

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Программное обеспечение информационных технологий»

ИНФОРМАТИКА

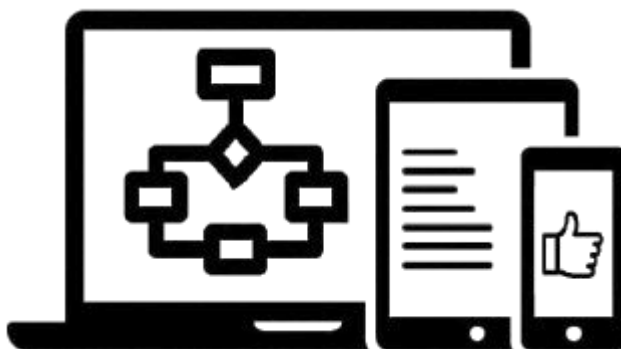
*Методические рекомендации к лабораторным работам
для студентов специальностей*

*6-05-0713-04 «Автоматизация технологических
процессов и производств»,*

*6-05-0714-03 «Инженерно-техническое проектирование
и производство материалов и изделий из них»*

*и 6-05-0722-05 «Производство изделий на основе
трехмерных технологий» дневной формы обучения*

Часть 2



Могилев 2024

УДК 004
ББК 32.973
И74

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Программное обеспечение информационных технологий» «30» января 2024 г., протокол № 6

Составитель ст. преподаватель О. А. Пономарева

Рецензент канд. техн. наук, доц. В. М. Ковальчук

Методические рекомендации к лабораторным работам для студентов специальностей 6-05-0713-04 «Автоматизация технологических процессов и производств», 6-05-0714-03 «Инженерно-техническое проектирование и производство материалов и изделий из них» и 6-05-0722-05 «Производство изделий на основе трехмерных технологий» дневной формы обучения.

Учебное издание

ИНФОРМАТИКА

Часть 2

Ответственный за выпуск	В. В. Кутузов
Корректор	А. А. Подошевка
Компьютерная верстка	М. М. Дударева

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 16 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский
университет, 2024

Содержание

7 Лабораторная работа № 7. Программирование.	4
8 Лабораторная работа № 8. Программирование.	5
9 Лабораторная работа № 9. Программирование.	8
10 Лабораторная работа № 10. Программирование.	10
11 Лабораторная работа № 11. Программирование.	11
12 Лабораторная работа № 12. Программирование.	14
13 Лабораторная работа № 13. Программирование.	15
14 Лабораторная работа № 14. Программирование.	19
Список литературы	21

Часть 2

7 Лабораторная работа № 7. Программирование

Цель работы: научиться устанавливать и настраивать Python для возможности выполнения всех лабораторных работ.

Python – высокоуровневый язык программирования общего назначения с динамической типизацией, автоматическим управлением памятью, поддержкой многопоточных вычислений и удобными структурами данных. На рисунке 7.1 перечислены области применения языка программирования Python.

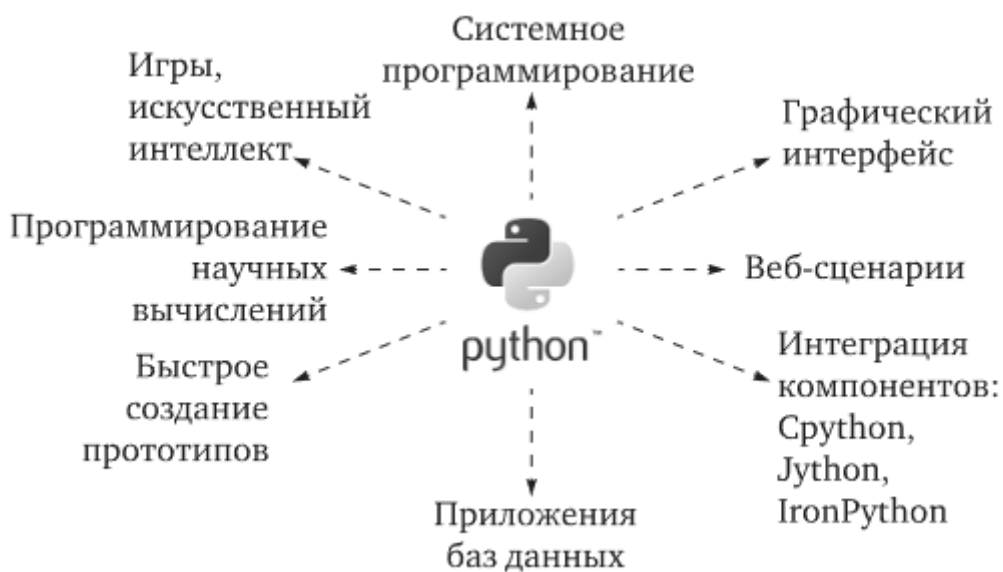


Рисунок 7.1 – Области применения языка программирования Python

Python является интерпретируемым языком программирования (команды выполняются шаг за шагом), в отличие от компилируемых, где текст программы переводится в эквивалентный машинный код. Поэтому для запуска программ, написанных на языке программирования Python, необходима программа-интерпретатор (виртуальная машина) Python. Данная программа скрывает от Python-программиста все особенности операционной системы, поэтому, написав программу на Python в системе Windows, ее можно запустить, например, в GNU/Linux и получить схожий результат.

