

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

**УТВЕРЖДЕНО**  
**Директор физтех-школы**  
**аэрокосмических технологий**  
**С.С. Негодяев**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Программирование на языке Python
<b>по направлению:</b>	Информатика и вычислительная техника
<b>профиль подготовки:</b>	Программная инженерия передовая инженерная школа радиолокации, радионавигации и программной инженерии кафедра технологий проектирования сложных технических систем
<b>курс:</b>	1
<b>квалификация:</b>	бакалавр

Семестры, формы промежуточной аттестации:

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

Аудиторных часов: 90 всего, в том числе:

- лекции: 30 час.
- семинары: 60 час.
- лабораторные занятия: 0 час.

Самостоятельная работа: 45 час.

Всего часов: 135, всего зач. ед.: 3

Количество контрольных работ, заданий: 4

Программу составил: С.В. Александров

Программа обсуждена на заседании кафедры технологий проектирования сложных технических систем  
14.04.2022

## Аннотация

Дисциплина «Программирование на языке Python» включает в себя изучение основ синтаксиса, базовых типов данных и коллекций, стандартных библиотек, массивов, правил структурирования кода. Второй семестр изучения дисциплины посвящен изучению визуализации данных с использованием возможностей языка Python, изучению библиотеки Pandas для обработки и анализа данных, затрагивает численные методы и символьные вычисления.

### 1. Цели и задачи

#### Цель дисциплины

- сформировать навыки программирования на языке Python;
- сформировать представление об используемых в научных приложениях библиотеках;
- дать навык использованию инфраструктуры языка Python для решения практических задач.

#### Задачи дисциплины

- научить программировать на языке Python;
- научить использовать библиотеки для решения практических задач;
- научить решать практические задачи с использованием языка Python и сторонних библиотек.

### 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ПК-1 Способен ставить, формализовывать и решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать новые научные результаты	ПК-1.1 Способен находить, анализировать и обобщать информацию об актуальных результатах исследований в рамках тематической области своей профессиональной деятельности
	ПК-1.2 Способен выдвигать гипотезы, строить математические модели для описания изучаемых явлений и процессов, оценивать качество разработанной модели
ПК-2 Способен самостоятельно или в качестве члена (руководителя) малого коллектива организовывать и проводить научные исследования и их апробацию	ПК-2.1 Знает принципы построения научной работы, методы сбора и анализа полученного материала, способы аргументации

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

В результате освоения дисциплины обучающиеся должны

знать:

- синтаксис языка Python;
- стандартные структуры данных языка Python;
- структуру и основные части стандартной библиотеке Python;
- основные библиотеки, используемые в научных вычислениях.

уметь:

- использовать язык Python в сочетании с библиотеками для решения научных вычислительных задач;
- представлять результат научных расчётов в удобном для анализа виде с использованием соответствующих библиотек Python;
- использовать возможности библиотеки sympy для решения задач в аналитическом виде.

владеть:

- навыком написания программ на языке Python;
- навыком использования библиотек Python для решения научных вычислительных задач.

#### 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, основы синтаксиса, базовые типы данных	4	6		6
2	Коллекции	6	6		6
3	Структурирование кода	4	6		6
4	Стандартная библиотека	6	6		6
5	Массивы и линейная алгебра	10	6		6
6	Визуализация данных		8		4
7	Численные методы		8		4
8	Символьные вычисления		8		4
9	библиотека Pandas		6		3
Итого часов		30	60		45
Подготовка к экзамену		0 час.			
Общая трудоёмкость		135 час., 3 зач.ед.			

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

Семестр: 1 (Осенний)

###### 1. Знакомство, основы синтаксиса, базовые типы данных

Знакомство, основы синтаксиса, базовые типы данных. Интерактивный режим программирования, запуск в программной строке, Скриптовый режим программирования, идентификаторы в Python.

###### 2. Коллекции

Коллекции: классификация, общие подходы и методы, конвертация. Программные объекты, экспериментальная функция.

###### 3. Структурирование кода

Структурирование кода в Python: модульное структурирование, структурирование в комплексы программ. Частичное структурирование.

###### 4. Стандартная библиотека

Стандартная библиотека языка Python: набор средств, встроенных модулей. Доступ к функциям системы: файловый ввод/вывод. Принятие решения.

## 5. Массивы и линейная алгебра

Классификация массивов: одномерные, двумерные. Задачи линейной алгебры: сложение матриц и векторов, умножение на число, скалярное произведение векторов, умножение и транспонирование матриц.

Семестр: 2 (Весенний)

## 6. Визуализация данных

Data Science, анализ больших данных на ранних стадиях эксперимента, Библиотека Matplotlib. Способы визуализации.

## 7. Численные методы

Численные методы и схемы.

## 8. Символьные вычисления

SymPy. Алгебраические преобразования. Упрощение выражений. Вычисления пределов. Дифференцирование. Разложение в ряд. Интегрирование. Решение уравнений. Системы линейных уравнений. Факторизация. Булевы уравнения. Линейная алгебра. Матрицы. Дифференциальные уравнения.

## 9. библиотека Pandas

Библиотека Pandas — программная библиотека на языке Python для обработки и анализа данных. Работа pandas с данными строится поверх библиотеки NumPy, являющейся инструментом более низкого уровня. Предоставляет специальные структуры данных и операции для манипулирования числовыми таблицами и временными рядами. DataFrame, Series. Чтение, запись в файлы. Группировка и агрегирование в pandas. Возможности баз данных. Сводные таблицы в pandas.

