; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания, консультации и др.

- для лиц с нарушениями слуха: в печатной форме; в форме электронного документа; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания, консультации и др.

- для лиц с нарушениями опорно-двигательного аппарата: в печатной форме; в форме электронного документа; индивидуальные задания, консультации и др.

6 Фонд оценочных средств для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине (модулю)

В соответствии с требованиями ФГОС ВО для аттестации обучающихся на соответствие их персональных достижений планируемым результатам обучения по дисциплине (модулю) созданы фонды оценочных средств. Типовые контрольные задания, методические материалы, определяющие процедуры оценивания знаний, умений и навыков, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в Приложении 1.

7 Учебно-методическое и информационное обеспечение дисциплины (модуля)

7.1 Основная литература

1. Мильнер, Б. З. Инновационное развитие: экономика, интеллектуальные ресурсы, управление знаниями : монография / под ред. Б.З. Мильнера. — Москва : ИНФРА-М, 2026. — 624 с. — (Научная мысль). - ISBN 978-5-16-003649-6. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2215527 (дата обращения: 31.05.2026)

2. Панов, М. А. Анализ данных с использованием языка программирования Python : учебное пособие / М. А. Панов. — Екатеринбург : УрГЭУ, 2024. — 329 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/481577 (дата обращения: 25.05.2026). — Режим доступа: для авториз. пользователей.

3. Федоров, Д. Ю. Программирование на Python : учебник для вузов / Д. Ю. Федоров. — 6-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 166 с. — (Высшее образование). — ISBN 978-5-534-22181-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/600874 (дата обращения: 19.05.2026).

7.2 Дополнительная литература

1. Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2025. — 343 с. — (Высшее образование). - ISBN 978-5-16-020255-6. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2166199 (дата обращения: 31.05.2026)

2. Объектно-ориентированное программирование : методические указания / составитель Т. А. Юрина. — Омск : СибАДИ, 2023. — 27 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/338579 (дата обращения: 25.05.2026). — Режим доступа: для авториз. пользователей.

3. Черпаков, И. В. Алгоритмизация и программирование на Python : учебник для вузов / И. В. Черпаков. — Москва : Издательство Юрайт, 2026. — 159 с. — (Высшее образование). — ISBN 978-5-534-21910-4. — Текст : электронный // Образовательная

платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/582412 (дата обращения: 19.05.2026).

7.3 Ресурсы информационно-телекоммуникационной сети "Интернет", включая профессиональные базы данных и информационно-справочные системы (при необходимости):

1. Образовательная платформа "ЮРАЙТ"
2. Электронно-библиотечная система "ZNANIUM.COM"
3. Электронно-библиотечная система "ЛАНЬ"
4. Open Academic Journals Index (OAJI). Профессиональная база данных - Режим доступа: http://oaji.net/
5. Президентская библиотека им. Б.Н.Ельцина (база данных различных профессиональных областей) - Режим доступа: https://www.prlib.ru/
6. Информационно-справочная система "Консультант Плюс" - Режим доступа: http://www.consultant.ru/

8 Материально-техническое обеспечение дисциплины (модуля) и перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения

Основное оборудование:

- Коммутатор SuperStack 3 (16*10/100 19")
- Монитор облачный 23" LG23CAV42K/мышь Geniu
- Мультимедийный проектор №1 Casio XJ-V2
- Облачный монитор 23" LG CAV42K
- Облачный монитор LG Electronics черный +клавиатура+мышь
- П/К DNS Office T300, мышь Genius NetScroll 100, клавиатура Genius KB-06X, монитор AOC919 19"
- Проектор Casio XJ-V1
- Уст-во бесп.питания UPS-3000

Программное обеспечение:

- Microsoft Office Professional Plus 2016
- Python

МИНОБРНАУКИ РОССИИ

ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

НАУЧНО-ОБРАЗОВАТЕЛЬНЫЙ ЦЕНТР "ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ"

Фонд оценочных средств
для проведения текущего контроля
и промежуточной аттестации по дисциплине (модулю)