Установка и настройка интерпретатора Python зависят от используемой операционной системы и решаемых задач. Для работы понадобится интерпретатор версии 3.10 или выше.

Скачать и установить его можно совершенно бесплатно с официального сайта <https://www.anaconda.com/download>. Выбор дистрибутива Anaconda обусловлен простотой использования и большим набором сторонних модулей, которые упрощают работу с данными [1].

Задания

- 1 Зайти на сайт <https://www.python.org/>, скачать и установить на компьютер Python.
- 2 Зайти на сайт <https://www.jetbrains.com/pycharm/download/>, скачать и установить на компьютер PyCharm Community.
- 3 Зайти на сайт <https://www.anaconda.com/>, скачать и установить на компьютер Anaconda.
- 4 Запустить PyCharm Community и настроить его для работы с Python.
- 5 Зайти на сайт <https://colab.research.google.com/>, зарегистрироваться на нем и ознакомиться с работой системы GoogleColaboratory для возможности программировать на Pythononline.

Контрольные вопросы

- 1 Назовите основные преимущества и недостатки языка программирования Python.
- 2 Укажите преимущества виртуальной машины Python.
- 3 Почему Python называют интерпретируемым языком программирования?
- 4 Опишите процесс установки Python.
- 5 Для чего предназначена система Google Colaboratory?

8 Лабораторная работа № 8. Программирование

Цель работы: изучить функциональные возможности Python; освоить работу с простейшими математическими операциями.

В самом начале обучения Python можно рассматривать как интерактивный интеллектуальный калькулятор. В математических выражениях в качестве операндов могут использоваться как целые числа (1, 4, -5), так и вещественные (в программировании их еще называют числами с плавающей точкой): 4.111, -9.3. Математические операции, доступные над числами в Python3, представлены в таблице 8.1.

Как и в любом другом языке программирования, в Python существует приоритет применения (выполнения) приведенных выше операторов. Ниже приведен список приоритетов исполнения операторов (в порядке убывания):

- возведение в степень (**);
- унарный плюс, минус и побитовое отрицание (~, +, -);
- умножение, деление, целочисленное и деление по модулю (*, /, %, //);
- сложение и вычитание (+, -);
- побитовый сдвиг вправо и влево (>>, <<);
- побитовое логическое «И» (&);
- побитовое логическое «ИЛИ» и «исключающее ИЛИ» (|, ^);
- операторы сравнения (<=, <, >, >=);

- операторы равенства (`==`, `!=`);
- операторы присваивания (`=`, `%=`, `/=`, `//=`, `-=`, `+=`, `*=`, `**=`);
- тождественные операторы (`is`, `is not`);
- операторы членства (`in`, `not in`);
- логические операторы (`not`, `or`, `and`).

Таблица 8.1 – Математические операции в Python3

Оператор	Описание
<code>+</code>	Сложение
<code>-</code>	Вычитание
<code>*</code>	Умножение
<code>/</code>	Деление (в результате вещественное число)
<code>//</code>	Деление с округлением вниз (целочисленное деление)
<code>**</code>	Возведение в степень
<code>%</code>	Остаток от деления

Рассмотрим выражение $y = x + 3 * 6$, где y и x являются переменными, которые могут содержать некоторые значения. На языке Python вычислить значение y при x , равном 1, можно следующим образом:

```
>>> x = 1
>>> y = x + 3 * 6
>>> y
19
```

В выражении нельзя использовать переменную, если ранее ей не было присвоено значение с помощью инструкции присваивания. Для Python такие переменные не определены, и их использование приведет к ошибке.

Содержимое переменной y можно вывести на экран, если в интерактивном режиме ввести ее имя.

Имена переменным задает программист, но есть несколько ограничений, связанных с их наименованием. Имена переменных нельзя начинать с цифры и в качестве имен переменных нельзя использовать ключевые слова, которые для Python имеют определенный смысл.

Задание

Создайте программу для вычисления уравнения из таблицы 8.2.

