

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

УТВЕРЖДЕНО
Заместитель директора

Ю.О. Соболев

	Рабочая программа дисциплины (модуля)
по дисциплине:	Язык Python для разработчиков
по направлению:	Информатика и вычислительная техника
профиль подготовки:	Разработка IT-продукта
	центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск"
	центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск"
курс:	1
квалификация:	магистр

Семестр, формы промежуточной аттестации: 1 (осенний) - Экзамен

Аудиторных часов: 36 всего, в том числе:

лекции: 14 час.

семинары: 22 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 204 час.

Подготовка к экзамену: 30 час.

Всего часов: 270, всего зач. ед.: 6

Программу составил: К.А. Лапин, старший методист

Программа обсуждена на заседании центра дополнительного, дополнительного профессионального и
онлайн-образования "Пуск" 01.03.2025

Аннотация

Дисциплина является начальным курсом для развития знаний и навыков в области программирования и алгоритмизации. Знание основ программирования позволит студентам сформировать для себя требования к современной парадигме разработки программного обеспечения. Дисциплина предполагает дать студентам основные понятия языка программирования Python, научить решать основные программные задачи, отлаживать и тестировать программы. Таким образом, студенты приобретут навыки, необходимые как для изучения последующих дисциплин образовательной программы, так и для работы в области бэкенд-разработки.

1. Цели и задачи

Цель дисциплины

- развитие навыков программирования на языке Python и их практическое применение, формирование навыка самостоятельного выполнения поставленных задач.

Задачи дисциплины

- изучение основных конструкций языка Python;
- освоение принципов объектно-ориентированного программирования;
- изучение специализированных сервисов и фреймворков Python для бэкенд-разработки;
- овладение навыками работы с данными;
- создание и использование функций для структурирования кода;
- ознакомление с системой контроля версий.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий	УК-1.2 Осуществляет поиск вариантов решения поставленной проблемной ситуации на основе доступных источников информации
УК-6 Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки	УК-6.1 Умеет решать задачи собственного личностного и профессионального развития, определять и реализовывать приоритеты совершенствования собственной деятельности
ОПК-5 Способен и готов к профессиональному росту и руководству коллективом в области информатики и вычислительной техники, толерантно воспринимая социальные, этнические, конфессиональные и культурные различия	ОПК-5.3 Стремится к получению новых знаний, профессиональному и личностному росту
ПК-4 Способен проектировать информационные системы, их компоненты и протоколы их взаимодействия, собирать, обрабатывать и интерпретировать экспериментальные данные, необходимые для проектной и производственно-технологической деятельности; способен к разработке новых алгоритмических, методических и технологических решений в конкретной сфере профессиональной деятельности	ПК-4.1 Знает основы проектирования и элементы архитектурных решений информационных систем
	ПК-4.2 Умеет применять в практической деятельности профессиональные стандарты в области информационных технологий
	ПК-4.3 Имеет опыт оценки качества, надежности и эффективности информационной системы в конкретной профессиональной сфере
ПК-6 Способен применять в профессиональной деятельности	ПК-6.1 Знает современные языки программирования и методы параллельной обработки данных

профессиональной деятельности
современные языки программирования и
программное обеспечение; операционные
системы и сетевые технологии, электронные
библиотеки и пакеты программ

ПК-6.2 Умеет реализовывать и применять численные
методы решения прикладных задач в профессиональной
сфере деятельности, используя пакеты программного
обеспечения, операционные системы, электронные
библиотеки, сетевые технологии

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- основные конструкции языка Python;
- принципы объектно-ориентированного программирования;
- специализированные сервисы и фреймворки Python, которые используют в бэкенд-разработке.

уметь:

- работать с данными: чтение, запись и обработка файлов различных форматов и структур данных;
- разрабатывать программы на Python для решения практических задач;
- создавать и использовать функции для структурирования кода и повторного использования, использовать классы в разработке программного кода;
- работать с системой контроля версий и репозиториями.

владеть:

- навыками разработки на Python.

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	Язык Python для разработки. Введение	2			30
2	Git — система контроля версий. Работа на GitHub	1	1		30
3	Исключения и обработка ошибок	1	1		28
4	Файлы и структуры данных	1	1		30
5	Параллельные вычисления. Многопоточность и многопроцессность	1	1		28
6	Классы и объекты (ООП)	2	4		30
7	Аннотации типов и валидация данных	2			28
8	Работа с API и веб-сервисами	2	2		
9	Фреймворки веб-разработки	2	12		
Итого часов		14	22		204
Подготовка к экзамену		30 час.			
Общая трудоёмкость		270 час., 6 зач.ед.			

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

Семестр: 1 (Осенний)

1. Язык Python для разработки. Введение

Введение в веб-разработку. Основные концепции и инструменты веб-разработки на Python.

