

**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**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Программирование на языке Python
<b>по направлению:</b>	Прикладная математика и информатика
<b>профиль подготовки:</b>	Искусственный интеллект и большие данные Сетевое обучение кафедра алгоритмов и технологий программирования
<b>курс:</b>	2
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 3 (осенний) - Дифференцированный зачет

Аудиторных часов: 30 всего, в том числе:

лекции: 0 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 78 час.

Всего часов: 108, всего зач. ед.: 3

Программу составил: Д.Е. Беляков, ассистент

Программа обсуждена на заседании кафедры алгоритмов и технологий программирования 30.08.2022

## Аннотация

Курс "Практикум Python" направлен на обучение студентов основам работы с языком программирования Python версии 3.x. Курс разделен на две части. В первой части курса студенты осваивают типы данных и синтаксис языка, углубленные аспекты функционального программирования и ООП. Вторая часть курса знакомит студентов с наиболее популярными библиотеками в Python, используемыми в таких областях, как web-программирование, тестирование, обработка и анализ данных. Для успешного окончания курса необходимо решить ряд практических задач, реализовать 2 курсовых проекта и выполнить лабораторную работу.

### 1. Цели и задачи

#### Цель дисциплины

Познакомить студентов с языком программирования Python и подготовить их к практической деятельности в должностях аналитиков и программистов программного обеспечения.

#### Задачи дисциплины

- \* Сформировать знания о правильном применении языка Python в разработке.
- \* Сформировать знания о популярных библиотеках и фреймворках на Python.

### 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-2 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК-2.1. Применяет знания основных положений и концепций в области программирования, архитектуру языков программирования, основную терминологию и базовые алгоритмы, основные требования
	ОПК-2.2. Анализирует типовые языки программирования, составляет программы
	ОПК-2.3 Применяет на практике опыт решения задач с использованием базовых алгоритмов, анализа типов коммуникаций и интеграции различных типов программного обеспечения

### 3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны знать:

синтаксис языка программирования Python;  
общепринятые способы решения базовых задач с использованием особенностей языка;  
основные библиотеки и фреймворки на Python;  
принцип исполнения программ на Python;  
типы данных языка Python;  
управление потоком выполнения в Python;  
возможности стандартной библиотеки;  
правила работы с исключениями;  
внутреннее строение контейнеров стандартной библиотеки и временную сложность операций с ними;  
принцип работы сборки мусора в Python;  
кодировки, используемые при хранении текстовых данных (ASCII, Windows-1250/1251, UTF-8, UTF-16).

уметь:

реализовывать библиотеку общего назначения на языке Python по заданным интерфейсам;  
решать задачи, связанные с обработкой данных, на языке Python.

владеть:

основными библиотеками и инструментами разработчика на языке 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	Знакомство с Python		2		8
2	Управление вычислениями. Контейнеры, итераторы		2		8
3	Словари, множества. Модуль collections		2		8
4	Функции		3		8
5	Работа со строками. Работа с файлами		3		8
6	ООП (часть 1)		3		8
7	ООП (часть 2)		3		6
8	Работа с сетью		3		6
9	Серверные приложения		3		6
10	Оптимизация кода		3		6
11	Аналитические инструменты		3		6
Итого часов			30		78
Подготовка к экзамену		0 час.			
Общая трудоёмкость		108 час., 3 зач.ед.			

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

Семестр: 3 (Осенний)

##### 1. Знакомство с Python

История языка Python  
 Сравнение Python2 vs Python3  
 Сравнение Python и C/C++  
 Интерпретатор командной строки  
 IDE PyCharm  
 Основы языка  
 Типы данных  
 Переменные, оператор связывания.  
 Арифметика  
 Приведения типов  
 Приведение к bool  
 Булева алгебра  
 Распаковка  
 print/input

##### 2. Управление вычислениями. Контейнеры, итераторы

Условный оператор  
 Тернарный условный оператор  
 Циклы while, for

tuple/list  
range/xrange  
list comprehensions  
generator expressions

### 3. Словари, множества. Модуль collections

Хэшируемость  
Тип dict  
Типы set/frozenset. Операции над множествами  
Модуль collections.  
Сравнение поведения словарей в версиях 3.5- и 3.6+