Таблица 8.2 – Задания

Номер варианта	Функция
1	$y = b^2 + \frac{ab - \sqrt{b^2 - 4ac}}{2a + b} + \sin^2 x^3$
2	$y = c^3 - \frac{b + \sqrt{a^2 - b^3c}}{2,5b - 3,4a} + \cos^4 x^2$
3	$y = ab + \frac{-b - \sqrt{a^4 - c^3b}}{2,7b - a/3,24} + \operatorname{tg}^4 x^2$
4	$y = a/b + \frac{a + \sqrt{a^3 - c^2/b}}{2,4b - 2,3/c} + \operatorname{tg}^2 x^3$
5	$y = \frac{a}{b} + \frac{\sqrt[3]{b} - \sqrt{a^4 - c^3b}}{2,8b - 3,6/a} + \operatorname{tg}^3 x^2$
6	$y = cb + \frac{\frac{b}{3,7} + \sqrt{2a^2 - c^4b}}{2,5/b - 3,9a} + \operatorname{tg}^3 x^5$
7	$y = b^{2,5} + \frac{-4,1b - \sqrt{a^4 - c^6b}}{2,2/b - 3,7a} + \operatorname{tg}^4 x^2$
8	$y = a\sqrt[3]{a/b} + \frac{-b + \sqrt{a^5 - c^3b}}{5,2b - 4,3a} + \operatorname{tg}^2 x^4$
9	$y = \frac{-b - \sqrt{a^2 - c^3b}}{1,2b - 2,5a} + \operatorname{tg}^2 x^4 - a\sqrt[3]{a/c}$
10	$y = b/c + \frac{-b + \sqrt{a^4 - c^3b}}{5,2b - 1,5a} + \operatorname{tg}^3 x^6$
11	$y = a\sqrt[5]{a/4b} - \frac{b - \sqrt{a^4 - c^3b}}{7,8b - 1,5a} + \operatorname{tg}^2 x^3$
12	$y = c/a + \frac{-b + \sqrt{a^4 - c^3b}}{3,6b - 1,5a} + \operatorname{tg}^5 x^6$
13	$y = a\sqrt[3]{a/7b} - \frac{b + \sqrt{a^4 - c^3b}}{6,9b - 1,5a} + \operatorname{tg}^3 x^6$
14	$y = a/b + \frac{-b - \sqrt{a^4 - c^3b}}{4,1b - 1,5a} + \operatorname{tg}^6 x^6$
15	$y = a\sqrt[3]{a/2b} - \frac{b + \sqrt{a^4 - c^3b}}{9,3b - 1,5a} + \operatorname{tg}^3 x^6$

Контрольные вопросы

- 1 Какие типы данных используются в Python?
- 2 Опишите процесс написания простых приложений.
- 3 Как в Python вводятся и выводятся данные?

9 Лабораторная работа № 9. Программирование

Цель работы: изучить работу условного оператора.

Оператор `if` в Python выбирает действия, которые будут выполняться в процессе работы программы в зависимости от условий. Он является основным инструментом выбора в Python и представляет большой объем логики.

Самая общая форма конструкции `if-then-else` в Python выглядит так:

```
if условие 1:
    блок 1
elif условие 2:
    блок 2
...
elif условие (n-1):
    блок (n-1)
else:
    блок (n)
```

Если результат проверки условия 1 равен `True`, то выполняется блок 1. В противном случае, если результат проверки условия 2 равен `True`, выполняется блок 2, и так далее, пока не будет найдено условие, которое возвращает `True`, или не будет достигнута секция `else` (тогда выполняется блок(n)). Если `else` отсутствует и все проверки условий вернули `False`, то ни один из блоков ветвления не выполнится [2].

Если блок кода состоит из большого числа цепочек `if-elif-else`, стоит призадуматься об использовании другой условной конструкции управления потоком выполнения программы: `match-case`. Ее синтаксис достаточно прост в использовании и позволяет сократить объем кода посредством замены блоков `if`, осуществляющих сравнение переменных встроенных типов данных, последовательностей или экземпляров классов с помощью оператора «`==`».

В общем виде конструкцию `match-case` можно записать следующим образом:

```
match выражение:
case шаблон 1:
    блок 1
case шаблон 2:
    блок 2
case шаблон 3:
    блок 3
...
case _:
    блок (n) # действие по умолчанию
```


Задания

1 Даны три целых числа. Найти количество положительных чисел в исходном наборе.

2 Даны три целых числа. Найти количество положительных и количество отрицательных чисел в исходном наборе.

3 Даны две переменные вещественного типа: А, В. Перераспределить значения данных переменных так, чтобы в А оказалось меньшее из значений, а в В – большее. Вывести новые значения переменных А и В.

4 Даны две переменные целого типа: А и В. Если их значения не равны, то присвоить каждой переменной сумму этих значений, а если равны, то присвоить переменным нулевые значения. Вывести новые значения переменных А и В.

5 Даны две переменные целого типа: А и В. Если их значения не равны, то присвоить каждой переменной большее из этих значений, а если равны, то присвоить переменным нулевые значения. Вывести новые значения переменных А и В.

6 Даны три числа. Найти наименьшее из них.

7 Даны три числа. Найти среднее из них (т. е. число, расположенное между наименьшим и наибольшим).

8 Даны три числа. Вывести вначале наименьшее, а затем наибольшее из данных чисел.

9 Даны три числа. Найти сумму двух наибольших из них.

10 Даны три переменные вещественного типа: А, В, С. Если их значения упорядочены по возрастанию, то удвоить их; в противном случае заменить значение каждой переменной на противоположное. Вывести новые значения переменных А, В, С.

