МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ГЕОДЕЗИИ И КАРТОГРАФИИ
(МИИГАиК)

Факультет геоинформатики и информационной безопасности
Кафедра информационно-измерительных систем

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 5BFEE1D286E9C95A0C0D2C611E434BD5
Владелец: Степанченко Алексей Леонидович
Действителен: с 23.09.2026 по 17.12.2027

«УТВЕРЖДАЮ»
Проректор
А. Л. Степанченко
15 мая 2026 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ) ПРОГРАММИРОВАНИЕ И ТЕХНОЛОГИИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ «Б1.О.06.03 ПРАКТИКУМ ПО ЯЗЫКАМ ПРОГРАММИРОВАНИЯ»

Уровень высшего образования: бакалавриат

Направление подготовки: 09.03.03 Прикладная информатика

Направленность (профиль) подготовки: Разработка программного обеспечения для обработки и анализа пространственных данных

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Год набора (приема на обучение): 2026

Срок получения образования: 4 года

Объем: в зачетных единицах: 3 з.е.
в академических часах: 108 ак.ч.

Москва, 2026

Разработчики:

Преподаватель кафедры Информационно-измерительных систем
Лазуренко Н. С.

Рабочая программа дисциплины (модуля) составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.03 Прикладная информатика, утвержденного приказом Минобрнауки от 19.09.2017 № 922

Согласование и утверждение

№	Подразделение	Ответственное лицо	ФИО	Виза	Дата, протокол (при наличии)
1	факультет Геоинформатики и информационной безопасности	Председатель ученого совета факультета	Матерухин А. В.	Рассмотрено	05.11.2025, № 12-25

Содержание

1. Цель и задачи освоения дисциплины (модуля)

2. Планируемые результаты обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы

3. Место дисциплины в структуре ОП

4. Объем дисциплины и виды учебной работы

5. Содержание дисциплины
   5.1. Разделы, темы дисциплины и виды занятий
   5.2. Содержание разделов, тем дисциплины и формы текущего контроля

6. Оценочные материалы текущего контроля

7. Оценочные материалы промежуточной аттестации

8. Порядок проведения промежуточной аттестации

9. Материально-техническое и учебно-методическое обеспечение дисциплины
   9.1. Перечень основной и дополнительной учебной литературы
   9.2. Профессиональные базы данных и ресурсы «Интернет», к которым обеспечивается доступ обучающихся
   9.3. Программное обеспечение и информационно-справочные системы, используемые при осуществлении образовательного процесса по дисциплине
   9.4. Оборудование и технические средства обучения

10. Методические указания по освоению дисциплины (модуля)

1. Цель и задачи освоения дисциплины (модуля)

Цель освоения дисциплины - Сформировать способность разрабатывать, проверять и улучшать небольшие программы на Python: формализовать задачу, выбирать алгоритмы и структуры данных, писать понятный модульный код, находить ошибки и подтверждать правильность и допустимые затраты решения.

Задачи изучения дисциплины:

- Освоить анализ требований, декомпозицию и обоснованный выбор алгоритмического решения по заданным ограничениям.;
- Научиться применять основные структуры данных Python, простые алгоритмы поиска, сортировки и обхода, оценивать их время и память.;
- Освоить функции, классы, модули, файловый ввод-вывод и параметризацию типов с соблюдением правил оформления и документирования кода.;
- Получить опыт отладки, обработки исключений, безопасного журналирования и разработки модульных и интеграционных тестов.;
- Освоить профилирование, небольшие улучшения производительности, представление измерений и проверку сохранения поведения.;
- Научиться сохранять историю изменений, воспроизводить среду и автоматически проверять учебный программный проект..

2. Планируемые результаты обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы

Компетенции, индикаторы и результаты обучения

ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения

ОПК-7.1 Анализирует поставленную задачу и требования к программному продукту, определяет необходимые алгоритмические и программные решения, а также оценивает их применимость для практического использования

Знать:

ОПК-7.1/Зн1 Основные парадигмы программирования и алгоритмические структуры, методы анализа и декомпозиции задач, критерии выбора алгоритмических решений, основы проектирования программных систем на языке Python, методы оценки вычислительной сложности алгоритмов и их практической применимости.