ОСНОВЫ PYTHON ДЛЯ АНАЛИЗА ДАННЫХ

Направление и направленность (профиль)
09.04.03 Прикладная информатика. Интеллектуальный анализ данных

Год набора на ОПОП
2026

Форма обучения
очная

Владивосток 2026

1 Перечень формируемых компетенций

Название ОПОП ВО, сокращенное	Код и формулировка компетенции	Код и формулировка индикатора достижения компетенции
09.04.03 «Прикладная информатика» (М-ПИ)	ПКВ-1 : Способен использовать и развивать методы научных исследований и инструментария в области интеллектуального анализа данных и интеллектуальных систем	ПКВ-1.2к : Разрабатывает принципиально новые инструментальные средства (модели, алгоритмы, технологии) в области интеллектуального анализа данных

Компетенция считается сформированной на данном этапе в случае, если полученные результаты обучения по дисциплине оценены положительно (диапазон критериев оценивания результатов обучения «зачтено», «удовлетворительно», «хорошо», «отлично»). В случае отсутствия положительной оценки компетенция на данном этапе считается несформированной.

2 Показатели оценивания планируемых результатов обучения

Компетенция ПКВ-1 «Способен использовать и развивать методы научных исследований и инструментария в области интеллектуального анализа данных и интеллектуальных систем»

Таблица 2.1 – Критерии оценки индикаторов достижения компетенции

Код и формулировка индикатора достижения компетенции	Результаты обучения по дисциплине			Критерии оценивания результатов обучения
	Код рез-та	Тип рез-та	Результат	
ПКВ-1.2к : Разрабатывает принципиально новые инструментальные средства (модели, алгоритмы, технологии) в области интеллектуального анализа данных	РД 1	Знание	основ объектно-ориненированного программирования	знание понятий классов, объектов, методов
	РД 2	Умение	умение настраивать среду для создания программного кода	настроенная среда для создания и отладки программного кода
	РД 3	Навык	использования классов, объектов, методов	адекватное использование классов, объектов, методов

Таблица заполняется в соответствии с разделом 1 Рабочей программы дисциплины (модуля).

3 Перечень оценочных средств

Таблица 3 – Перечень оценочных средств по дисциплине (модулю)

Контролируемые планируемые результаты обучения	Контролируемые темы дисциплины	Наименование оценочного средства и представление его в ФОС	
		Текущий контроль	Промежуточная аттестация

Очная форма обучения				
РД1	Знание : основ объектно-оринениованнного программирования	1.1. Введение объектно-ориентированное программирование.	Практическая работа	Тест
		1.2. Язык Python	Практическая работа	Тест
		1.3. Python: модули, пакеты, библиотеки	Практическая работа	Тест
		1.4. Переменные, константы, типы данных	Практическая работа	Тест
		1.5. Циклы, условия, функции	Практическая работа	Тест
		1.6. Математические вычисления	Практическая работа	Тест
		1.7. Составные типы данных: строки, списки, кортежи, словари, множества.	Практическая работа	Тест
РД2	Умение : умение настраивать среду для создания программного кода	1.3. Python: модули, пакеты, библиотеки	Практическая работа	Тест
		1.8. Работа с файлами	Практическая работа	Тест
РД3	Навык : использования классов, объектов, методов	1.9. Библиотека numpy	Практическая работа	Тест
		1.10. Библиотека pandas	Практическая работа	Тест

4 Описание процедуры оценивания

Качество сформированности компетенций на данном этапе оценивается по результатам текущих и промежуточных аттестаций при помощи количественной оценки, выраженной в баллах. Максимальная сумма баллов по дисциплине (модулю) равна 100 баллам.

Качество сформированности компетенций на данном этапе оценивается по результатам текущих и промежуточных аттестаций при помощи количественной оценки, выраженной в баллах. Максимальная сумма баллов по дисциплине (модулю) равна 100 баллам.