11 Даны три переменные вещественного типа: А, В, С. Если их значения упорядочены по возрастанию или убыванию, то удвоить их; в противном случае заменить значение каждой переменной на противоположное. Вывести новые значения переменных А, В, С.

12 Даны три целых числа, одно из которых отлично от двух других, равных между собой. Определить порядковый номер числа, отличного от остальных.

13 Даны четыре целых числа, одно из которых отлично от трех других, равных между собой. Определить порядковый номер числа, отличного от остальных.

14 На числовой оси расположены три точки: А, В, С. Определить, какая из двух последних точек (В или С) расположена ближе к А, и вывести эту точку и ее расстояние от точки А.

15 Даны целочисленные координаты точки на плоскости. Если точка совпадает с началом координат, то вывести 0. Если точка не совпадает с началом координат, но лежит на оси ОХ или ОУ, то вывести соответственно 1 или 2. Если точка не лежит на координатных осях, то вывести 3.

Контрольные вопросы

1 Как работает оператор выбора?

2 Как записывается оператор выбора?

3 Как можно выполнить проверку нескольких условий?

10 Лабораторная работа № 10. Программирование

Цель работы: изучить типы данных Python.

Поскольку в Python все является объектом, то не удивительно, что, в отличие от ряда других языков программирования, помимо пользовательских типов (классов) существуют такие типы данных, как функция, файл, модуль, метод и скомпилированный код.

К стандартным (встроенным) типам Python относят [2]:

- 1) none (аналог nullptr из C++);
- 2) булевы значения;
- 3) числа (Numeric type):
 - int – целое число;
 - float – число с плавающей точкой;
 - complex – комплексное число;
- 4) списки (Sequence Type):
 - list – список;
 - tuple – кортеж;
 - range – диапазон;
- 5) строки (Text Sequence Type) – str;
- 6) бинарные списки (Binary Sequence Types):
 - bytes – байты;
 - bytearray – массивы байтов;
 - memoryview – специальные объекты для доступа к внутренним данным объекта через protocol buffer;
- 7) множества (Set Types):
 - .set – множество;
 - frozenset – неизменяемое множество;
- 8) словари (Mapping Types) – dict.

Все типы данных в Python делятся на две категории: изменяемые (mutable) и неизменяемые (immutable).

К неизменяемым типам данных относят: символьные строки, числа, булевы значения, неизменяемые множества и кортежи. Остальные типы данных относятся к изменяемым.

Разница заключается в том, что в случае объявления имени переменной неизменяемого типа ей задается идентификатор, который невозможно изменить.

Задания

- 1 Напишите скрипт, который удаляет задаваемый произвольный символ в строке.
- 2 Напишите скрипт, вычисляющий двумя способами длину строки.
- 3 Напишите скрипт, который позволяет из строки собрать другую по следующим правилам: новая строка должна состоять из двух первых и двух по-

следних элементов исходной строки. Если длина исходной строки меньше двух, то результатом будет пустая строка.

4 Напишите скрипт, который заменяет произвольный символ/букву в строке на «\$».

5 Напишите скрипт, который позволяет инвертировать последовательность элементов в строке.

6 Напишите скрипт, позволяющий из исходной строки собрать две новые. Первая строка должна состоять только из элементов с нечетными индексами исходной строки, а вторая – с четными.

Контрольные вопросы

- 1 Какие из типов данных являются неизменяемыми?
- 2 На какие категории делятся типы данных в Python?
- 3 Какие из перечисленных типов данных являются изменяемыми?

11 Лабораторная работа № 11. Программирование

Цель работы: изучить работу циклов.

Цикл (оператор) `for` перебирает значения, возвращаемые любым итерируемым объектом, т. е. любым объектом, который может сгенерировать последовательность значений. Например, цикл `for` может перебрать элементы списка, кортежа или строки. Итерируемый объект может быть создан функцией `range` или специальной разновидностью функций – генератором:

```
for элемент in последовательность:
... #тело цикла
else: # необязательная часть else
...
# если для выхода из цикла не использовался break
```

Тело цикла выполнится по одному разу для каждого элемента последовательности. Необязательная часть `else` используется достаточно редко и работает аналогично циклу `while`. То же самое относится к командам `break` и `continue`. Таким образом, более полный формат цикла `for` может быть записан следующим образом:

```
for элемент in последовательность:
# тело цикла
if проверка: break
if проверка: continue
else: #блок else
```

Оператор (цикл) `while` используется для итераций в языке Python. Он многократно выполняет блок операторов до тех пор, пока проверка условия в заго-

ловочной части возвращает True. Когда результат проверки становится False, управление переходит на оператор, следующий после блока while. Если с самого начала результат условия выполнения цикла вернет False, то тело цикла никогда не выполнится (оператор while пропускается).