Уметь:

ОПК-7.1/Ум1 Анализировать техническое задание и выделять функциональные требования к программному продукту, выполнять декомпозицию сложных задач на подзадачи, выбирать оптимальные алгоритмические решения с учетом специфики языка Python, оценивать временную и пространственную сложность разрабатываемых алгоритмов, обосновывать выбор программных решений для конкретной практической задачи.

Владеть:

ОПК-7.1/Нв1 Навыками системного анализа предметной области и формализации требований, методами алгоритмического проектирования и выбора структур данных в Python, инструментами профилирования и оценки производительности Python-приложений, навыками технико-экономического обоснования выбранных программных решений.

ОПК-7.2 Разрабатывает эффективные алгоритмы для решения задач, а также выбирает оптимальные структуры данных для их реализации

Знать:

ОПК-7.2/Зн1 Основные алгоритмические парадигмы (жадные алгоритмы, динамическое программирование, метод "разделяй и властвуй", рекурсивные алгоритмы), базовые и продвинутые структуры данных Python (списки, кортежи, множества, словари, очереди, стеки, деревья), методы анализа временной и пространственной сложности алгоритмов с использованием нотации О-большое, критерии выбора оптимальных структур данных для различных типов задач, классические алгоритмы сортировки, поиска, обхода графов и области их применения.

Уметь:

ОПК-7.2/Ум1 Разрабатывать эффективные алгоритмические решения с учетом ограничений по времени и памяти, выбирать подходящие структуры данных на основе анализа требований задачи, оценивать и сравнивать эффективность различных алгоритмических подходов, реализовывать классические алгоритмы и адаптировать их под конкретные задачи, применять встроенные структуры данных Python и библиотеки collections для оптимизации решений.

Владеть:

ОПК-7.2/Нв1 Навыками разработки и реализации эффективных алгоритмов на языке Python, техниками оптимизации алгоритмов и выбора оптимальных структур данных, методами профилирования и анализа производительности алгоритмов, инструментами визуализации алгоритмов и структур данных.

ОПК-7.3 Пишет программный код на одном или нескольких языках программирования, следуя принципам модульности, читаемости, документирования и соответствия стандартам кодирования

Знать:

ОПК-7.3/Зн1 Синтаксис и семантику языка программирования Python, принципы структурного и модульного программирования, стандарты оформления кода PEP 8 и PEP 257, системы документирования кода (docstrings, type hints), принципы написания чистого и поддерживаемого кода, паттерны проектирования, применимые в Python.

Уметь:

ОПК-7.3/Ум1 Писать читаемый и эффективный код на языке Python с соблюдением принципов DRY и KISS, структурировать программу с использованием функций, классов и модулей, документировать код с использованием строк документации и комментариев, аннотацию типов для повышения надежности кода, использовать инструменты автоматической проверки качества кода (pylint, flake8, black).

Владеть:

ОПК-7.3/Нв1 Навыками разработки модульных Python-приложений с четким разделением ответственности, инструментами рефакторинга и оптимизации кода, методами написания самодокументируемого кода, навыками работы с системами контроля версий для командной разработки.

ОПК-7.3/Нв4 Навыками пользования современными информационными технологиями с учетом их особенностей и возможной области применения

ОПК-7.4 Проводит отладку и тестирование программного кода, выявляет и устраняет ошибки, а также оптимизирует производительность программы для обеспечения её стабильной и эффективной работы в реальных условиях

Знать:

ОПК-7.4/Зн1 Виды ошибок в программах (синтаксические, логические, времени выполнения), методологии и стратегии отладки программного кода, основы модульного и интеграционного тестирования, инструменты отладки и профилирования в Python, принципы обработки исключений и написания отказоустойчивого кода.