### 4. Функции

Хэшируемость  
Тип dict  
Типы set/frozenset. Операции над множествами  
Модуль collections.  
Сравнение поведения словарей в версиях 3.5- и 3.6+

### 5. Работа со строками. Работа с файлами

Тип str  
Методы строк  
Форматирование строк: C-style, str.format()  
Модуль string  
Кодировки  
Тип bytes  
Работа с файлами

### 6. ООП (часть 1)

Объявление класса, создание экземпляра  
Атрибуты, методы  
Статические атрибуты и методы  
Наследование. Множественное наследование. Ромбовидное наследование.  
Приватность атрибутов  
Метод `__call__`  
Объектно-ориентированный подход к созданию итераторов и генераторов

### 7. ООП (часть 2)

Методы `__str__` и `__repr__`  
Перегрузка арифметических операций  
Перегрузка приведений к базовым типам  
Динамическая работа с атрибутами  
Контексты. Модуль `contextlib`  
Метод `__new__`

### 8. Работа с сетью

Обзор протокола HTTP:  
Структура URI  
Методы HTTP

Структура ответа, коды состояний  
Обзор языка HTML  
Установка внешних пакетов. PyPi  
Чтение документов из сети:  
Модуль urllib  
Пакет requests  
Парсинг HTML-страниц:  
Обзор регулярных выражений. Модуль re  
Модуль lxml  
Модуль BeautifulSoup

## 9. Серверные приложения

Модуль flask  
Работа с реляционными базами данных:  
PEP 249 (DB API)  
Пакеты psycopg2/sqlite3  
Знакомство ORM: пакеты Peewee и SQLAlchemy

## 10. Оптимизация кода

Знакомство с Jupyter Notebook  
Пакет numru  
Типы данных  
U-functions  
Агрегации  
Сравнение с чистым Python. Модуль timeit  
Знакомство с Cython  
Профилирование кода

## 11. Аналитические инструменты

Пакет matplotlib  
Примитивы  
Сложные графики  
Пакет pandas  
Типы DataFrame, Series  
Понятие индекса  
Агрегационные функции  
Оконные функции  
Интеграция с matplotlib

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

Учебная аудитория, оснащенная компьютерами для каждого студента.

## **6.Перечень рекомендуемой литературы**

Основная литература

1. Основы операционных систем [Текст] : Курс лекций : учеб. пособие для вузов / В. Е. Карпов, К. А. Коньков .— 2-е изд., доп. и испр. — М. : Интернет - Ун-т информац. технологий, 2005 .— 536 с.
2. Алгоритмы : построение и анализ [Текст] / Т. Кормен, Ч. Лейзерсон, Р. Ривест .— М. : МЦНМО, 2001 .— 960 с.

Дополнительная литература  
Не предусмотрено

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

Видеолекции, доступные по ссылке:

<https://www.youtube.com/watch?v=VP2wRhwlg6c&list=PLJOzdkh8T5kpIBTG9mM2wVBjh-5OpdwBl>

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

IDE PyCharm

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

Можно изучать дополнительно материалы похожих курсов:

Основы программирования на Python

<https://www.coursera.org/learn/python-osnovy-programmirovaniya>

Язык программирования Python

<http://www.intuit.ru/studies/courses/49/49/info>

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

<b>по направлению:</b>	Прикладная математика и информатика
<b>профиль подготовки:</b>	Искусственный интеллект и большие данные Сетевое обучение кафедра алгоритмов и технологий программирования
<b>курс:</b>	2
<b>квалификация:</b>	бакалавр
Семестр, формы промежуточной аттестации: 3 (осенний) - Дифференцированный зачет	
<b>Разработчик:</b>	Д.Е. Беляков, ассистент

## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-2 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК-2.1. Применяет знания основных положений и концепций в области программирования, архитектуру языков программирования, основную терминологию и базовые алгоритмы, основные требования
	ОПК-2.2. Анализирует типовые языки программирования, составляет программы
	ОПК-2.3 Применяет на практике опыт решения задач с использованием базовых алгоритмов, анализа типов коммуникаций и интеграции различных типов программного обеспечения

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Программирование на языке Python» обучающийся должен:

### знать:

синтаксис языка программирования Python;  
общепринятые способы решения базовых задач с использованием особенностей языка;  
основные библиотеки и фреймворки на Python;  
принцип исполнения программ на Python;  
типы данных языка Python;  
управление потоком выполнения в Python;  
возможности стандартной библиотеки;  
правила работы с исключениями;  
внутреннее строение контейнеров стандартной библиотеки и временную сложность операций с ними;  
принцип работы сборки мусора в Python;  
кодировки, используемые при хранении текстовых данных (ASCII, Windows-1250/1251, UTF-8, UTF-16).