В общем виде оператор while состоит из строки заголовка с условием выполнения цикла, тела цикла с одним или большим количеством операторов с отступами и необязательной части else, которая выполняется, если цикл не прерывается оператором break:

```
while условие: # Проверка цикла
    операторы # Тело цикла
else: # Необязательная часть else
    операторы
#Выполняются, если не произведен
#выход из цикла с помощью break
```

Задание 1

1 Вывести все четные числа в промежутке от 1 до X. Найти сумму этих чисел.

2 Вывести все нечетные числа, делящиеся нацело на 3, в промежутке от 1 до 100. Найти количество этих чисел.

3 Вывести цифру (!!!) $N > 0$, вывести все числа в диапазоне от 1 до 100, делящиеся на N без остатка. Найти произведение выведенных чисел.

4 Вывести цифру (!!!) $N > 0$, вывести таблицу умножения этой цифры на числа от 0 до 10 включительно.

5 Запросить ввести K целых чисел. Определить количество положительных и отрицательных чисел среди введенных.

6 Вывести цифру (!!!) $N > 0$, вывести все числа в диапазоне от 1 до 100, заканчивающиеся на эту цифру. Найти количество выведенных чисел.

7 Запросить имя пользователя и вывести его на экран N раз, по три имени в строке (в последней может быть меньше).

8 Вывести на экран квадратные корни чисел от 1 до A ($A > 1$). Найти среднее арифметическое этих чисел.

9 Вывести все нечетные числа в промежутке от M до 1. Найти произведение этих чисел.

10 Вывести все числа, делящиеся нацело на 4, в промежутке от 1 до 100. Найти среднее геометрическое этих чисел.

11 Вывести цифру (!!!) $N > 0$, вывести все числа в диапазоне от 1 до Z, не заканчивающиеся на эту цифру. Найти количество выведенных чисел.

12 Факториал числа N ($N!$) – это произведение чисел от 1 до N включительно. Ввести число N ($N < 20$). Вывести на экран последовательность « $1*2*3*4*...N$ » и вывести значение факториала.

13 Вывести на экран квадраты чисел от -K до K. Найти среднее арифметическое этих чисел.

14 Запросить ввести N целых чисел. Определить максимальное число среди введенных.

15 Ввести число N ($N < 10$). Вывести на экран последовательность « $1+2+3+4+\dots+N$ » и вывести значение их суммы.

Задание 2

1 Построить таблицу значений для функции $f(x) = x - \sin(x)$ на отрезке $[0; \pi/2]$ с числом разбиений отрезка $m = 10$.

2 Построить таблицу значений для функции $f(x) = \sin(x)$ на отрезке $[\pi/4; \pi/2]$ с числом разбиений отрезка $m = 15$.

3 Построить таблицу значений для функции $f(x) = \cos(x)$ на отрезке $[\pi/3; 2\pi/3]$ с числом разбиений отрезка $m = 20$.

4 Построить таблицу значений для функции $f(x) = \operatorname{tg}(x)$ на отрезке $[0; \pi/4]$ с числом разбиений отрезка $m = 10$.

5 Построить таблицу значений для функции $f(x) = \operatorname{ctg}(x)$ на отрезке $[\pi/4; \pi/2]$ с числом разбиений отрезка $m = 15$.

6 Построить таблицу значений для функции $f(x) = \arcsin(x)$ на отрезке $[0; 1]$ с числом разбиений отрезка $m = 20$.

7 Построить таблицу значений для функции $f(x) = \arccos(x)$ на отрезке $[0,5; 1]$ с числом разбиений отрезка $m = 10$.

8 Построить таблицу значений для функции $f(x) = \operatorname{arctg}(x)$ на отрезке $[2; 7]$ с числом разбиений отрезка $m = 15$.

9 Построить таблицу значений для функции $f(x) = \sin(x) - \cos(x)$ на отрезке $[0; \pi/2]$ с числом разбиений отрезка $m = 20$.

10 Построить таблицу значений для функции $f(x) = x \sin(x)$ на отрезке $[0; 3\pi]$ с числом разбиений отрезка $m = 10$.

11 Построить таблицу значений для функции $f(x) = \sin(1/x)$ на отрезке $[\pi/8; 2/\pi]$ с числом разбиений отрезка $m = 10$.

12 Построить таблицу значений для функции $f(x) = x \cos(x)$ на отрезке $[0,2; \pi/3]$ с числом разбиений отрезка $m = 15$.

13 Построить таблицу значений для функции $f(x) = |\sin(x)|$ на отрезке $[\pi/2; 3\pi/2]$ с числом разбиений отрезка $m = 20$.

14 Построить таблицу значений для функции $f(x) = (\sin(x))^2$ на отрезке $[\pi/3; 2\pi/3]$ с числом разбиений отрезка $m = 15$.

15 Построить таблицу значений для функции $f(x) = \cos(x^2)$ на отрезке $[0; \pi]$ с числом разбиений отрезка $m = 10$.

Контрольные вопросы

- 1 Какие циклы используются в Python?
- 2 Расскажите про цикл while.
- 3 Расскажите про цикл for.