Уметь:

ОПК-7.4/Ум1 Использовать встроенные и внешние отладчики для поиска и устранения ошибок, разрабатывать unit-тесты с использованием модулей unittest и pytest, применять методы логирования для диагностики проблем в работе программы, проводить профилирование кода и выявлять узкие места производительности, обрабатывать исключения и создавать информативные сообщения об ошибках.

Владеть:

ОПК-7.4/Нв1 Навыками систематической отладки и рефакторинга программного кода, техниками написания тестируемого кода и создания тестовых сценариев, инструментами автоматизированного тестирования и непрерывной интеграции, методами оптимизации производительности Python-приложений, навыками использования систем контроля версий для отслеживания и исправления ошибок.

3. Место дисциплины в структуре ОП

Дисциплина (модуль) Б1.О.06.03 «Практикум по языкам программирования» относится к обязательной части образовательной программы и изучается в семестре(ах): 1.

В процессе изучения дисциплины студент готовится к решению типов задач профессиональной деятельности, предусмотренных ФГОС ВО и образовательной программой.

4. Объем дисциплины (модуля) и виды учебной работы

Период обучения	Общая трудоемкость (часы)	Общая трудоемкость (ЗЕТ)	Практические занятия (часы)	Самостоятельная работа (часы)	Промежуточная аттестация (часы)
1 семестр	108	3	36	72	Зачет с оценкой
Всего	108	3	36	72	

5. Содержание дисциплины (модуля)

5.1. Разделы, темы дисциплины и виды занятий

(часы промежуточной аттестации не указываются)

Наименование раздела, темы		ческие занятия	оятельная работа	уемые результаты ия, соотнесенные с атами освоения ммы

	Всего	Практи	Самост	Планир обучени результ програм
Раздел 1. Введение в программирование	**16**	**4**	**12**	ОПК-7.1
Тема 1.1. Введение в программирование	16	4	12	
Раздел 2. Синтаксис Python	**18**	**6**	**12**	ОПК-7.1 ОПК-7.2
Тема 2.1. Синтаксис Python	18	6	12	
Раздел 3. Классы, интерфейсы, пакеты	**18**	**6**	**12**	ОПК-7.1 ОПК-7.2 ОПК-7.3
Тема 3.1. Классы, интерфейсы, пакеты	18	6	12	
Раздел 4. Обработка исключений	**16**	**6**	**10**	ОПК-7.2 ОПК-7.3
Тема 4.1. Обработка исключений	16	6	10	
Раздел 5. Ввод-вывод	**22**	**8**	**14**	ОПК-7.2 ОПК-7.3 ОПК-7.4
Тема 5.1. Ввод-вывод	22	8	14	
Раздел 6. Параметризация типов в Python	**18**	**6**	**12**	ОПК-7.2 ОПК-7.3 ОПК-7.4
Тема 6.1. Параметризация типов в Python	18	6	12	
Итого	**108**	**36**	**72**	

5.2. Содержание разделов, тем дисциплин и формы текущего контроля

Раздел 1. Введение в программирование

Тема 1.1. Введение в программирование

Методологические основы процесса решения задач на ПК. Виды алгоритмов. Основные алгоритмические структуры. Алгоритм разветвляющейся структуры. Алгоритм циклической структуры. Алгоритмы цикла с предусловием, цикла с постусловием. Алгоритм цикла со счетчиком.

Текущий контроль

Вид (форма) контроля
Тестирование
Практическая работа

Раздел 2. Синтаксис Python

Тема 2.1. Синтаксис Python

Основные понятия программирования на языке Python. Принципы работы с интерпретатором языка. Структура программы. Типы данных, операции, операторы.

Текущий контроль

Вид (форма) контроля
Тестирование

Практическая работа

Раздел 3. Классы, интерфейсы, пакеты

Тема 3.1. Классы, интерфейсы, пакеты

Классы. Описание класса. Описание объектов. Конструкторы объектов. Статические элементы класса. Дружественные функции. Деструкторы.

Текущий контроль

Вид (форма) контроля
Тестирование
Практическая работа

Раздел 4. Обработка исключений

Тема 4.1. Обработка исключений

Наследование. Шаблоны классов и обработка исключительных ситуаций.

Текущий контроль

Вид (форма) контроля
Тестирование
Практическая работа

Раздел 5. Ввод-вывод

Тема 5.1. Ввод-вывод

Основные операторы циклического выполнения в языке Python. Вложенные циклические структуры. операторы прерывания цикла. Функция в языке Python. Определение пользовательских функций. Файловый ввод -вывод. Чтение строк с помощью итераторов файлов.

Текущий контроль

Вид (форма) контроля
Тестирование
Практическая работа

Раздел 6. Параметризация типов в Python

Тема 6.1. Параметризация типов в Python

Формы наследования. Формы вложения.

Текущий контроль

Вид (форма) контроля
Тестирование
Практическая работа

6. Порядок проведения промежуточной аттестации

Порядок проведения промежуточной аттестации регламентируется локальными нормативными актами Университета:
- Положением о формах, периодичности, порядке текущего контроля успеваемости и промежуточной аттестации обучающихся по основным образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры в федеральном государственном бюджетном образовательном учреждении высшего образования «Московский государственный университет геодезии и картографии», п.6 «Организация и проведение промежуточной аттестации».
- Положением о порядке осуществления образовательного процесса, проведения текущего контроля успеваемости и промежуточной аттестации обучающихся с применением электронного обучения и дистанционных образовательных технологий в федеральном государственном бюджетном образовательном учреждении высшего образования «Московский государственный университет геодезии и картографии», п.8 «Проведение промежуточной аттестации с применением ЭО и ДОТ».

Промежуточная аттестация: Зачет с оценкой, 1 семестр.
Промежуточная аттестация по дисциплине проводится в форме «дифференцированный зачет». Конкретная форма проведения — компьютерное тестирование по теоретическому материалу или устное собеседование по вопросам билетов — утверждается кафедрой и доводится до сведения обучающихся не позднее чем за месяц до начала промежуточной аттестации. При компьютерном тестировании итоговый результат определяется как отношение количества набранных баллов к максимально возможному количеству баллов. При устном собеседовании билет включает вопросы из перечня вопросов; преподаватель вправе задать дополнительные и уточняющие вопросы.

Критерии оценивания при проведении промежуточной аттестации в форме компьютерного тестирования:
Зачет оценивается по пятибалльной шкале:
- зачтено с оценкой «отлично» выставляется, если обучающийся набрал 86–100 баллов и продемонстрировал системное и полное освоение содержания дисциплины; ответы правильные, обоснованные и терминологически точные;
- зачтено с оценкой «хорошо» выставляется обучающемуся, набравшему 71–85 баллов, в случае, если основное содержание освоено, но допущены отдельные неточности, не влияющие на общее понимание материала;
- зачтено с оценкой «удовлетворительно» выставляется обучающемуся, набравшему 56–70 баллов, если освоены базовые понятия и положения, но имеются существенные пробелы и ошибки;
- не зачтено («не удовлетворительно») – если обучающийся набрал 0–55 баллов, не освоен необходимый минимум содержания дисциплины; результаты обучения не достигнуты.

Критерии оценивания при проведении промежуточной аттестации в форме устного собеседования по билетам:
Зачет оценивается по пятибалльной шкале:
- зачтено с оценкой «отлично» выставляется, если обучающийся набрал 86–100 баллов и дал полные, правильные и логически последовательные ответы; терминология применяется корректно, приведены необходимые обоснования и примеры, дополнительные вопросы раскрыты;
- зачтено с оценкой «хорошо» выставляется обучающемуся, набравшему 71–85 баллов, если вопросы раскрыты правильно и достаточно полно, а отдельные неточности исправлены после уточняющих вопросов;
- зачтено с оценкой «удовлетворительно» выставляется обучающемуся, набравшему 56–70 баллов, если основные положения раскрыты фрагментарно, имеются существенные неточности, однако базовое понимание продемонстрировано;
- не зачтено («не удовлетворительно») – если обучающийся набрал 0–55 баллов, вопросы не раскрыты либо допущены принципиальные ошибки, отсутствует владение базовой

7. Оценочные материалы текущего контроля

Раздел 1. Введение в программирование

Контролируемые ИДК: ОПК-7.1

Тема 1.1. Введение в программирование

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Какой результат выведет следующий код?

```
x = 5
y = "5"
print(x == y)
```

1) True
2) False
3) Error
4) None

2. Что такое изменяемый (mutable) тип данных в Python?