Вид учебной деятельности	Оценочное средство		
	Отчёт по практическим работам	Вопросы к экзамену	Итого
Лекция		10	10
Промежуточная аттестация		30	30
Практические занятия	60		60
Самостоятельная работа			
Итого	60	40	100

Сумма баллов, набранных студентом по всем видам учебной деятельности в рамках дисциплины, переводится в оценку в соответствии с таблицей.

Сумма баллов по дисциплине	Оценка по промежуточной аттестации	Характеристика качества сформированности компетенции
от 91 до 100	«зачтено» / «отлично»	Студент демонстрирует сформированность дисциплинарных компетенций, обнаруживает всестороннее, систематическое и глубокое знание учебного материала, усвоил основную литературу и знаком с дополнительной литературой, рекомендованной программой, умеет свободно выполнять практические задания, предусмотренные программой, свободно оперирует приобретенными знаниями, умениями, применяет их в ситуациях повышенной сложности.
от 76 до 90	«зачтено» / «хорошо»	Студент демонстрирует сформированность дисциплинарных компетенций: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
от 61 до 75	«зачтено» / «удовлетворительно»	Студент демонстрирует сформированность дисциплинарных компетенций: в ходе контрольных мероприятий допускаются значительные ошибки, проявляется отсутствие отдельных знаний, умений, навыков по некоторым дисциплинарным компетенциям, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.
от 41 до 60	«не зачтено» / «неудовлетворительно»	У студента не сформированы дисциплинарные компетенции, проявляется недостаточность знаний, умений, навыков.
от 0 до 40	«не зачтено» / «неудовлетворительно»	Дисциплинарные компетенции не сформированы. Проявляется полное или практически полное отсутствие знаний, умений, навыков.

Сумма баллов, набранных студентом по всем видам учебной деятельности в рамках дисциплины, переводится в оценку в соответствии с таблицей.

Сумма баллов по дисциплине	Оценка по промежуточной аттестации	Характеристика качества сформированности компетенции
от 91 до 100	«зачтено» / «отлично»	Студент демонстрирует сформированность дисциплинарных компетенций, обнаруживает всестороннее, систематическое и глубокое знание учебного материала, усвоил основную литературу и знаком с дополнительной литературой, рекомендованной программой, умеет свободно выполнять практические задания, предусмотренные программой, свободно оперирует приобретенными знаниями, умениями, применяет их в ситуациях повышенной сложности.
от 76 до 90	«зачтено» / «хорошо»	Студент демонстрирует сформированность дисциплинарных компетенций: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
от 61 до 75	«зачтено» / «удовлетворительно»	Студент демонстрирует сформированность дисциплинарных компетенций: в ходе контрольных мероприятий допускаются значительные ошибки, проявляется отсутствие отдельных знаний, умений, навыков по некоторым дисциплинарным компетенциям, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.
от 41 до 60	«не зачтено» / «неудовлетворительно»	У студента не сформированы дисциплинарные компетенции, проявляется недостаточность знаний, умений, навыков.
от 0 до 40	«не зачтено» / «неудовлетворительно»	Дисциплинарные компетенции не сформированы. Проявляется полное или практически полное отсутствие знаний, умений, навыков.

5 Примерные оценочные средства

5.1 Примеры заданий для выполнения практических работ

Задание 1: Написание простой программы для знакомства с переменными, константами, типами данных и математических вычислений.

Задание 2: Написание простой программы для знакомства с циклами, условиями и функциями.

Задание 3: Чтение, запись, удаление и модификация файлов. Операции с файлами и директориями с использованием модуля os.

Задание 4: Создание массивов, основные операции с массивами. Решение задач с использованием массивов NumPy.

Задание 5: Создание и работа с DataFrame, использование методов .loc[], .iloc[], sort_values(), sort_index().

Краткие методические указания

Для прохождения каждой темы, студент должен подтвердить приобретенные навыки с помощью выполнения практической работы.