12 Лабораторная работа № 12. Программирование

Цель работы: получить навыки работы со строками.

Для работы с текстом в Python предусмотрен специальный строковый тип данных `str`. Строковые объекты создаются, если текст поместить в одиночные апострофы или двойные кавычки:

```
>>> 'hello'
'hello'
>>> "Hello"
'Hello'
```

Строковый тип данных применяется для хранения текстовой информации и произвольных значений байтов. Строки относятся к так называемым последовательностям – позиционно упорядоченным коллекциям объектов, в которых объекты располагаются в порядке слева направо, сохраняют свои позиции и могут извлекаться по их относительному номеру.

Таким образом, строки представляют собой последовательность из односимвольных строк (также к последовательностям в Python относятся списки и кортежи) и поддерживают операции, которые предполагают наличие позиционного порядка среди элементов: индексация, срез, конкатенация и повторение [2].

Для работы со строками в Python предусмотрено большое число встроенных функций. Рассмотрим, например, функцию `len`, которая определяет длину строки, переданной ей в качестве аргумента: функция `len` возвращает количество элементов для различных типов данных.

Пример вызова функции `len` для строкового аргумента:

```
>>> len('Привет!')
7
```

С помощью операции конкатенации (оператор «+» для строк) Python позволяет объединить несколько строк в одну. Строки можно не только складывать, но и умножать [1].

Задания

1 Напишите программу для определения суммы и произведения трех целочисленных значений, введенных с клавиатуры.

Пример выполнения программы.

Введите первое число: 1.

Введите второе число: 4.

Введите третье число: 7.

Сумма введенных чисел: 12.

Произведение введенных чисел: 28.

2 Напишите программу, проверяющую целое число на четность. Реализуйте ее в виде вызова собственной функции.

3 Напишите программу, которая по атомному номеру элемента в периодической таблице химических элементов определяет его название. В качестве входных значений необходимо использовать атомные номера Li, Mn, Hg, Cl.

4 Напишите собственную чистую функцию `my_abs` для поиска абсолютного значения (напомним, что чистыми называют функции, которые для одного и того же набора входных значений возвращают одинаковый результат и не обладают побочными эффектами, т. е. не осуществляют внутри себя ввод-вывод и пр.).

5 Напишите программу, определяющую максимальное из двух введенных чисел. Реализуйте ее в виде вызова функции, возвращающей большее из двух переданных ей чисел.

Контрольные вопросы

1 Что такое строковый тип данных?

2 Какие операции осуществляются над строками?

13 Лабораторная работа № 13. Программирование

Цель работы: получить навыки работы со списками (массивами, матрицами).

Список (`list`) в Python является объектом, поэтому может быть присвоен переменной (напоминаем, что в переменной хранится адрес объекта).

Представим список для задачи с курсом валют:

```
>>> e = [66.3309, 66.4438, 66.7617, 67.0820, 67.1920]
```

```
>>> e
```

```
[66.3309, 66.4438, 66.7617, 67.0820, 67.1920]
```

Список позволяет хранить разнородные данные, обращаться к которым можно через имя списка. На месте элементов списка могут находиться сложные выражения.

Обращаться к отдельным элементам списка можно по их индексу (позиции), начиная с нуля (по аналогии со строками) [1].

Задание

Задания для выполнения лабораторной работы представлены в таблицах 13.1 и 13.2.

Таблица 13.1 – Одномерные массивы (списки)

Номер варианта	Задание
1	<p>В одномерном массиве, состоящем из n вещественных элементов, вычислить:</p> <ol style="list-style-type: none"> 1) сумму отрицательных элементов массива; 2) произведение элементов массива, расположенных между максимальным и минимальным элементами. <p>Упорядочить элементы массива по возрастанию</p>
2	<p>В одномерном массиве, состоящем из n вещественных элементов, вычислить:</p> <ol style="list-style-type: none"> 1) сумму положительных элементов массива; 2) произведение элементов массива, расположенных между максимальным по модулю и минимальным по модулю элементами. <p>Упорядочить элементы массива по убыванию</p>
3	<p>В одномерном массиве, состоящем из n целых элементов, вычислить:</p> <ol style="list-style-type: none"> 1) произведение элементов массива с четными номерами; 2) сумму элементов массива, расположенных между первым и последним нулевыми элементами. <p>Преобразовать массив таким образом, чтобы сначала располагались все положительные элементы, а потом – все отрицательные (элементы, равные 0, считать положительными)</p>
4	<p>В одномерном массиве, состоящем из n вещественных элементов, вычислить:</p> <ol style="list-style-type: none"> 1) сумму элементов массива с нечетными номерами; 2) сумму элементов массива, расположенных между первым и последним отрицательными элементами. <p>Сжать массив, удалив из него все элементы, модуль которых не превышает 1. Освободившиеся в конце массива элементы заполнить нулями</p>
5	<p>В одномерном массиве, состоящем из n вещественных элементов, вычислить:</p> <ol style="list-style-type: none"> 1) максимальный элемент массива; 2) сумму элементов массива, расположенных до последнего положительного элемента. <p>Сжать массив, удалив из него все элементы, модуль которых находится в интервале $[a; b]$. Освободившиеся в конце массива элементы заполнить нулями</p>
6	<p>В одномерном массиве, состоящем из n вещественных элементов, вычислить:</p> <ol style="list-style-type: none"> 1) минимальный элемент массива; 2) сумму элементов массива, расположенных между первым и последним положительными элементами. <p>Преобразовать массив таким образом, чтобы сначала располагались все элементы, равные нулю, а потом – все остальные</p>
7	<p>В одномерном массиве, состоящем из n целых элементов, вычислить:</p> <ol style="list-style-type: none"> 1) номер максимального элемента массива; 2) произведение элементов массива, расположенных между первым и вторым нулевыми элементами. <p>Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в нечетных позициях, а во второй половине – элементы, стоявшие в четных позициях</p>
8	<p>В одномерном массиве, состоящем из n вещественных элементов, вычислить:</p> <ol style="list-style-type: none"> 1) номер минимального элемента массива; 2) сумму элементов массива, расположенных между первым и вторым отрицательными элементами. <p>Преобразовать массив таким образом, чтобы сначала располагались все элементы, модуль которых не превышает 1, а потом – все остальные</p>