2. Git — система контроля версий. Работа на GitHub

Работа на GitHub.

Основные команды и функциональность Git.

3. Исключения и обработка ошибок

Введение в обработку ошибок и исключения в Python. Обработка исключений с помощью try-except. Создание собственных исключений и их обработка. Управление потоком выполнения при возникновении ошибок.

4. Файлы и структуры данных

Работа с файлами и папками в Python. Текстовые и бинарные файлы. Сериализация и десериализация данных. Различные структуры данных в Python (списки, кортежи, словари).

5. Параллельные вычисления. Многопоточность и многопроцессность

Понятие параллельных вычислений и их применение. Многопоточность и многопроцессность в Python. Практические примеры параллельных вычислений на Python.

6. Классы и объекты (ООП)

Основы объектно-ориентированного программирования (ООП) в Python. Создание классов и объектов, работа с атрибутами и методами. Наследование, полиморфизм и инкапсуляция в ООП.

7. Аннотации типов и валидация данных

Аннотация типов: понятие и преимущества. Технологии для валидации данных: Pydantic, attrs, dataclasses. Использование регулярных выражений для валидации данных.

8. Работа с API и веб-сервисами

Базовое устройство запросов. API сторонних сервисов: понятие и назначение. Отправка запросов с помощью библиотеки requests. Создание API с помощью FastAPI. Настройка взаимодействия с внешними сервисами и API различных сервисов.

9. Фреймворки веб-разработки

Основные современные фреймворки веб-разработки на Python: FastAPI, Flask, Django. Использование фреймворка FastAPI для веб-разработки на Python.

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

Занятия по учебной дисциплине проводятся с применением дистанционных образовательных технологий. Каждый обучающийся обеспечен доступом к образовательной платформе (LMS).

6. Перечень рекомендуемой литературы

Основная литература

1. Язык программирования PYTHON [Текст] : учеб. пособие для вузов / Р. А. Сузи .— 2 изд., испр. — М. : Интернет-Ун-т Информ. Технологий : БИНОМ. Лаб. знаний, 2007 .— 326 с.

Дополнительная литература

Литература из средств кафедр:

Чернышев, С. А. Основы программирования на Python: учебное пособие для вузов / С. А. Чернышев. — Москва: Издательство Юрайт, 2023. — 286 с. — (Высшее образование). — ISBN 978-5-534-14350-8. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/519949>

Жуков, Р. А. Язык программирования Python: практикум: учебное пособие / Р.А. Жуков. — Москва: ИНФРА-М, 2023. — 216 с. + Доп. материалы [Электронный ресурс]. — (Высшее образование: Бакалавриат). — DOI 10.12737/textbook_5cb5ca35aaa7f5.89424805. - ISBN 978-5-16-016971-2. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1915716> – Режим доступа: по подписке.

Федоров, Д. Ю. Программирование на языке высокого уровня Python: учебное пособие для вузов / Д. Ю. Федоров. — 5-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2024. — 227 с. — (Высшее образование). — ISBN 978-5-534-17323-9. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/539651>

Карякин, М. И. Технологии программирования и компьютерный практикум на языке Python: учебное пособие: [16+] / М. И. Карякин, К. А. Ватульян, Р. М. Мнухин; Южный федеральный университет. – Ростов-на-Дону; Таганрог: Южный федеральный университет, 2022. – 244 с.: ил., табл. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=698687> – Библиогр. в кн. – ISBN 978-5-9275-4108-9. – Текст: электронный.

Копырин, А. С. Программирование на Python: учебное пособие / А. С. Копырин, Т. Л. Салова. — Москва: ФЛИНТА, 2021. — 48 с. — ISBN 978-5-9765-4753-7. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book>

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

1. Официальный сайт Python - <https://www.python.org/>
2. Официальная документация (англ.): <https://docs.python.org/3/>

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

Студенту для занятий потребуются:

1. Google Drive / Yandex disk для доступа к материалам курса
2. Zoom
3. LMS МФТИ
4. Приложение для коммуникации с преподавателями
5. Ноутбук для участия в интерактивных занятиях

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. В результате изучения дисциплины студент должен знать основные определения дисциплины, уметь применять полученные знания для решения прикладных задач.

Успешное освоение курса требует:

- посещения всех онлайн-занятий, предусмотренных учебным планом по дисциплине;
- ведения конспекта занятий;
- напряжённой самостоятельной работы студента.

Самостоятельная работа включает в себя:

- чтение материалов на платформе дистанционного обучения и рекомендованной литературы;

- проработку учебного материала, подготовку ответов на вопросы, предназначенных для самостоятельного изучения;
- решение задач, предлагаемых студентам на занятиях;
- подготовку к выполнению заданий текущей и промежуточной аттестации.

Показателем владения материалом служит умение без вспомогательных материалов и конспектов отвечать на вопросы по темам дисциплины.

