

федеральное государственное автономное образовательное учреждение
высшего образования

**«Московский физико-технический институт
(национальный исследовательский университет)»
(МФТИ, Физтех)**

Ректор МФТИ

д-р физ.-мат. наук, профессор

Д. В. Ливанов

2023 г.



Дополнительная общеобразовательная общеразвивающая программа
«Программирование на Python»

Москва 2023

1. Общая характеристика программы

1.1 В обществе всё большее значение приобретает умение человека использовать компьютер не на пользовательском уровне, а на уровне начинающего программиста. В обязательном школьном курсе информатики программирование представлено на уровне, достаточном для прохождения экзамена, но не предполагает овладение практическими навыками применения языка. Следствием этого - формальное восприятие обучающимися основ современного программирования и неумение применять полученные знания на практике.

Программа «Программирование на Python» (начальный уровень) направлена на знакомство с базовыми понятиями и элементами языка Python (операторы, числовые и строковые переменные, списки, условия и циклы, функции) и формирование компетенций в области решения задач по программированию. Курс является вводным и лучше всего подойдет слушателям, не имеющим опыта написания программ ни на одном из языков программирования. Содержит множество разнообразных задач по базовым конструкциям языка, что позволяет получить практические навыки программирования на языке Python для решения типовых задач математики и информатики и использовать полученные знания в дальнейшей учебной и профессиональной деятельности.

Цель программы - формирование познавательной активности обучающихся в области функционального программирования, приобретение навыков работы с базовыми структурами языка в интегрированных средах разработки, получение навыков самостоятельного написания кода и разработки эффективных алгоритмов и программ.

1.2. Категории слушателей, на обучение которых рассчитана программа дополнительного образования (далее – программа): учащиеся 8-11 классов.

1.3. Нормативный срок освоения программы – 144 академических часа.

1.4. Форма обучения: очная с применением дистанционных образовательных технологий.

Основной вид занятий – комбинированный, сочетающий в себе элементы теории и практики. Единицей учебного процесса является блок уроков (модуль). Каждый модуль охватывает отдельную информационную технологию или её часть. Внутри модулей разбивка по времени изучения производится педагогом самостоятельно, но с учётом рекомендованного календарно-тематического плана. Темп изучения отдельных разделов блока определяется субъективными и объективными факторами и необходимостью повторения.

2. Планируемые результаты обучения

Личностные результаты

- формирование ответственного отношения к учению, готовности и способности, обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию, осознанному выбору и построению дальнейшей индивидуальной траектории образования на базе ориентировки в мире профессий и профессиональных предпочтений, с учётом устойчивых познавательных интересов;
- формирование целостного мировоззрения, соответствующего современному уровню развития науки и общественной практики, учитывающего социальное, культурное, языковое, духовное многообразие современного мира;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками, детьми старшего и младшего возраста, взрослыми в процессе образовательной, общественно полезной, учебно-исследовательской, творческой и других видов деятельности.
- повышение уровня самооценки благодаря реализованным проектам.

Метапредметные результаты

- умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;
- умение самостоятельно планировать пути достижения целей, в том числе альтернативные, осознанно выбирать наиболее эффективные способы решения учебных и познавательных задач;
- умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности в процессе достижения результата, определять способы действий в рамках предложенных условий и требований, корректировать свои действия в соответствии с изменяющейся ситуацией;
- умение оценивать правильность выполнения учебной задачи, собственные возможности её решения;
- владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;
- умение определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, самостоятельно выбирать основания и критерии для классификации, устанавливать причинно-следственные связи, строить логическое рассуждение, умозаключение (индуктивное, дедуктивное и по аналогии) и делать выводы;
- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебных и познавательных задач;

– умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками; работать индивидуально и в группе: находить общее решение и разрешать конфликты на основе согласования позиций и учёта интересов; формулировать, аргументировать и отстаивать своё мнение;

– формирование и развитие компетентности в области использования информационно коммуникационных технологий.

Предметные результаты

– знание необходимой терминологии («информация», «алгоритм», «исполнитель», «модель»), смысла этих понятий и умение применять полученные знания на практике;

– умение соблюдать сетевой этикет, другие базовые нормы информационной этики и права при работе с компьютерными программами и в сети Интернет;

– умение выполнять созданные программы, осуществлять их разработку, тестирование и отладку, используя изученный язык программирования;

– развитие умений составить и записать алгоритм для конкретного исполнителя; формирование знаний об алгоритмических конструкциях— линейной, условной и циклической, логических значениях и операциях;

– навыки пошагового выполнения алгоритмов, умение осуществлять данные операции как вручную, так и с использованием компьютера;

– умение определять результат выполнения алгоритма при заданных исходных данных;

– знание основных понятий и этапов проектной деятельности.