Окончание таблицы 13.1

Номер варианта	Задание
9	<p>В одномерном массиве, состоящем из n вещественных элементов, вычислить:</p> <ol style="list-style-type: none"> 1) максимальный по модулю элемент массива; 2) сумму элементов массива, расположенных между первым и вторым положительными элементами. <p>Преобразовать массив таким образом, чтобы элементы, равные нулю, располагались после всех остальных</p>
10	<p>В одномерном массиве, состоящем из n целых элементов, вычислить:</p> <ol style="list-style-type: none"> 1) минимальный по модулю элемент массива; 2) сумму модулей элементов массива, расположенных после первого элемента, равного нулю. <p>Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в четных позициях, а во второй половине – элементы, стоявшие в нечетных позициях</p>
11	<p>В одномерном массиве, состоящем из n вещественных элементов, вычислить:</p> <ol style="list-style-type: none"> 1) номер минимального по модулю элемента массива; 2) сумму модулей элементов массива, расположенных после первого отрицательного элемента. <p>Сжать массив, удалив из него все элементы, величина которых находится в интервале $[a; b]$. Освободившиеся в конце массива элементы заполнить нулями</p>
12	<p>В одномерном массиве, состоящем из n вещественных элементов, вычислить:</p> <ol style="list-style-type: none"> 1) номер максимального по модулю элемента массива; 2) сумму элементов массива, расположенных после первого положительного элемента. <p>Преобразовать массив таким образом, чтобы сначала располагались все элементы, целая часть которых лежит в интервале $[a; b]$, а потом – все остальные</p>
13	<p>В одномерном массиве, состоящем из n вещественных элементов, вычислить:</p> <ol style="list-style-type: none"> 1) количество элементов массива, лежащих в диапазоне от A до B; 2) сумму элементов массива, расположенных после максимального элемента. <p>Упорядочить элементы массива по убыванию модулей элементов</p>
14	<p>В одномерном массиве, состоящем из n вещественных элементов, вычислить:</p> <ol style="list-style-type: none"> 1) количество элементов массива, равных 0; 2) сумму элементов массива, расположенных после минимального элемента. <p>Упорядочить элементы массива по возрастанию модулей элементов</p>
15	<p>В одномерном массиве, состоящем из n вещественных элементов, вычислить:</p> <ol style="list-style-type: none"> 1) количество элементов массива, больших C; 2) произведение элементов массива, расположенных после максимального по модулю элемента. <p>Преобразовать массив таким образом, чтобы сначала располагались все отрицательные элементы, а потом – все положительные (элементы, равные 0, считать положительными)</p>

Таблица 13.2 – Двумерные массивы (списки)