### уметь:

реализовывать библиотеку общего назначения на языке Python по заданным интерфейсам;  
решать задачи, связанные с обработкой данных, на языке Python.

### владеть:

основными библиотеками и инструментами разработчика на языке Python.

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

Примеры контрольных заданий:

1\_3\_2. Дан массив целых чисел  $A[0..n-1]$ . Известно, что на интервале  $[0, m]$  значения массива строго возрастают, а на интервале  $[m, n-1]$  строго убывают. Найти  $m$  за  $O(\log m)$ .  
 $2 \leq n \leq 10000$ .

2\_4\_1. Первые  $k$  элементов длинной последовательности.

Дана очень длинная последовательность целых чисел длины  $n$ . Требуется вывести в отсортированном виде её первые  $k$  элементов. Последовательность может не помещаться в память. Время работы  $O(n * \log(k))$ . Доп. память  $O(k)$ . Использовать слияние.

3\_3\_2. Порядковые статистики. Дано число  $N$  и  $N$  строк. Каждая строка содержит команду добавления или удаления натуральных чисел, а также запрос на получение  $k$ -ой порядковой статистики. Команда добавления числа  $A$  задается положительным числом  $A$ , команда удаления числа  $A$  задается отрицательным числом  $-A$ . Запрос на получение  $k$ -ой порядковой статистики задается числом  $k$ . Требуемая скорость выполнения запроса -  $O(\log n)$ .

4\_4\_1. Самая удаленная вершина.

Для каждой вершины определите расстояние до самой удаленной от нее вершины. Время работы должно быть  $O(n)$ .

Формат входных данных:

В первой строке записано количество вершин  $n \leq 10000$ . Затем следует  $n - 1$  строка, описывающая ребра дерева. Каждое ребро – это два различных целых числа – индексы вершин в диапазоне

$[0, n-1]$ . Индекс корня – 0. В каждом ребре родительской вершиной является та, чей номер меньше.

Формат выходных данных:

Выход должен содержать  $n$  строк. В  $i$ -ой строке выводится расстояние от  $i$ -ой вершины до самой удаленной от нее.

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

1. Среды разработки Python.
2. Сингулярные типы данных.
3. Типы контейнеров.
4. Изменяемые и неизменяемые типы данных.
5. Условные операторы и циклы.
6. Итераторы и генераторы. Отличия реализации.
7. Функциональное программирование. Анонимные функции.
8. Замыкания и декораторы. Декорирование рекурсивных функций.
9. Объектно-ориентированное программирование. Наследование, инкапсуляция, приватность.
10. Специальные методы классов. Менеджеры контекстов.
11. Библиотеки Python, используемые в web-разработке.
12. Библиотеки Python, используемые в анализе данных.

#### Критерии оценивания

отлично

10 всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений;

9 систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений;

8 глубокие знания учебной программы дисциплины и умение применять их на практике при решении конкретных задач, правильное обоснование принятых решений;

хорошо

7 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;

6 знает материал, грамотно излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;

5 знает основной материал, грамотно излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач неточности;

удовлетворительно

4 фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;

3 характер знаний достаточен для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;

неудовлетворительно

2 не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет правильно использовать полученные знания при решении типовых практических задач.

1 не знает формулировок основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

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

Дифференцированный зачет может проводиться по итогам текущей успеваемости и сдачи заданий, лабораторных и других видов работ, предусмотренных программой дисциплины и (или) путем организации специального опроса, проводимого в устной и (или) письменной форме, а также с выдачей заданий для реализации на компьютере.