**Практическая работа №1. Программирование линейных алгоритмов**

Объём учебного времени – 2ч

Методические рекомендации

1. Цель работы: научиться создавать простейшие программы на языке python, использовать различные типы данных.

2.Перечень необходимых средств обучения:

* 1. Программное обеспечение:
* Программа ОС Windows 10;
* Python (версии не ниже 2.4);
* Интегрированная среда IDLE Python.

3. Основные теоретические положения:

**Арифметическая операция** — вычислительная операция над числами.

**Переменная** — это простейшая именованная структура данных, в которой может быть сохранён промежуточный или конечный результат работы программы.

Для удобства разработки кода программисты используют среды разработки (IDE).

**Python** — язык с динамической типизацией: каждая переменная в каждый момент времени имеет определенный тип, но этот тип может меняться по ходу выполнения программы, достаточно просто присвоить ей новое значение другого типа.

На самом деле переменная в python является лишь ссылкой на объект в памяти. При создании любой переменной (число, строка или массив) в неё записывается ссылка на объект, а сам объект находится где-то в оперативной памяти далеко от самой переменной со ссылкой. Таким образом, несколько переменных могут указывать на один объект, и при изменении объекта (например, списка) изменится результат обращения к нему с использованием каждой переменной.

Все данных на языке Python являются объектами, даже типы данных. Все типы данных делятся на изменяемые и неизменяемые. В Python 3 объекты могут иметь следующие **типы данных**:

* **bool** логический (принимает значения True или False);
* **NoneType** объект со значением None (отсутствие значения)
* **int** целое число (размер ограничен оперативной памятью);
* **float** вещественное число с плавающей точкой;
* **complex** – комплексные числа (вещественная часть + мнимая частьJ)
* **str** строка (Unicode);
* **bytes** – неизменяемая последовательность байтов;
* **bytearray** – изменяемая последовательность байтов;
* **list** – списки;
* **tuple** – кортежи;
* **range** – диапазоны;
* **dict** – словари (аналогичен ассоциативным массивам);
* **set** – множества (коллекции уникальных объектов);
* **frozenset** – неизменяемые множества;
* **ellipses** обозначается в виде трёх точек или ellipses;
* **function** – функции;
* **module** – модули;
* **type** – классы и типы данных.

**Ввод данных**

Для считывания строки со стандартного ввода используется функция input(), которая считывает строку с клавиатуры и возвращает значение считанной строки, которое сразу же можно присвоить переменным

Для ввода данных предназначена функция **input()**



a = input()

b = input()

Правда, функция input возвращает текстовую строку. Если нужно сделать так, чтобы переменные имели целочисленные значения, то сразу же после считывания выполним преобразование типов при помощи функции int, и запишем новые значения в переменные a и b:

a = int(a)

b = int(b)

Можно объединить считывание строк и преобразование типов, если вызывать функцию int для того значения, которое вернет функция input():

a = int(input())

b = int(input())

**Вывод данных**

Для вывода данных используется функция print может выводить не только значения переменных, но и значения любых выражений. Например, допустима запись print(2 + 2 \*\* 2). Также при помощи функции print можно выводить значение не одного, а нескольких выражений, для этого нужно перечислить их через запятую:

a = 1

b = 2

print(a, '+', b, '=', a + b)

В данном случае будет напечатан текст 1 + 2 = 3: сначала выводится значение переменной a, затем строка из знака “+”, затем значение переменной b, затем строка из знака “=”, наконец, значение суммы a + b.

Обратите внимание, выводимые значение разделяются одним пробелом. Но такое поведение можно изменить: можно разделять выводимые значения двумя пробелами, любым другим символом, любой другой строкой, выводить их в отдельных строках или не разделять никак. Для этого нужно функции print передать специальный именованный параметр, называемый **sep**, равный строке, используемый в качестве разделителя (sep — аббревиатура от слова separator, т.е. разделитель). По умолчанию параметр **sep** равен строке из одного пробела и между значениями выводится пробел. Чтобы использовать в качестве разделителя, например, символ двоеточия нужно передать параметр sep, равный строке ‘:’:

print(a, b, c, sep = ':')

Аналогично, для того, чтобы совсем убрать разделитель при выводе нужно передать параметр sep, равный пустой строке:

print(a, '+', b, '=', a + b, sep = '')

Для того, чтобы значения выводились с новой строке, нужно в качестве параметра sep передать строку, состоящую из специального символа новой строки, которая задается так:

print(a, b, sep = '\n')

Символ обратного слэша в текстовых строках является указанием на обозначение специального символа, в зависимости от того, какой символ записан после него. Наиболее часто употребляется символ новой строки ‘\n’. А для того, чтобы вставить в строку сам символ обратного слэша, нужно повторить его два раза: ‘\\’.

Вторым полезным именованным параметром функции print является параметр **end,** который указывает на то, что выводится после вывода всех значений, перечисленных в функции print. По умолчанию параметр end равен ‘\n’, то есть следующий вывод будет происходить с новой строки. Этот параметр также можно исправить, например, для того, чтобы убрать все дополнительные выводимые символы можно вызывать функцию print так:

print(a, b, c, sep = '', end = '')

 Иногда бывает полезно целое число записать как строку. И, наоборот, если строка состоит из цифр, то полезно эту строку представить в виде числа, чтобы дальше можно было выполнять арифметические операции с ней. Для этого используются функции, название которых совпадает с именем типа, то есть int, float, str. Например, int(‘123’) вернет целое число 123, str(123) вернет строку ‘123’, а следующая инструкция:

print(str(2 + 2) \* int('2' + '2'))

выведет символ “4”, повторенный 22 раза.

Функция int также поможет превратить дробное число в целое, отбросив дробную часть: int(12.3) = 12, int(-12.3) = 12.

Для хранения целых чисел в Python используется тип int (от англ. integer — целое число). Этот тип позволяет хранить как положительные, так и отрицательные числа, при этом сколь угодно большие (в отличие от большинства языков программирования, где целочисленная переменная обычно занимает лишь несколько байт). Более точно, размер числа ограничен лишь имеющейся памятью.

Для преобразования к целому типу используется функция одноименная с именем типа — int:

s = int(input())

t = int(2.5)

Для целых чисел определены арифметические операции:

+ сложение;

— вычитание;

\* умножение (пропускать его в выражениях вида 2 \* x нельзя);

/ деление;

// целочисленное деление (дробная часть отбрасывается);

% остаток от деления;

\*\* возведение в степень.

В арифметических выражениях порядок действий соответствует принятому в математике, для изменения порядка действий используются круглые скобки:

(x - a) \* 2 - (a - (1 - c))

Функция abs(x) возвращает модуль (абсолютное значение) числа x.

Присваивания вида x = x + a, x = x // a и т. п. можно писать сокращенно: x += a, x //= a.

Пример считывания одного числа

s = int(input())

Пример считывания двух чисел, вводимых в разных строчках

s = int(input())

t = int(input())

**Модуль math**

import math

from math import \*

# возведение числа 2 в степень 3

n1 = math.pow(2, 3)

print(n1)  # 8

# ту же самую операцию можно выполнить так

n2 = 2\*\*3

print(n2)

# возведение в квадрат

print(math.sqrt(9))  # 3

# ближайшее наибольшее целое число

print(math.ceil(4.56))  # 5

# ближайшее наименьшее целое число

print(math.floor(4.56))  # 4

# перевод из радиан в градусы

print(math.degrees(3.14159))  # 180

# перевод из градусов в радианы

print(math.radians(180))   # 3.1415.....

# косинус

print(math.cos(math.radians(60)))  # 0.5

# cинус

print(math.sin(math.radians(90)))   # 1.0

# тангенс

print(math.tan(math.radians(0)))    # 0.0

print(math.log(8,2))    # 3.0

print(math.log10(100))    # 2.0

import math

radius = 30

# площадь круга с радиусом 30

area = math.pi \* math.pow(radius, 2)

print(area)

# натуральный логарифм числа 10

number = math.log(10, math.e)

print(number)

**Генератор случайных чисел**

from random import \*

Целые числа на отрезке [a, b]

X=randint(10, 60)

Генератор на [0, 1)

X=random()

Генератор на [a, b] (вещественные числа)

X=uniform(1.2, 3.5)

Задачи

1. Напишите программу, которая находит сумму, произведение и среднее арифметическое трёх целых чисел, введённых с клавиатуры. Например, при вводе чисел 4, 5 и 7 мы должны получить ответ 4+5+7=16, 4\*5\*7=140, (4+5+7)/3=5.333333

2. Напишите программу, которая вводит радиус круга и вычисляет его площадь и длину окружности. Используйте встроенную константу math.pi из модуля math, приближённо равную числу.

3.Напишите программу, которая вводит трёхзначное число и разбивает его на цифры. Например, при вводе числа 123 программа должна вывести «1, 2, 3».

4. Напишите программу, которая вводит координаты двух точек на числовой оси и выводит расстояние между ними.

5. Напишите программу, которая вводит два целых числа, а и b (a < b), и выводит на экран 5 случайных целых чисел на отрезке [a,b].

6. Напишите программу, которая моделирует бросание двух игральных кубиков: при запуске выводит случайное число в диапазоне от 2 до 12.

7. Напишите программу, которая случайным образом выбирает дежурных: выводит два различных случайных числа в диапазоне от 1 до N, где N – количество учеников вашего класса.

*8. n* школьников делят между собой *k.* яблок поровну, неделящийся остаток остаётся в корзинке. Сколько яблок достанется каждому школьнику? Сколько яблок останется в корзинке? Программа получает на вход в первой строке натуральное число *n.*, а во второй — целое неотрицательное число *k*, и должна вывести два целых числа: количество яблок у каждого школьника и количество яблок, оставшихся в корзинке.

9. Феофилакт хочет купить ракетки и шарики для игры в настольный теннис. Один комплект ракеток стоит *A* рублей, один шарик стоит *B* рублей. У Феофилакта есть *C* рублей, *C*≥*A*, он покупает один комплект ракеток и шарики на оставшиеся деньги. Сколько шариков купит Феофилакт? Программа получает на вход три целых числа: *A* , *B*, *C*. Программа должна вывести ответ на задачу.

10. Дано натуральное число. Найдите число десятков в его десятичной записи (то есть предпоследнюю цифру его десятичной записи). Если заданное число является однозначным, то необходимо вывести 0.