Номер варианта	Задание
1	Дана целочисленная прямоугольная матрица. Определить: 1) количество строк, не содержащих ни одного нулевого элемента; 2) максимальное из чисел, встречающихся в заданной матрице более одного раза
2	Дана целочисленная прямоугольная матрица. Определить количество столбцов, не содержащих ни одного нулевого элемента. Характеристикой строки целочисленной матрицы назовем сумму ее положительных четных элементов. Переставляя строки заданной матрицы, расположить их в соответствии с ростом характеристик
3	Дана целочисленная прямоугольная матрица. Определить: 1) количество столбцов, содержащих хотя бы один нулевой элемент; 2) номер строки, в которой находится самая длинная серия одинаковых элементов
4	Дана целочисленная квадратная матрица. Определить: 1) произведение элементов в тех строках, которые не содержат отрицательных элементов; 2) максимум среди сумм элементов диагоналей, параллельных главной диагонали матрицы
5	Дана целочисленная квадратная матрица. Определить: 1) сумму элементов в тех столбцах, которые не содержат отрицательных элементов; 2) минимум среди сумм модулей элементов диагоналей, параллельных побочной диагонали матрицы
6	Дана целочисленная прямоугольная матрица. Определить: 1) сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент; 2) номера строк и столбцов всех седловых точек матрицы. <i>Примечание</i> – Матрица A имеет седловую точку A_{ij} , если A_{ij} является минимальным элементом в i -й строке и максимальным в j -м столбце
7	Для заданной матрицы размером 8×8 найти такие k , что k -я строка матрицы совпадает с k -м столбцом. Найти сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент
8	Характеристикой столбца целочисленной матрицы назовем сумму модулей его отрицательных нечетных элементов. Переставляя столбцы заданной матрицы, расположить их в соответствии с ростом характеристик. Найти сумму элементов в тех столбцах, которые содержат хотя бы один отрицательный элемент
9	Соседями элемента A_{ij} в матрице назовем элементы A_{kl} с $i - 1 \leq k \leq i + 1$, $j - 1 \leq l \leq j + 1$, $(k, l) \neq (i, j)$. Операция сглаживания матрицы дает новую матрицу того же размера, каждый элемент которой получается как среднее арифметическое имеющихся соседей соответствующего элемента исходной матрицы. Построить результат сглаживания заданной вещественной матрицы размером 10×10 . В сглаженной матрице найти сумму модулей элементов, расположенных ниже главной диагонали
10	Элемент матрицы называется локальным минимумом, если он строго меньше всех имеющихся у него соседей. Подсчитать количество локальных минимумов заданной матрицы размером 10×10 . Найти сумму модулей элементов, расположенных выше главной диагонали

Окончание таблицы 13.2

Номер варианта	Задание
11	Коэффициенты системы линейных уравнений заданы в виде прямоугольной матрицы. С помощью допустимых преобразований привести систему к треугольному виду. Найти количество строк, среднее арифметическое элементов которых меньше заданной величины
12	Уплотнить заданную матрицу, удаляя из нее строки и столбцы, заполненные нулями. Найти номер первой из строк, содержащих хотя бы один положительный элемент
13	Осуществить циклический сдвиг элементов прямоугольной матрицы на n элементов вправо или вниз (в зависимости от введенного режима). n может быть больше количества элементов в строке или в столбце
14	Осуществить циклический сдвиг элементов прямоугольной матрицы размерности $M \times N$ вправо на k элементов следующим образом: элементы 1-й строки сдвигаются в последний столбец сверху вниз, из него – в последнюю строку справа налево, из нее – в первый столбец снизу вверх, из него – в первую строку; для остальных элементов – аналогично
15	Дана целочисленная прямоугольная матрица. Определить номер первого из столбцов, содержащих хотя бы один нулевой элемент. Характеристикой строки целочисленной матрицы назовем сумму ее отрицательных четных элементов. Переставляя строки заданной матрицы, расположить их в соответствии с убыванием характеристик
<i>Примечание</i> – Исходные данные должны включать и положительные числа, и отрицательные числа, и нули. Массив заполнить случайными числами	

Контрольные вопросы

- 1 Что такое список?
- 2 Как записываются в Python списки (массив, матрица)?
- 3 В чем заключается отличие списка от многомерного списка?

14 Лабораторная работа № 14. Программирование

Цель работы: научиться работать с функциями.

Функции в Python относятся к объектам первого класса. Их можно присваивать переменным, хранить в структурах данных, передавать в качестве аргументов другим функциям и даже возвращать в качестве значений из других функций. Они могут принимать произвольное количество аргументов или не принимать их вовсе.

Функции позволяют использовать код, заключенный в них, сколько угодно

раз в процессе выполнения программы и являются альтернативой любимого метода начинающих программистов: «copy&paste». Они также позволяют разбивать сложные системы на составные части, что упрощает процесс проектирования систем.

На рисунке 14.1 приведены примеры всех операторов и выражений, которые тем или иным образом связаны с функциями.

Оператор или выражение	Пример
Выражения вызовов	<code>my_test_func('text', meat=ham, *rest)</code>
<code>def</code>	<code>def print_text(message) : print('Hi, ' + message)</code>
<code>return</code>	<code>def add_test(a, b=1, *c): return a + b + c[0]</code>
<code>global</code>	<code>x = 'old' def changer(): global x; x = 'new'</code>
<code>nonlocal</code>	<code>def outer(): x = 'old' def changer (): nonlocal x; x = 'new'</code>
<code>yield</code>	<code>def test(x): for i in range (x): yield i ** 2</code>
<code>lambda</code>	<code>funcs = [lambda x: x**2, lambda x: x**3]</code>

Рисунок 14.1 – Операторы и выражения, связанные функциями

Функции и методы класса в Python начинаются с оператора `def`. В отличие от функций в компилируемых языках, таких как C, `def` представляет собой исполняемый оператор. