

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

**УТВЕРЖДЕНО**

**Директор института-заместитель  
директора ФАКТ**

**М.А. Кудров**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Программирование на языке Python
<b>по направлению:</b>	Техническая физика
<b>профиль подготовки:</b>	Радиолокационные технологии Физтех-школа авиационных и цифровых технологий кафедра технологий проектирования сложных технических систем
<b>курс:</b>	1
<b>квалификация:</b>	магистр

Семестры, формы промежуточной аттестации:

1 (осенний) - Дифференцированный зачет

2 (весенний) - Экзамен

Аудиторных часов: 120 всего, в том числе:

лекции: 120 час.

семинары: 0 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 75 час.

Подготовка к экзамену: 30 час.

Всего часов: 225, всего зач. ед.: 5

Программу составил: А.М. Власов

Программа обсуждена на заседании кафедры технологий проектирования сложных технических систем  
12.03.2024

## Аннотация

Язык Python является одним из самых простых, и в то же время востребованных на рынке труда языков программирования. Он используется в различных сферах прикладного применения: от научных вычислений и анализа данных до разработки промышленных решений. Предлагаемый курс является вводным как в язык программирования, так и в используемые связанные с ним технологии для анализа, визуализации и наглядного представления данных.

Курс реализуется с применением электронного обучения и дистанционных образовательных технологий.

### 1. Цели и задачи

#### Цель дисциплины

- обучить слушателей основам программирования на языке Python, необходимым для анализа данных на практике, и познакомить со связанными технологиями анализа, визуализации и наглядного представления данных.

#### Задачи дисциплины

- познакомить студентов с основными конструкциями, объектами и процедурами языка Python;
- сформировать навыки написания эффективного, простого, понятного и гибкого кода, оптимального с точки зрения повышения скорости и качества разработки;
- научить методам обработки ошибок и тестирования кода на языке Python;
- обучить обработке данных с помощью языка Python и основам функционального программирования;
- обучить технологиям анализа и визуализации представления данных.

### 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-3 Способен организовывать и руководить работой команды, вырабатывая командную стратегию для достижения поставленной задачи	УК-3.1 Организует и координирует работу участников проекта, способствует конструктивному преодолению возникающих разногласий и конфликтов
	УК-3.4 Способен планировать командную работу, распределять поручения членам команды, организовать обсуждение разных идей и мнений
ОПК-5 Способен осваивать и применять современные физико-математические методы и методы искусственного интеллекта для решения профессиональных задач, составлять практические рекомендации по использованию полученных результатов	ОПК-5.1 Владеет современными расчетно-теоретическими методами, методами компьютерного моделирования и средами разработки программного обеспечения, применяемыми при решении задач в своей профессиональной области
	ОПК-5.2 Владеет аналитическими и вычислительными методами решения, способен обосновать эффективность выбранного метода

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

В результате освоения дисциплины обучающиеся должны

знать:

- парадигму объектно-ориентированного программирования;
- основные объекты и процедуры языка Python;
- методы обработки ошибок в языке Python;
- основные, используемые на практике, библиотеки.

уметь:

- писать эффективный код, отлаживать и документировать код на языке Python;
- использовать основные библиотеки научных вычислений языка Python для решения типовых прикладных задач в области анализа данных и адаптировать их к своим потребностям в ходе выполнения НИР.

владеть:

- средствами разработки и тестирования программного кода на языке Python;
- объектами и средствами, предлагаемыми стандартными библиотеками научных вычислений языка;
- базовыми принципами проведения вычислений с использованием библиотек 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	12			6
2	Основные инструменты python	12			6
3	Строки	12			6
4	Основные структуры данных в python	12			6
5	Функции. Часть 1	12			6
6	Функции. Часть 2	12			7
7	Классы. Часть 1	12			7
8	Классы. Часть 2	12			8
9	Модули в python	12			8
10	Numpy	6			8
11	Визуализация данных	6			7
Итого часов		120			75
Подготовка к экзамену		30 час.			
Общая трудоёмкость		225 час., 5 зач.ед.			

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

###### Семестр: 1 (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. Основные инструменты python

Условный оператор. Тернарный условный оператор. Циклы while, for. tuple/list. range/xrange. - mutable и immutable.

###### 3. Строки

Копирование объектов. Строки. Индексация и слайсинг. Методы str. Форматирование строк. Регулярные выражения.

#### 4. Основные структуры данных в python

dict и работа с dict. set и методы set. list, dict and set comprehensions. Немного про модули. Модуль collections и Counter (как пример полезного модуля).