1) int
2) str
3) list
4) tuple

3. Какой индекс у последнего элемента списка [1, 2, 3, 4, 5] ?

1) 5
2) 4
3) -1
4) Оба варианта b и c верны

4. Что выведет print("Hello"[1:4]) ?

1) "Hel"
2) "ell"
3) "ello"
4) "Hell"

5. Какой оператор используется для проверки принадлежности элемента списку?

1) ==
2) is
3) in
4) has

Форма контроля/оценочное средство: Практическая работа

Вопросы/Задания:

1. Формализация задачи и выбор решения

По описанию небольшого реестра заявок выделить входы, результаты, ограничения и проверяемые требования. Разделить обработку на подзадачи, сравнить два решения по ожидаемым времени, памяти и заданным учебным затратам; объяснить выбор без вымышленных рыночных цен

Раздел 2. Синтаксис Python

Контролируемые ИДК: ОПК-7.1 ОПК-7.2

Тема 2.1. Синтаксис Python

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Что выведет следующий код?

```
for i in range(3):
 print(i)
```

1) 1 2 3
2) 0 1 2
3) 1 2
4) 0 1 2 3

2. Как правильно определить функцию в Python?

1) function myFunc():
2) def myFunc():
3) func myFunc():
4) define myFunc():

3. Что выведет print(len("Python")) ?

1) 5
2) 6
3) 7
4) Error

4. Какой тип данных возвращает функция input() ?

1) int
2) float
3) str
4) bool

5. Что выведет print(3 // 2) ?

1) 1.5
2) 1
3) 2
4) Error

Форма контроля/оценочное средство: Практическая работа

Вопросы/Задания:

1. Коллекции для реестра

Для небольшого реестра выбрать список, кортеж, множество и словарь по разным задачам: последовательность записей, неизменяемая пара, уникальные коды и поиск по ключу. Реализовать операции, проверить повторы и отсутствие ключа

Раздел 3. Классы, интерфейсы, пакеты

Контролируемые ИДК: ОПК-7.1 ОПК-7.2 ОПК-7.3

Тема 3.1. Классы, интерфейсы, пакеты

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Какая конструкция используется для обработки исключений?

1) if-else
2) try-except
3) switch-case
4) while-else

2. Что такое self в методах класса?

1) Ключевое слово
2) Ссылка на экземпляр класса
3) Глобальная переменная
4) Имя функции

3. Какой метод вызывается при создании объекта класса?

1) create

2) new
3) init
4) start

4. Как открыть файл для чтения в Python?

1) file = open("file.txt", "r")
2) file = read("file.txt")
3) file = open("file.txt", "read")
4) file = file.open("file.txt")

5. Что выведет следующий код?

```
def func(x=[]):
 x.append(1)
 return x
print(len(func()))
print(len(func()))
```

1) 1, 1
2) 1, 2
3) 0, 1
4) Error

Форма контроля/оценочное средство: Практическая работа

Вопросы/Задания:

1. Классы и простой паттерн

Создать классы двух видов учебного отчета с общим способом получения представления. Выделить функцию-фабрику выбора нужного объекта, объяснить ее назначение и проверить независимость состояния двух экземпляров

Раздел 4. Обработка исключений

Контролируемые ИДК: ОПК-7.2 ОПК-7.3

Тема 4.1. Обработка исключений

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Какая функция преобразует список в итератор?

1) list()
2) iter()
3) next()
4) range()

2. Что такое lambda в Python?

1) Ключевое слово цикла
2) Анонимная функция
3) Тип данных
4) Оператор

3. Какой модуль используется для работы с регулярными выражениями?