Шкала оценки

№	Баллы	Описание
5	12–14	Студент демонстрирует умения на итоговом уровне: умеет свободно выполнять практические задания, предусмотренные программой, свободно оперирует приобретенными умениями, применяет их в ситуациях повышенной сложности.
4	9–11	Студент демонстрирует умения на среднем уровне: освоил основные умения, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.
3	6–8	Студент демонстрирует умения и навыки на базовом уровне: в ходе контрольных мероприятий допускаются значительные ошибки, проявляется отсутствие отдельных умений, навыков по дисциплинарной компетенции, испытываются значительные затруднения при оперировании умениями и при их переносе на новые ситуации.
2	3–5	Студент демонстрирует умения и навыки на уровне ниже базового: проявляется недостаточность умений и навыков.
1	0–2	Студентом проявляется полное или практически полное отсутствие умений и навыков.

5.2 Итоговый тест

Тест: Основы ООП в Python

1. Что из перечисленного является основными принципами ООП?

- Рекурсия, итерация, генерация
- Инкапсуляция, наследование, полиморфизм
- Импорт, экспорт, сериализация
- Лямбда-функции, декораторы, генераторы

2. Как создать класс в Python?

- `def MyClass():`
- `class MyClass:`
- `new class MyClass:`
- `object MyClass:`

3. Что такое инкапсуляция?

- Способность класса наследовать методы другого класса

- Сокрытие внутренней реализации класса и предоставление интерфейса для работы с ним
- Возможность использовать один метод для разных типов данных
- Автоматическое управление памятью в Python

4. Как обозначить приватный атрибут в классе?

- Через ключевое слово `private`
- С помощью двойного подчеркивания: `__attribute`
- Через символ `#` в начале имени
- В Python нет приватных атрибутов

5. Что выведет этот код?

```
class A :
def   __init__  ( self ,  x )  :
self . x  =  x
a  =  A (  5  )
print  ( a . x )
```

- Ошибка, так как `x` не определен
- `5`
- `None`
- `>`

6. Какой метод вызывается при создании объекта?

- `__call__`
- `__init__`
- `__new__`
- `__create__`

7. Что такое полиморфизм в ООП?

- Возможность создавать несколько экземпляров класса
- Способность функции работать с разными типами данных через единый интерфейс
- Механизм сокрытия реализации методов
- Наследование свойств родительского класса

8. Как наследовать класс `B` от класса `A`?

- `class B inherit A:`
- `class B(A):`
- `class B extends A:`
- `class B -> A:`

9. Как вызвать метод родительского класса из дочернего?

- `parent.method()`
- `super().method()`
- `self.parent.method()`
- `ParentClass.method(self)`

10. Что такое "магические методы" в Python?

- Методы, доступные только в Jupyter Notebook
- Методы с двойными подчеркиваниями (например, `__str__`, `__len__`), которые вызываются автоматически
- Методы, работающие с зашифрованными данными
- Методы, определенные в модуле `magic`

Тест: Настройка среды для разработки на Python

1. Какой командой установить пакет в Python?

- `python get package_name`
- `pip install package_name`
- `python install package_name`
- `import package_name`

2. Как создать виртуальное окружение в Python?

- `python create venv`
- `python -m venv myenv`
- `virtualenv --new myenv`
- `pip init venv`

3. Как активировать виртуальное окружение в Windows?

- `source myenv/bin/activate`
- `myenv\Scripts\activate`
- `activate myenv`
- `python activate myenv`

4. Какой файл используется для описания зависимостей проекта?

- `requirements.txt`
- Оба варианта (и `requirements.txt`, и `pyproject.toml`)
- `pyproject.toml`
- `dependencies.yml`

5. Какой командой можно узнать список установленных пакетов?

- `python --list`
- `pip list`
- `pip show all`
- `python -m packages`

6. Какой инструмент используется для форматирования кода в соответствии с PEP 8?

- `pylint`
- `black`
- `flake8`
- `mypy`

7. Какой командой запустить Python-скрипт?

- `run script.py`
- `python script.py`
- `start script.py`
- `python exec script.py`

8. Какой модуль позволяет работать с путями файловой системы кроссплатформенно?