#### 5. Функции. Часть 1

- def
- параметры и аргументы функций
- return, рекурсия, стек вызовов
- работа с файлами
- контекстный менеджер with
- модуль os

#### Семестр: 2 (Весенний)

#### 6. Функции. Часть 2

- итерируемые объекты и итераторы
- генераторы
- функциональное программирование
- namespaces, globals() and locals()
- scopes
- LEGB
- global и nonlocal

#### 7. Классы. Часть 1

- определение класса
- методы и атрибуты
- instance objects, \_\_init\_\_
- доступ к переменным
- приватность

#### 8. Классы. Часть 2

- magic methods
- итерируемые объекты и итераторы
- operator overloading
- inheritance, object
- исключения
- \_\_mro\_\_

#### 9. Модули в python

- модель данных в python
- statements and expressions
- ключевые слова
- try-except-else-finally
- built-in

#### 10. Numpy

- Numpy

## 11. Визуализация данных

- Пакет matplotlib
- Пакет seaborn
- Пакет plotly

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

Для освоения курса необходимо наличие персонального компьютера со стабильным интернет-соединением. Рекомендуемая скорость соединения – от 2 Мбит/с.  
Просмотр видеолекций, прием тестов (проведение текущего контроля успеваемости и промежуточной аттестации), форум предоставляется через Google Chrome, Яндекс.Браузер.

## 6. Перечень рекомендуемой литературы

### Основная литература

1. Python и машинное обучение, Электронная версия печатной публикации / С. Рашка. — Москва, ДМК Пресс, 2017

### Дополнительная литература

1. Python на практике, Электронная версия печатной публикации / М. Саммерфилд. — Москва, ДМК Пресс, 2014

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

<http://dm.fizteh.ru/>

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

Доступ к учебным материалам осуществляется посредством системы дистанционного обучения. Для просмотра видеолекций, прохождения промежуточной аттестации, участие в форуме курса необходим интернет-браузер Google Chrome или Яндекс Браузер.

Дистрибутив языков программирования Anaconda

Интегрированная среда разработки для языка программирования PyCharm

## 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 и понятия, уметь применять полученные знания для решения различных задач.

Успешное освоение курса требует:

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

Самостоятельная работа включает в себя:

- чтение 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 конспекта 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 промежуточный контроль знаний студентов в виде решения задач в соответствии с тематикой занятий.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

<b>по направлению:</b>	Техническая физика
<b>профиль подготовки:</b>	Радиолокационные технологии Физтех-школа авиационных и цифровых технологий кафедра технологий проектирования сложных технических систем
<b>курс:</b>	<u>1</u>
<b>квалификация:</b>	магистр

Семестры, формы промежуточной аттестации:

- 1 (осенний) - Дифференцированный зачет
- 2 (весенний) - Экзамен

**Разработчик:** А.М. Власов

## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-3 Способен организовывать и руководить работой команды, вырабатывая командную стратегию для достижения поставленной задачи	УК-3.1 Организует и координирует работу участников проекта, способствует конструктивному преодолению возникающих разногласий и конфликтов
	УК-3.4 Способен планировать командную работу, распределять поручения членам команды, организовать обсуждение разных идей и мнений
ОПК-5 Способен осваивать и применять современные физико-математические методы и методы искусственного интеллекта для решения профессиональных задач, составлять практические рекомендации по использованию полученных результатов	ОПК-5.1 Владеет современными расчетно-теоретическими методами, методами компьютерного моделирования и средами разработки программного обеспечения, применяемыми при решении задач в своей профессиональной области
	ОПК-5.2 Владеет аналитическими и вычислительными методами решения, способен обосновать эффективность выбранного метода

## 2. Показатели оценивания компетенций

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

### знать:

- парадигму объектно-ориентированного программирования;
- основные объекты и процедуры языка Python;
- методы обработки ошибок в языке Python;
- основные, используемые на практике, библиотеки.

### уметь:

- писать эффективный код, отлаживать и документировать код на языке Python;
- использовать основные библиотеки научных вычислений языка Python для решения типовых прикладных задач в области анализа данных и адаптировать их к своим потребностям в ходе выполнения НИР.

### владеть:

- средствами разработки и тестирования программного кода на языке Python;
- объектами и средствами, предлагаемыми стандартными библиотеками научных вычислений языка;
- базовыми принципами проведения вычислений с использованием библиотек Python.