1) regex
2) re
3) regexp
4) pattern

4. Что выведет print([1, 2, 3] + [4, 5]) ?

1) [1, 2, 3, 4, 5]
2) [5, 7, 8]
3) Error
4) [1, 2, 3], [4, 5]

5. Как создать пустой словарь?

1) dict = []
2) dict = {}
3) dict = ()
4) dict = set()

Форма контроля/оценочное средство: Практическая работа

Вопросы/Задания:

1. Разбор ошибок и исключений

В трех коротких фрагментах различить синтаксическую, логическую ошибку и ошибку выполнения. Исправить причины, предусмотреть проверку недопустимого ввода и понятное сообщение; показать, когда исключение обрабатывается, а когда передается вызывающему коду

Раздел 5. Ввод-вывод

Контролируемые ИДК: ОПК-7.2 ОПК-7.3 ОПК-7.4

Тема 5.1. Ввод-вывод

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Что такое GIL в Python?

1) Графический интерфейс
2) Global Interpreter Lock
3) Garbage Collector
4) Generic Iterator Library

2. Какой декоратор делает метод статическим?

1) @static
2) @staticmethod
3) @classmethod
4) @property

3. Что выведет следующий код?

```
class A:
 x = 1
a1 = A()
a2 = A()
A.x = 2
print(a1.x, a2.x)
```

1) 1, 1
2) 2, 2
3) 1, 2
4) 2, 1

4. Какой модуль используется для создания виртуального окружения?

1) virtualenv
2) venv
3) pyenv
4) env

5. Что такое comprehension в Python?

1) Способ создания коллекций
2) Метод сжатия
3) Тип данных
4) Функция

Форма контроля/оценочное средство: Практическая работа

Вопросы/Задания:

1. Текстовый ввод-вывод и JSON

Прочитать учебные записи в подтвержденной кодировке, сформировать результат и сохранить его в JSON. Использовать контекстный менеджер, проверить отсутствие файла, пустой вход и успешное повторное чтение сохраненного результата

Раздел 6. Параметризация типов в Python

Контролируемые ИДК: ОПК-7.2 ОПК-7.3 ОПК-7.4

Тема 6.1. Параметризация типов в Python

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Как правильно импортировать конкретную функцию из модуля?

1) import module.function
2) from module import function
3) import function from module
4) using module import function

2. Что выведет `print(2 3 2)`?

1) 64
2) 512
3) 18
4) Error

3. Какой протокол должен реализовывать объект для использования в цикле for?

1) Iterable
2) Callable
3) Hashable
4) Comparable

4. Что такое monkey patching?

1) Исправление ошибок
2) Динамическое изменение кода во время выполнения
3) Тестирование
4) Отладка

5. Какой магический метод отвечает за преобразование объекта в строку?

1) string
2) str
3) text
4) format

Форма контроля/оценочное средство: Практическая работа

Вопросы/Задания:

1. Воспроизводимая проверка изменений

Для учебного репозитория зафиксировать зависимости и настроить короткую задачу CI с запуском pytest. Показать неуспех на намеренной ошибке и успех после исправления, связав каждый результат с коммитом; передать коллеге понятный порядок воспроизведения

8. Оценочные материалы промежуточной аттестации

1 семестр, Зачет с оценкой

Контролируемые ИДК: ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-7.4

Вопросы/Задания:

1. Что такое переменная в языке программирования Python и как она объявляется?

2. Перечислите основные типы данных в Python и приведите примеры.

3. Объясните разницу между списком (list) и кортежем (tuple) в Python.

4. Что такое индексация в Python? Как работает отрицательная индексация?

5. Опишите основные операторы сравнения в Python.

6. Объясните работу цикла for в Python.

7. Что такое срезы (slicing) в Python? Приведите примеры.

8. В чем разница между циклами for и while?

9. Что такое функция в Python? Как она определяется и вызывается?

10. Объясните понятие области видимости переменных в Python.

11. Что такое исключения в Python? Как их обрабатывать?

12. Объясните принципы объектно-ориентированного программирования.

13. Что такое класс и объект в Python? Приведите пример.