- `sys.path`
- `pathlib` или `os.path`
- `glob`
- `shutil`

9. Как добавить текущую директорию в `PYTHONPATH`?

- `export PYTHONPATH=$PWD` (Linux/macOS)
- Любой из этих вариантов (в зависимости от ОС)
- `set PYTHONPATH=%cd%` (Windows)
- `sys.path.append(os.getcwd())` в коде Python

10. Какой инструмент помогает управлять зависимостями и сборкой проекта?

- `pipenv`
- Все перечисленные (`pipenv`, `poetry`, `setuptools`)
- `poetry`
- `setuptools`

Тест: Классы, объекты и методы в Python

1. Как создать экземпляр класса?

- `new MyClass()`
- `MyClass()`
- `class MyClass()`
- `MyClass.create()`

2. Что такое `self` в методах класса?

- Зарезервированное ключевое слово, без которого код не работает
- Ссылка на экземпляр класса, передается первым аргументом
- Указатель на родительский класс
- Специальный декоратор

3. Как определить статический метод в классе?

- Через `@staticmethod`
- Любой из вариантов (и `@staticmethod`, и `@classmethod` с `cls`)
- Через `@classmethod`
- Через `self.staticmethod()`

4. Какой метод вызывается при выводе объекта через `print()`?

- `__repr__`
- `__str__`
- `__print__`
- `__display__`

5. Что делает декоратор `@property`?

- Позволяет метод сделать приватным
- Превращает метод в свойство (геттер)
- Делает метод статическим
- Позволяет переопределить оператор

6. Как перегрузить оператор + для класса?

- `def __plus__(self, other)`
- `def __add__(self, other)`
- `def __sum__(self, other)`
- `def operator+(self, other)`

7. Как удалить атрибут у объекта?

- `del obj.attr`
- Любой из вариантов (`del obj.attr` или `delattr(obj, \'attr\')`)
- `obj.remove(\'attr\')`
- `delattr(obj, \'attr\')`

8. Какой метод вызывается при удалении объекта?

- `__remove__`
- `__del__`
- `__delete__`
- `__destruct__`

9. Как создать класс, который нельзя наследовать?

- Через `@final` (в Python 3.8+)
- Никак, в Python нет строгой запрета на наследование
- Через `__noinherit__`
- Через `class FinalClass(metaclass=Final):`

10. Как проверить, является ли объект экземпляром класса?

- `obj.isinstance(MyClass)`
- `isinstance(obj, MyClass)`
- `obj.type() == MyClass`
- `type(obj) is MyClass`

Краткие методические указания

Для аттестации, по итогам практических заданий, студент должен пройти итоговое тестирование.

Шкала оценки

Оценка	Баллы	Описание

5	9-10	
4	6-8	
3	3-5	
2	0-2	

Тест: Основы ООП в Python (10 вопросов)

1. Что из перечисленного является основными принципами ООП?

- Рекурсия, итерация, генерация
- Инкапсуляция, наследование, полиморфизм
- Импорт, экспорт, сериализация
- Лямбда-функции, декораторы, генераторы

2. Как создать класс в Python?

- `def MyClass():`
- `class MyClass:`
- `new class MyClass:`
- `object MyClass:`

3. Что такое инкапсуляция?

- Способность класса наследовать методы другого класса
- Сокрытие внутренней реализации класса и предоставление интерфейса для работы с ним
- Возможность использовать один метод для разных типов данных
- Автоматическое управление памятью в Python

4. Как обозначить приватный атрибут в классе?

- Через ключевое слово `private`
- С помощью двойного подчеркивания: `__attribute`
- Через символ `#` в начале имени
- В Python нет приватных атрибутов

5. Что выведет этот код?

python

Copy

Download

```python
class A:
    def __init__(self, x):
        self.x = x
```

```
a = A(5)
print(a.x)
```

- Ошибка, так как `x` не определен
- `5`
- `None`
- `<bound method A.x of <__main__.A object>>`

6. Какой метод вызывается при создании объекта?

- `__call__`
- `__init__`
- `__new__`
- `__create__`

7. Что такое полиморфизм в ООП?

- Возможность создавать несколько экземпляров класса
- Способность функции работать с разными типами данных через единый интерфейс
- Механизм сокрытия реализации методов
- Наследование свойств родительского класса

8. Как наследовать класс `B` от класса `A`?

- `class B inherit A:`
- `class B(A):`
- `class B extends A:`
- `class B -> A:`

9. Как вызвать метод родительского класса из дочернего?

- `parent.method()`
- `super().method()`
- `self.parent.method()`
- `ParentClass.method(self)`

10. Что такое "магические методы" в Python?

- Методы, доступные только в Jupyter Notebook
- Методы с двойными подчеркиваниями (например, `__str__`, `__len__`), которые вызываются автоматически
- Методы, работающие с зашифрованными данными
- Методы, определенные в модуле `magic`

Тест: Настройка среды для разработки на Python (10 вопросов)

1. Какой командой установить пакет в Python?

- `python get package_name`
- `pip install package_name`
- `python install package_name`
- `import package_name`

2. Как создать виртуальное окружение в Python?

- `python create venv`
- `python -m venv myenv`
- `virtualenv --new myenv`
- `pip init venv`

3. Как активировать виртуальное окружение в Windows?

- `source myenv/bin/activate`
- `myenv\Scripts\activate`
- `activate myenv`
- `python activate myenv`

4. Какой файл используется для описания зависимостей проекта?

- `requirements.txt`
- Оба варианта (и `requirements.txt`, и `pyproject.toml`)
- `pyproject.toml`
- `dependencies.yml`

5. Какой командой можно узнать список установленных пакетов?

- `python --list`
- `pip list`
- `pip show all`
- `python -m packages`

6. Какой инструмент используется для форматирования кода в соответствии с PEP 8?

- `pylint`
- `black`
- `flake8`
- `mypy`

7. Какой командой запустить Python-скрипт?

- `run script.py`
- `python script.py`
- `start script.py`
- `python exec script.py`

8. Какой модуль позволяет работать с путями файловой системы кроссплатформенно?

- `sys.path`
- `pathlib` или `os.path`
- `glob`
- `shutil`

9. Как добавить текущую директорию в `PYTHONPATH`?

- `export PYTHONPATH=$PWD` (Linux/macOS)
- Любой из этих вариантов (в зависимости от ОС)
- `set PYTHONPATH=%cd%` (Windows)
- `sys.path.append(os.getcwd())` в коде Python

10. Какой инструмент помогает управлять зависимостями и сборкой проекта?

- `pipenv`
- Все перечисленные (`pipenv`, `poetry`, `setuptools`)
- `poetry`
- `setuptools`

Тест: Классы, объекты и методы в Python (10 вопросов)

1. Как создать экземпляр класса?

- `new MyClass()`
- `MyClass()`
- `class MyClass()`
- `MyClass.create()`

2. Что такое `self` в методах класса?

- Зарезервированное ключевое слово, без которого код не работает
- Ссылка на экземпляр класса, передается первым аргументом
- Указатель на родительский класс
- Специальный декоратор

3. Как определить статический метод в классе?

- Через `@staticmethod`
- Любой из вариантов (и `@staticmethod`, и `@classmethod` с `cls`)
- Через `@classmethod`
- Через `self.staticmethod()`

4. Какой метод вызывается при выводе объекта через `print()`?

- `__repr__`
- `__str__`
- `__print__`
- `__display__`

5. Что делает декоратор `@property`?

- Позволяет метод сделать приватным
- Превращает метод в свойство (геттер)
- Делает метод статическим
- Позволяет переопределить оператор

6. Как перегрузить оператор `+` для класса?

- `def __plus__(self, other)`
- `def __add__(self, other)`
- `def __sum__(self, other)`
- `def operator+(self, other)`

7. Как удалить атрибут у объекта?

- `del obj.attr`
- Любой из вариантов (`del obj.attr` или `delattr(obj, 'attr')`)
- `obj.remove('attr')`
- `delattr(obj, 'attr')`

8. Какой метод вызывается при удалении объекта?

- `__remove__`
- `__del__`
- `__delete__`
- `__destruct__`

9. Как создать класс, который нельзя наследовать?

- Через `@final` (в Python 3.8+)
- Никак, в Python нет строгой запрета на наследование
- Через `__noinherit__`
- Через `class FinalClass(metaclass=Final):`

10. Как проверить, является ли объект экземпляром класса?

- `obj.isinstance(MyClass)`
- `isinstance(obj, MyClass)`
- `obj.type() == MyClass`

- `type(obj) is MyClass`

Ответы

Тест: Основы ООП в Python

1. Основные принципы ООП:

- **Инкапсуляция, наследование, полиморфизм** *(Это "три кита" ООП.)*

2. Создание класса:

- `class MyClass:` *(Синтаксис объявления класса.)*

3. Инкапсуляция — это:

- **Сокрытие внутренней реализации класса** *(Доступ к данным через методы, а не напрямую.)*

4. Приватный атрибут:

- `__attribute` *(Двойное подчеркивание делает имя атрибута приватным.)*

5. Результат кода:

```
class A:
    def __init__(self, x):
        self.x = x
a = A(5)
print(a.x)
```

- `5` *(Инициализация и доступ к атрибуту.)*

6. Метод при создании объекта:

- `__init__` *(Конструктор класса.)*

7. Полиморфизм — это:

- **Один интерфейс для разных типов данных** *(Например, `+` работает и для чисел, и для строк.)*

8. Наследование класса:

- `class B(A):` *(Дочерний класс `B` наследует от `A`.)*

9. Вызов метода родителя:

- `super().method()` *(Стандартный способ.)*

10. Магические методы:

- **Методы с** `__названием__` *(Например, `__str__`, `__len__`.)*

Тест: Настройка среды Python

1. Установка пакета:

- `pip install package_name` *(Основной инструмент установки.)*

2. Создание виртуального окружения:

- `python -m venv myenv` *(Стандартный модуль `venv`.)*

3. Активация окружения (Windows):

- `myenv\Scripts\activate` *(Путь для Windows.)*

4. Файл зависимостей:

- `requirements.txt` *(Список пакетов для `pip install -r`.)*

5. Список установленных пакетов:

- `pip list` *(Выводит все пакеты в окружении.)*

6. Инструмент форматирования кода:

- `black` *(Автоматически форматирует код по PEP 8.)*

7. Запуск скрипта:

- `python script.py` *(Основной способ.)*

8. Работа с путями:

- `pathlib` *(Современная альтернатива `os.path`.)*

9. Добавление в `PYTHONPATH`:

- `export PYTHONPATH=$PWD` *(Для Linux/macOS.)*

10. Управление зависимостями:

- `poetry` *(Инструмент для управления пакетами и виртуальными окружениями.)*

Тест: Классы, объекты и методы

1. Создание экземпляра:

- `MyClass()` *(Вызов класса как функции.)*

2. `self` — это:

- **Ссылка на экземпляр класса** *(Первый аргумент методов.)*

3. Статический метод:

- `@staticmethod` *(Не требует `self` или `cls`.)*

4. Метод для `print()`:

- `__str__` *(Определяет строковое представление объекта.)*

5. Декоратор `@property`:

- **Превращает метод в свойство** *(Геттер для атрибута.)*

6. Перегрузка `+`:

- `__add__` *(Магический метод для сложения.)*

7. Удаление атрибута:

- `del obj.attr` *(Синтаксис удаления.)*

8. Метод при удалении объекта:

- `__del__` *(Вызывается перед удалением.)*

9. Запрет наследования:

- `@final` **(Python 3.8+)** *(Декоратор для запрета наследования.)*

10. Проверка типа объекта:

- `isinstance(obj, MyClass)` *(Рекомендуемый способ.)*