Важно добиться понимания изучаемого материала, а не механического его запоминания. При затруднении изучения отдельных тем, вопросов, следует обращаться за консультациями к преподавателю.

Возможен текущий контроль знаний студентов в виде решения задач в соответствии с тематикой занятий.

ПРИЛОЖЕНИЕ

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Информатика и вычислительная техника
профиль подготовки:	Разработка IT-продукта центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск" центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск"
курс:	<u>1</u>
квалификация:	магистр
Семестр, формы промежуточной аттестации: 1 (осенний) - Экзамен	
Разработчик:	К.А. Лапин, старший методист

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий	УК-1.2 Осуществляет поиск вариантов решения поставленной проблемной ситуации на основе доступных источников информации
УК-6 Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки	УК-6.1 Умеет решать задачи собственного личностного и профессионального развития, определять и реализовывать приоритеты совершенствования собственной деятельности
ОПК-5 Способен и готов к профессиональному росту и руководству коллективом в области информатики и вычислительной техники, толерантно воспринимая социальные, этнические, конфессиональные и культурные различия	ОПК-5.3 Стремится к получению новых знаний, профессиональному и личностному росту
ПК-4 Способен проектировать информационные системы, их компоненты и протоколы их взаимодействия, собирать, обрабатывать и интерпретировать экспериментальные данные, необходимые для проектной и производственно-технологической деятельности; способен к разработке новых алгоритмических, методических и технологических решений в конкретной сфере профессиональной деятельности	ПК-4.1 Знает основы проектирования и элементы архитектурных решений информационных систем
	ПК-4.2 Умеет применять в практической деятельности профессиональные стандарты в области информационных технологий
	ПК-4.3 Имеет опыт оценки качества, надежности и эффективности информационной системы в конкретной профессиональной сфере
ПК-6 Способен применять в профессиональной деятельности современные языки программирования и программное обеспечение; операционные системы и сетевые технологии, электронные библиотеки и пакеты программ	ПК-6.1 Знает современные языки программирования и методы параллельной обработки данных
	ПК-6.2 Умеет реализовывать и применять численные методы решения прикладных задач в профессиональной сфере деятельности, используя пакеты программного обеспечения, операционные системы, электронные библиотеки, сетевые технологии

2. Показатели оценивания компетенций

В результате изучения дисциплины «Язык Python для разработчиков» обучающийся должен:

знать:

- основные конструкции языка Python;
- принципы объектно-ориентированного программирования;
- специализированные сервисы и фреймворки Python, которые используют в бэкенд-разработке.

уметь:

- работать с данными: чтение, запись и обработка файлов различных форматов и структур данных;
- разрабатывать программы на Python для решения практических задач;
- создавать и использовать функции для структурирования кода и повторного использования, использовать классы в разработке программного кода;
- работать с системой контроля версий и репозиториями.

владеть:

- навыками разработки на Python.

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

В начале занятия проводится опрос по материалам предыдущего занятия и даются задачи для совместного решения.

Пример задания:

1.

Перед выполнением задания установите Jupyter Notebook или используйте Google Colab.

Печатные газеты использовали свой формат дат для каждого выпуска. Для каждой газеты из списка напишите формат указанной даты для перевода в объект `datetime`:

The Moscow Times — Wednesday, October 2, 2002

The Guardian — Friday, 11.10.13

Daily News — Thursday, 18 August 1977

Пример работы программы

Программа должна выводить на экран объекты типа `datetime`, соответствующие датам в условии задачи.

Также программа должна работать итерационно, а именно: принимать входные данные до момента ввода специального символа завершения программы.

Если введённое значение не соответствует ни одному формату, то следует продолжать выполнение программы, переходя на следующую итерацию.

Безопасный переход следует реализовать с помощью механизма исключений.

Критерии выполнения задания:

Установлен Jupyter Notebook или используется Google Colab

Создан профиль на GitHub, если используется Jupyter Notebook |

Программа выводит на экран объекты типа `datetime`, соответствующие датам в условии задачи |

2.

Цель: разработать систему, которая будет формировать описания покупателей на основе характеристик, представленных в файле.

На входе — CSV-файл с характеристиками покупателей.

На выходе — TXT-файл с описаниями.

Порядок выполнения задания.

1. Загрузить CSV-файл.
2. Выполнить парсинг (извлечение атрибутов из структурированного текста).
3. Преобразовать данные (при необходимости).
4. Сформировать текстовое описание по шаблону.
5. Записать в единый TXT-файл.

Пример строки на входе:

ФИО: Allen Miss. Elisabeth Walton

Пол: female

Возраст: 29

Устройство, с которого выполнялась покупка: mobile

Браузер: Chrome

Сумма чека: 885

Регион покупки: St Louis: MO

Пример строки на выходе:

Пользователь Allen Miss. Elisabeth Walton женского пола, 29 лет совершила покупку на 885 у.е. с мобильного браузера Chrome. Регион, из которого совершалась покупка: St Louis: MO.