14. Объясните работу с файлами в Python.

15. Что такое модули и пакеты в Python?

16. Объясните понятие итератора и генератора.

17. Что такое декораторы в Python?

18. Объясните работу с многопоточностью в Python.

19. Что такое lambda-функции в Python?

20. Объясните работу с регулярными выражениями в Python.

21. Объясните принцип работы сборщика мусора в Python.

22. Что такое метаклассы в Python?

23. Объясните паттерны проектирования Singleton и Factory в Python.

24. Что такое контекстные менеджеры в Python?

25. Объясните принципы тестирования в Python.

26. Что такое виртуальное окружение и зачем оно нужно?

27. Объясните работу с базами данных в Python.

28. Что такое асинхронное программирование в Python?

29. Объясните принципы оптимизации производительности Python-программ.

30. Что такое сериализация данных? Какие форматы поддерживает Python?

9. Материально-техническое и учебно-методическое обеспечение дисциплины

9.1. Перечень основной и дополнительной учебной литературы

Основная литература

1. Советов Б. Я. Информационные технологии: теоретические основы: учебное пособие для вузов / Б. Я. Советов, В. В. Цехановский. - 3-е изд., стер. - Санкт-Петербург: Лань, 2022. - 444 с. - 978-5-507-45305-4. - Текст: электронный // Издательство Лань: [сайт]. - URL: https://e.lanbook.com/book/264935 - Режим доступа: по подписке

Дополнительная литература

1. Цветкова А. В. Информатика и информационные технологии: учебное пособие / А. В. Цветкова. - Информатика и информационные технологии - Саратов: Научная книга, 2012. - 189 с. - 2227-8397. - Текст: электронный // IPR SMART: [сайт]. - URL: https://www.iprbookshop.ru/6276.html - Режим доступа: по подписке

9.2. Профессиональные базы данных и ресурсы «Интернет», к которым обеспечивается доступ обучающихся

Профессиональные базы данных

1. https://student2.consultant.ru/cgi/online.cgi - Правовой Сервер КонсультантПлюс

Ресурсы «Интернет»

1. https://intuit.ru/ - Электронная образовательная среда "ИНТУИТ"
2. https://znanium.ru/ - Электронно-библиотечная система "Znanium"
3. https://www.elibrary.ru/ - НЭБ eLIBRARY
4. https://rusneb.ru/ - Национальная электронная библиотека (НЭБ)

9.3. Программное обеспечение и информационно-справочные системы, используемые при осуществлении образовательного процесса по дисциплине

Перечень программного обеспечения
(обновление производится по мере появления новых версий программы)

1. Sublime Text 3;
2. Microsoft Visual Studio;

Перечень информационно-справочных систем
(обновление выполняется еженедельно)

Не используется.

9.4. Оборудование и технические средства обучения

Компьютерный класс

Маркерная доска - 0 шт.

Моноблок - 96 шт.
Проектор - 6 шт.

10. Методические указания по освоению дисциплины (модуля)

С целью успешного освоения данного предмета студенту необходимо выполнять домашние работы по указанным преподавателям темам и представленным на занятиях материалам. Для более успешного выполнения работы студенту предлагается использование учебно-методических материалов, соответствующих указанной теме.

Самостоятельная работа студентов описывается и регулируется методическими рекомендациями по дисциплине.

Описание возможностей изучения дисциплины лицами с ОВЗ и инвалидами

При необходимости обучения студентов-инвалидов и лиц с ограниченными возможностями здоровья аудиторные занятия могут быть заменены или дополнены изучением полнотекстовых лекций, презентаций, видео- и аудиоматериалов в электронной информационно-образовательной среде (ЭИОС) университета. Индивидуальные задания подбираются в адаптированных к ограничениям здоровья формах (письменно или устно, в форме презентаций). Выбор методов обучения зависит от их доступности для инвалидов и лиц с ограниченными возможностями здоровья.

Рабочая программа дисциплины для инвалидов и лиц с ограниченными возможностями здоровья разрабатывается индивидуально с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.