3. Структура программы

Учебный план

№	Тема (модуль)	Кол-во часов	В том числе		
			Аудит. занятия	Самост. работа	Промежуточная аттестация
1	Модуль 1. Введение в Python	36	18	14	4
1.1	Установка языка Python. Среда разработки	5	3	2	
1.2	Вывод данных	5	3	2	
1.3	Типы данных и переменные	5	3	2	
1.4	Арифметика строк	5	3	2	
1.5	Арифметика чисел	6	3	3	
1.6	Обработка цифр числа	6	3	3	
	Промежуточная аттестация	4			4
2	Модуль 2. Базовые конструкции	36	18	14	4
2.1	Логический тип данных. Логические операции. Условный оператор	5	3	2	
2.2	Составные условия. Каскадный условный оператор	5	3	2	
2.3	Цикл с параметром for	5	3	2	
2.4	Переменная цикла for	5	3	2	
2.5	Варианты цикла for	6	3	3	
2.6	Цикл while. Организация разветвлений в цикле	6	3	3	
	Промежуточная аттестация	4			4
3	Модуль 3. Знакомство с коллекциями	36	18	14	4
3.1	Работа со строками. Индексы, срезы строк	5	3	2	
3.2	Сравнение строк, методы строк	5	3	2	
3.3	Списки. Методы списков	5	3	2	
3.4	Списочные выражения. Кортежи	5	3	2	
3.5	Множества	6	3	3	
3.6	Словари	6	3	3	
	Промежуточная аттестация	4			4
4	Модуль 4. Решение прикладных задач	36	18	10	8
4.1	Функции. Обязательные параметры. Значения функций	5	3	2	
4.2	Функции. Необязательные параметры	5	3	2	
4.3	Функции с переменным числом аргументов	5	3	2	
4.4	Функции как объекты	5	3	2	
4.5	Функции высшего порядка map(), zip(), filter(), reduce()	6	3	3	

4.6	Анонимные функции	6	3	3	
	Промежуточная аттестация	4			4
Итого		144	72	52	20

4. Содержание программы

4.1. Содержание программы по модулям

Модуль 1. Введение в Python

Тема 1.1. Установка языка Python. Среда разработки

знакомство с процессом установки интерпретатора Python;
знакомство со средой разработки и исполнением кода;
организация личного кабинета, поиск и выкладывание материалов;
знакомство с системой автоматизированной проверки задач.

Тема 1.2. Вывод данных

понятие о языке высокого уровня Python;
структура программы,
функция print()
именованными аргументами sep и end;
стандарт PEP8

Тема 1.3. Типы данных и переменные

понятие константы;
понятие переменной и требования к ней;
оператор присваивания, множественное присваивание;
функция input(), ввод данных с клавиатуры;
устройство памяти и ссылок в Python

Тема 1.4. Арифметика строк

представление строк в памяти компьютера;
арифметические операции над строками.

Тема 1.5. Арифметика чисел

представление чисел в памяти компьютера;
арифметические операции над числами;
«утиная типизация» в Python

Тема 1.6. Обработка цифр числа

Алгоритм получения цифр n-значного числа

Модуль 2. Базовые конструкции

Тема 2.1. Логический тип данных. Логические операции. Условный оператор

логический тип данных, запись логических условий на языке Python;
условный оператор;
неполный условный оператор

Тема 2.2. Составные условия. Каскадный условный оператор

составление сложных логических условий с использованием операторов and, or, not;

особенности выполнения сложных логических условий в Python;

многообразие способов записи ветвления — каскадное ветвление

Тема 2.3. Цикл с параметром for

синтаксис оператора for, особенности записи цикла;

чтение последовательности;

итерирование по объектам;

фильтрация;

функция next от iterable-объекта;

функция enumerate

Тема 2.4. Переменная цикла for

функция range();

переменная цикла;

использование переменной цикла;

изменение переменной;

изменение переменной в цикле;

выражения с числами от 1 до n

Тема 2.5. Варианты цикла for

разные виды range;

шаг в range;

отрицательный шаг;

использование шага цикла;

цикл по строке

Тема 2.6. Цикл while. Организация разветвлений в цикле

синтаксис оператора while;

операторы досрочного выхода из цикла continue, break, pass;

условие после выхода из цикла;

бесконечный цикл;

причины бесконечного цикла;

обработка последовательностей

Модуль 3. Знакомство с коллекциями

Тема 3.1. Работа со строками. Индексы, срезы строк

знакомство с типом данных str;

индексация строки, особенности индексов, отрицательные индексы;

срезы строк, срезы с отрицательными индексами;

сокращенная запись среза;

первые и последние n символов;

срез с шагом;

разные типы срезов

Тема 3.2. Сравнение строк, методы строк

операции со строками;

сравнение букв;

сравнение слов;

перестановка строк;

латинские буквы;

сравнение произвольных символов;

стандарт ASCII, таблица ASCII;
сравнение произвольных строк;
прописные буквы;
сравнение строк с числами;
методы строк: find(), rfind(), count(), replace()
форматированные строки;
f-строки

Тема 3.3. Списки. Методы списков

структура данных – списки, особенности данной структуры;
понятие элемента списка, индекса элемента, значения элемента, срезы;
заполнение списка;
вывод списка;
обработка списка;
последовательный поиск, сортировка: метод sort(), функция sorted();
методы списков: append(), index(), pop(), split(), метод строки join()

Тема 3.4. Списочные выражения. Кортежи

способ создания списков: list comprehension, примеры использования;
считыванием входных данных;
условия в списочном выражении;
вложенные циклы;
структура данных – кортеж, примеры кортежей;
кортеж с одним элементом;
особенности кортежей, зачем использовать кортежи вместо списков;
индексация кортежей, срезы;
операция конкатенация и умножение на число;
встроенные функции sum(), min(), max();
методы index(), count()
перебор кортежей, распаковка;
сортировка кортежей, сравнение;
преобразование кортежей в список и строку и наоборот

Тема 3.5. Множества

множества в математике: конечные и бесконечные множества, равенство множеств, подмножество и надмножество, пустое множество;
структура данных – множество в python;
создание множества, пустое множество;
вывод множества;
дубликаты при создании множеств;
приемы работы с множествами: функция len(), оператор принадлежности in, встроенные функции sum(), min(), max();
методы множеств;
операции над множествами

Тема 3.6. Словари

структура данных – словарь;
отличия словарей от списков;

создание словарей, обращение по ключу, создание словарей на основе списков и кортежей;
пустой словарь;
вывод словаря;
особенности словарей;
встроенные функции len(), sum(), min(), max();
оператор принадлежности in;
перебор словарей;
распаковка словаря;
форматированный вывод словарей;
сравнение словарей;
методы keys(), values() и items()
добавление и изменение элементов в словаре, удаление элементов из словаря;
методы get(), update(), pop(), popitem(), clear(), copy()

Модуль 4. Решение прикладных задач

Тема 4.1. Функции. Обязательные параметры. Значения функций

синтаксис функции, назначение функции
функции без параметров;
функции с параметрами;
локальные и глобальные переменные

Тема 4.2. Функции. Необязательные параметры

позиционные, необязательные аргументы;
именованные аргументы, целесообразность применения именованные аргументы;

комбинирование позиционных и именованных аргументов;
изменяемые типы в качестве значений по умолчанию

Тема 4.3. Функции с переменным числом аргументов

переменное количество аргументов;
передача аргументов в форме списка и кортежа;
получение и передача именованных аргументов в виде словаря;
keyword-only аргументы

Тема 4.4. Функции как объекты

функции как объекты;
функции в качестве аргументов других функций;
встроенные функции, принимающие функции в качестве аргументов;
функции в качестве возвращаемых значений других функций

Тема 4.5. Функции высшего порядка map(), zip(), filter(), reduce()

функции высшего порядка;
самописные функции map(), zip(), filter(), reduce()
модуль operator

Тема 4.6. Анонимные функции

определение анонимных функций;
условный оператор в теле анонимной функции;

передача аргументов в анонимную функцию;
ограничения анонимных функций

4.2. Список рекомендуемой литературы

Поляков К. Ю., Еремин Е. А. Информатика. Углублённый уровень. Учебник для 10 класса в 2 частях. М.: БИНОМ. Лаборатория знаний, 2014.

Информатика и ИКТ. Задачник-практикум в 2 частях. Под ред. И. Г. Семакина и Е. К. Хеннера. М.: БИНОМ. Лаборатория знаний, 2014.

Задачи по программированию. Под ред. С. М. Окулова, М.: БИНОМ. Лаборатория знаний, 2006.

Окулов С. М. Основы программирования. М.: Бином. Лаборатория знаний, 2012.

Лутц М. Изучаем Python. СПб.: Символ-Плюс, 2011.

Хахаев, И.А. Практикум по алгоритмизации и программированию на Python: курс / И.А. Хахаев. -2-е изд., исправ. - Москва: Национальный Открытый Университет «ИНТУИТ», 2016. - 179 с.: ил. - Библиогр. в кн.

5. Материально-технические условия реализации программы

Требования к помещению:

- помещение для занятий, отвечающие требованиям СанПин для учреждений дополнительного образования;
- качественное освещение;
- столы, стулья по количеству обучающихся и 1 рабочим местом для педагога.

Информационное обеспечение:

- операционная система (желательно Windows);
- браузеры: Yandex Browser, Chrome, Chrome Mobile, Firefox, Opera, Safari, Mobile Safari, Edge, Python;
- среда Wing IDE 101 (версии 6 или выше);
- среда PyCharm Community Edition.

Требования к оборудованию

Таблица 4

Наименование специализированных аудиторий, кабинетов, лабораторий	Вид занятий	Наименование оборудования, программного обеспечения
Аудитория с доступом в Интернет	Аудиторные занятия	Компьютер, Visual Studio, проектор, видеокамера, доступ в Интернет
Аудитория с доступом в Интернет	Самостоятельная работа	Персональные компьютеры/ноутбуки, Visual Studio, доступ в Интернет

6. Оценка качества освоения программ

Оценка качества освоения программы осуществляется в процессе промежуточной аттестации.

Формы и методы промежуточного контроля представлены в таблице 5.

Таблица 5

Наименование модулей	Основные показатели оценки	Формы и методы контроля и оценки
Модуль 1. Введение в Python	Установленное количество выполненных заданий	Самостоятельное выполнение заданий
Модуль 2. Базовые конструкции	Установленное количество выполненных заданий	Самостоятельное выполнение заданий
Модуль 3. Знакомство с коллекциями	Установленное количество выполненных заданий	Самостоятельное выполнение заданий
Модуль 4. Решение прикладных задач	Установленное количество выполненных заданий	Самостоятельное выполнение заданий

Примерные задания для проведения промежуточной аттестации по модулям

Модуль 1. Введение в Python

1. Найти сумму нечетных делителей натурального числа.
2. Найти все натуральные числа a , b и c из интервала от 1 до 20, для которых выполняется равенство: $a + b = c^2$
3. Найти количество трехзначных чисел, сумма цифр которых равна A , а само число заканчивается цифрой B (A и B вводятся с клавиатуры)
4. Даны координаты трех точек A , B , C . Найти сумму всех отрезков (AB , BC , AC).
5. Найдите среднее геометрическое двух чисел (квадратный корень произведения)
6. Напишите программу, которая на вход будет получать два значения: строку и ключ, а на выходе возвращать строку, зашифрованную путем применения функции XOR (^) над символами строки с ключом.
7. Дано натуральное число n . Вывести на экран n первых простых чисел.
8. Дано натуральное число n . Проверить, представляет ли собой палиндром его десятичная запись.
9. Дано натуральное число. Проверить, является ли оно совершенным.

Примечание: совершенным числом называется натуральное число, равное сумме всех своих собственных делителей (то есть натуральных делителей, отличных от самого числа). Например, 6 – совершенное число, оно имеет три собственных делителя: 1, 2, 3, и их сумма равна $1+2+3 = 6$.

10. Дано натуральное число n (которое также может быть равно нулю). Вычислить $n!$

Модуль 2. Базовые конструкции

1. Введите с клавиатуры строку, разбейте её на слова. Выведите список из слов, записанный в обратном порядке.
2. Введите с клавиатуры две строки. Разбейте первую строку на слова и добавьте в конец получившегося списка вторую строку целиком. Выведите получившийся список.
3. Дан список, заполненный случайным образом нулями и единицами. Найти самую длинную непрерывную последовательность единиц и определить индексы первого и последнего элементов в ней.
4. Из списка чисел удалить элементы, значения которых больше 50 и меньше 100. При этом удаляемые числа сохранить в другом списке.
5. Написать функцию, которая принимает на вход любое количество чисел и возвращает их среднее арифметическое. Числа передаются через запятую.
6. Написать рекурсивную функцию, переводящую числа из одной системы счисления в другую.
7. Написать функцию, проверяющую, есть ли среди значений словаря хоть одно число больше 4. Функция должна возвращать *True*, если такое значение есть, и *False* в противном случае. Гарантируется, что в словаре есть хотя бы одна запись и значениями словаря являются только числа.
8. Написать функцию *count_len(words)*, принимающую на вход список строк, а возвращающую словарь, ключами которого являются строки из списка *words*, а значениями — их длины (в символах). Например, *count_len(['hello', 'world!'])* возвращает словарь *{'hello':5, 'world!':6}*.
9. Написать функцию, принимающую на вход список *students*, каждый элемент которого является кортежем: первый элемент кортежа является именем студента, а второй его оценкой. Например: *students = [('Bob', 3), ('Alice', 4)]*. Функция должна вернуть оценку студента, имя которого является первым при алфавитной сортировке. Все студенты имеют разные имена.
Написать функцию, которая на вход принимает список строк с именами и фамилиями, сортирующую его по фамилии, а при совпадении фамилий — по именам, и возвращающую в отсортированном виде. Например, для списка *['Alice Smith', 'John Doe', 'Jack Doe']* должно быть возвращено *['Jack Doe', 'John Doe', 'Alice Smith']*.

Модуль 3. Знакомство с коллекциями

1. Написать telegram эхо-бота, способного получать от пользователя текстовое сообщение и возвращать его.
2. Научить бота выдавать статьи из Wikipedia
3. Написать бота с двумя виртуальными кнопками
4. Написать бота, который будет вести диалог с пользователем.

Модуль 4. Решение прикладных задач

1. Разработка игры «Шахматы» для двух игроков с консольным интерфейсом.

2. Разработка игры «Змейка» с графическим интерфейсом (tkinter)
3. Разработка игры «Футбол» с графическим интерфейсом
4. Создание приложения для автоматической генерации писем и документов по шаблону MS Word
5. Создание консольного графического редактора (применение фильтров, изменение размеров, повороты изображения и пр.)

7. Составители программы

Малеев Алексей Викторович, директор Центра развития ИТ-образования МФТИ

Тимохин Владимир Николаевич, д.э.н., профессор, специалист по УМР Центра развития ИТ-образования

Созыкин Андрей Владимирович, к. т. н., заместитель директора Центра развития ИТ-образования МФТИ

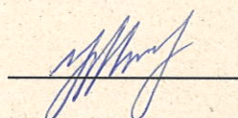
Мартемьянов Роман Юрьевич, заместитель директора Центра развития ИТ-образования МФТИ

Даревский Сергей Григорьевич, руководитель группы Центра развития ИТ-образования МФТИ

Сырцова Елена Леонидовна, PhD, к. пед. н., доцент, руководитель проектов Центра развития ИТ-образования МФТИ

Токмакова Ольга Викторовна, PhD, к. пед. н., доцент, специалист по учебно-методической работе Центра развития ИТ-образования МФТИ

Согласовано
Ведущий специалист отдела
сопровождения образовательных
программ

 Ж.И. Зубцова

Согласовано
Директор ЦРИТО

 А. В. Малеев

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

ВЫПИСКА ИЗ ПРОТОКОЛА № 8

заседания учебно-методического совета от 29 марта 2023 года.

ПОВЕСТКА:

Рассмотрение дополнительных общеобразовательных и профессиональных программ.

Проректор по учебной работе А. А. Воронов

СЛУШАЛИ: заместителя директора Центра дополнительного, дополнительного профессионального и онлайн-образования «ПУСК» Рыбакову А.И. о представлении дополнительных общеобразовательных и профессиональных программ. (Центр «ПУСК» МФТИ).

ПОСТАНОВИЛИ:

Рекомендовать к утверждению в установленном порядке дополнительную общеобразовательную программу «Программирование на Python».

Решение принято единогласно.

Форма проведения заседания: заочная.

Председатель УМС МФТИ

Ученый секретарь УМС МФТИ



А.А. Воронов

М.В. Березникова