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

Примеры заданий с автоматизированной проверкой на экзамен:

1. К какому виду языков относится Python?

- Интерпретируемый
- Компилируемый
- Промежуточно-компилируемый
- Низкоуровневый

2. Чем отличаются многопроцессные и многопроцессорные вычисления?

- Отличий нет
- В многопроцессных вычислениях может использоваться только один процессор, а в многопроцессорных вычислениях всегда может использоваться два процессора и более
- В многопроцессорных вычислениях может использоваться только один процессор, а в многопроцессных вычислениях может использоваться два процессора и более
- Многопроцессорных вычислений не существует

3. Чем класс отличается от объекта?

- Отличий нет, это синонимы
- Объект является экземпляром класса
- Класс является экземпляром объекта
- Класс и объект — разные языковые конструкции без взаимосвязи

4. Чем библиотека threading отличается от multiprocessing?

- threading — библиотека для работы с потоками, multiprocessing — библиотека для работы с разными процессами
- threading — библиотека для работы с процессами, multiprocessing — библиотека для работы потоками
- threading и multiprocessing — синонимы одной библиотеки
- threading — библиотека для работы со строками, multiprocessing — библиотека для работы с числами

5. Чем функции отличаются от процедур?

- Функция всегда возвращает результат (return), процедура иногда не возвращает результат
- Функция является производной процедуры
- Функция всегда возвращает результат (return), процедура всегда не возвращает результат
- Функция и процедура — синонимы и отличаются только названием

6. Что такое рекурсия?

- Метод решения задачи, при котором используются итеративные циклы вместо рекурсивных вызовов
- Мыслительный процесс, направленный на решение системы линейных уравнений
- Принцип, который определяет, что некоторая сущность должна вызвать саму себя
- Способ самоочистки программы

7. Как организована область видимости переменных в Python?

- Любая переменная доступна только 10 секунд после инициализации
- Созданная переменная доступна во всём пространстве функции, даже если она была инициализирована во вложенном if, while, for и т. д.
- Созданная переменная доступна только в том пространстве if, while, for и т. д., в котором она была инициализирована
- Переменные в Python не имеют области видимости

8. В чём принципиальное отличие между while и for?

- while — счётный, for — итерационный
- while и for не отличаются
- while — итерационный, for — счётный
- while не является циклом, а for является

9. Можно ли обойтись без try .. Except в языке программирования?

- Без try .. Except можно спокойно обойтись: есть конструкции if, которые позволят описать все виды исключительных ситуаций
- Try .. Except — принципиально иной вид конструкций, который нельзя полноценно заменить какой-либо другой технологией
- Наверное. Не понимаю, как работает try .. Except
- try .. Except только усложняет код и не является необходимым для обработки исключений

10. Можно ли изменить передаваемый аргумент внутри функции в Python?

- Да, внутри функции изменять аргумент можно, только если они являются простыми — immutable
- Нет, внутри функции изменять нельзя
- Да, внутри функции изменять аргументы можно, только если они являются сложными — mutable (dict, list, etc)
- Да, внутри функции можно изменять аргументы независимо от их типа данных

Критерии оценивания

Оценка отлично (10 баллов) - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины, проявляющему интерес к данной предметной области, продемонстрировавшему умение уверенно и творчески применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично (9 баллов) - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично (8 баллов) - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений, с некоторыми недочетами.

Оценка хорошо (7 баллов) - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но недостаточно грамотно обосновывает полученные результаты.

Оценка хорошо (6 баллов) - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности.

Оценка хорошо (5 баллов) - выставляется студенту, если он в основном знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач достаточно большое количество неточностей.

Оценка удовлетворительно (4 балла) - выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он освоил основные разделы учебной программы, необходимые для дальнейшего обучения, и может применять полученные знания по образцу в стандартной ситуации.

Оценка удовлетворительно (3 балла) - выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, допускающему ошибки в формулировках базовых понятий, нарушения логической последовательности в изложении программного материала, слабо владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и с трудом применяет полученные знания даже в стандартной ситуации.

Оценка неудовлетворительно (2 балла) - выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных принципов и не умеет использовать полученные знания при решении типовых задач.

Оценка неудовлетворительно (1 балл) - выставляется студенту, который не знает основного содержания учебной программы дисциплины, допускает грубейшие ошибки в формулировках базовых понятий дисциплины и вообще не имеет навыков решения типовых практических задач.

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

Экзамен проходит в письменном формате, на lms платформе.

Время проведения письменного экзамена оставляет 2 академических часа.

Во время проведения экзамена обучающиеся могут пользоваться программой дисциплины.