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

Необходимое оборудование: аудитория, компьютер (ноутбук) и мультимедийное оборудование (проектор, звуковая система).

## 6. Перечень рекомендуемой литературы

### Основная литература

1. Программирование на Python 3 : Подробное руководство [Текст] = Programming in Python 3 : [учеб. пособие для вузов] / М. Саммерфилд; пер. с англ. А. Киселева. — СПб : Символ-Плюс, 2015. — 608 с.
2. Python для сложных задач: наука о данных и машинное обучение [Текст], [учеб. пособие для вузов] / Дж. Вандер Плас ; [пер. с англ. И. Пальти]. -СПб., Питер, 2018
3. Python и машинное обучение [Текст], крайне необходимое издание по новейшей предсказательной аналитике для более глубокого понимания методологии машинного обучения/С. Рашка, -М., ДМК Пресс, 2017

### Дополнительная литература

1. Introduction to Machine Learning with Python. A Guide for Data Scientists /Andreas C. Muller, Sarah Guido. Beiling ; Boston, O'REILLY, 2017
2. Python machine learning, Machine learning and deep learning with Python, scikit-learn, and TensorFlow/S. Raschka, V. Mirjalili, -Birmingham ; Mumbai, Packt, 2017

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

Не используются

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

Информационные ресурсы: <https://docs.python.org/3/>, <https://www.scipy.org>, <https://matplotlib.org>, <https://anaconda.org>, учебные пособия и сборники задач, разработанные для данного курса. <https://github.com/> для выполнения заданий.

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

Дисциплина "Программирование на языке Python" предполагает собой изучение использования технологии сетевого взаимодействия, разрабатывать многопоточные программы, обеспечивать взаимодействие приложений Python с реляционными базами данных. Разработаете графический интерфейс с помощью библиотеки PyQt5 и подготовите дистрибутив приложения. Примените модульное тестирование с использованием библиотеки PyTest и рассмотрите вопросы безопасности приложения.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

<b>по направлению:</b>	Информатика и вычислительная техника
<b>профиль подготовки:</b>	Программная инженерия передовая инженерная школа радиолокации, радионавигации и программной инженерии кафедра технологий проектирования сложных технических систем
<b>курс:</b>	1
<b>квалификация:</b>	бакалавр
Семестры, формы промежуточной аттестации:	
1 (осенний) - Дифференцированный зачет	
2 (весенний) - Дифференцированный зачет	
<b>Разработчик:</b>	С.В. Александров

## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ПК-1 Способен ставить, формализовывать и решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать новые научные результаты	ПК-1.1 Способен находить, анализировать и обобщать информацию об актуальных результатах исследований в рамках тематической области своей профессиональной деятельности
	ПК-1.2 Способен выдвигать гипотезы, строить математические модели для описания изучаемых явлений и процессов, оценивать качество разработанной модели
ПК-2 Способен самостоятельно или в качестве члена (руководителя) малого коллектива организовывать и проводить научные исследования и их апробацию	ПК-2.1 Знает принципы построения научной работы, методы сбора и анализа полученного материала, способы аргументации

## 2. Показатели оценивания компетенций

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

### знать:

- синтаксис языка Python;
- стандартные структуры данных языка Python;
- структуру и основные части стандартной библиотеке Python;
- основные библиотеки, используемые в научных вычислениях.

### уметь:

- использовать язык Python в сочетании с библиотеками для решения научных вычислительных задач;
- представлять результат научных расчётов в удобном для анализа виде с использованием соответствующих библиотек Python;
- использовать возможности библиотеки sympy для решения задач в аналитическом виде.

### владеть:

- навыком написания программ на языке Python;
- навыком использования библиотек Python для решения научных вычислительных задач.

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

1. Как классифицируется язык программирования Python?
2. Какие коллекции языка Python являются изменяемыми, а какие нет?
3. Каким классическим структурам данных соответствуют типы list, dict, tuple, set?
4. Как устроена операция распаковки коллекций в языке Python 3 и где она применяется?
5. Какие виды аргументов есть у функций в Python?
6. Что такое декоратор и как он работает?
7. Для чего используются и как реализуются менеджеры контекста?
8. Какой способ разбора аргументов командной строки является в Python рекомендованным и как он устроен?
9. Какие коллекции определены в модуле collections?
10. Какие типы данных может хранить numpy.ndarray и чем он отличается от list?

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

1. Чем отличаются функции из `math` от их аналогов из `numpy`?
2. Какие виды умножения `ndarray` на `ndarray` реализованы в `numpy`?
3. Чем отличается синтаксис срезов `numpy` от стандартного?
4. Какие основные типы графиков реализует `matplotlib`?
5. Опишите общий принцип построения анимированных иллюстраций в `matplotlib`.
6. Какие преимущества и недостатки имеет использование использования символьной математики?
7. Как с помощью `sympy` из строки получить функцию?
8. Как с помощью Python выполняется численное интегрирование?
9. Как с помощью Python выполняется численное решение ДУ?

#### Билет 1

1. Основные типы графиков, которые реализует `matplotlib`.
2. Общий принцип построения анимированных иллюстраций в `matplotlib`.

#### Билет 2

1. Виды аргументов функций в Python.
2. Определение декоратора и схемы его работы.

#### 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**

Во время проведения дифференцированного зачета обучающиеся могут пользоваться программой дисциплины, а также справочной литературой, вычислительной техникой, конспектами лекций.

Дифференцированный зачет проводится путем организации специального опроса, проводимого в устной форме.