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

Пример домашнего задания:

1. Реализуйте бесконечный генератор `row_pascal_triangle`, возвращающий строки из треугольника Паскаля. Начало последовательности, возвращаемой генератором, будет выглядеть так: "1" "1 1" "1 2 1" "1 3 3 1"
2. Дано число  $n$ . Создайте массив размером  $n \times n$  и заполните его по следующему правилу:
3. Числа на диагонали, идущей из правого верхнего в левый нижний угол равны 1.
4. Числа, стоящие выше этой диагонали, равны 0.
5. Числа, стоящие ниже этой диагонали, равны 2.
6. Полученный массив выведите на экран.
7. Числа в строке разделяйте одним пробелом."
8. Напишите функцию `fib(n)`, которая по данному целому неотрицательному  $n$  возвращает  $n$ -е число Фибоначчи. В этой задаче нельзя использовать циклы — используйте рекурсию.



9. Дан текст: в первой строке записано количество строк в тексте, а затем сами строки. Выведите все слова, встречающиеся в тексте, по одному на каждую строку. Слова должны быть отсортированы по убыванию их количества появления в тексте, а при одинаковой частоте появления — в лексикографическом порядке.

10. Дан текст: в первой строке задано число строк, далее идут сами строки. Выведите слово, которое в этом тексте встречается чаще всего. Если таких слов несколько, выведите то, которое меньше в лексикографическом порядке.

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

Вопросы к дифференцированному зачету:

1. Типы данных (bool, int, float, complex, str)
2. Типы последовательностей
3. Типы set и mapping: структура, использование, методы, магические методы, subscriptions, slicing.
4. Модель данных Python: объекты, контейнеры, отношения, проверка equality & identity, изменчивость.
5. Арифметические операции, операции сравнения, булевы операции, специальные (магические) методы для них.
6. Оператор if, тернарный оператор, цикл while.
7. For loop, range, comprehensions, generator expressions.
8. Исключения в python. Конструкция try - except для обработки исключений.
9. Функция, вызываемые объекты, упаковка и распаковка, лямбда-выражение.
10. Namespaces, scope, closure.

Вопросы к экзамену:

1. Декораторы: механизм, использование, встроенные декораторы.
2. Элементы функционального программирования (обзор), модули функционального программирования (обзор).
3. Определение класса, создание и удаление, ссылка на атрибут, подписки.
4. Наследование классов, super(), \_\_mro\_\_ .
5. Файлы, контекстный менеджер
6. Типы данных (bool, int, float, complex, str): свойства, методы, магические методы
7. Приведение типов, проверка истинностных значений.
8. Типы последовательностей: структура, использование, методы
9. Магические методы, subscriptions, slicing.
10. Типы set и mapping: структура, использование, методы, магические методы, subscriptions, slicing.

Билет №1.

Типы данных (bool, int, float, complex, str): свойства, методы, магические методы, приведение типов, проверка истинностных значений.

Определение класса, создание и удаление, ссылка на атрибут, подписки.

#### **Критерии оценивания**

отлично (10) - 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, при этом выполнены все контрольные задания семестра.

отлично (9) - выставляется студенту, показавшему свободное оперирование знаниями учебной программы дисциплины, выполнение заданий творческого характера, при этом выполнены все контрольные задания семестра.

отлично (8) - выставляется студенту, показавшему владение программным учебным материалом с наличием несущественных ошибок в действиях, самостоятельно исправляемых учащимся, при этом выполнены более 90% учебных заданий семестра.

хорошо (7) - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускается в ответе или в решении задач некоторые неточности, при этом выполнены более 80% учебных заданий семестра.

хорошо (6) - выставляется студенту если он осознает воспроизведение программного учебного материала, в том числе и различной степени сложности, с несущественными ошибками, затруднения в применении отдельных навыков, при этом выполнены более 80% учебных заданий семестра.

хорошо (5) - выставляется студенту если теоретическое содержание освоено не полностью, некоторые практические навыки сформированы недостаточно, в некоторых случаях были допущены ошибки, при этом выполнены более 70% учебных заданий семестра.

удовлетворительно (4) - выставляется студенту, показавшему 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, при этом выполнены более 60% учебных заданий семестра.

удовлетворительно (3) - выставляется студенту в случае большого количества недочетов и неправильных ответов, а также пассивной работе в ходе занятий, многие учебные задания не выполнены (но не более 50%).

неудовлетворительно, (2) - выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач, при этом большинство учебных заданий семестра не выполнены.

неудовлетворительно (1) - выставляется студенту, который не освоил теоретическое и практическое содержание курса, все выполненные учебные задания содержат грубые ошибки, большинство учебных заданий семестра не выполнены.

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

Во время проведения дифференцированного зачета и экзамена могут пользоваться программой дисциплины, а также справочной литературой, вычислительной техникой, конспектами лекций. Дифференцированный зачет и экзамен может проводиться по итогам текущей успеваемости и сдачи заданий, или путем организации специального опроса, проводимого в устной форме.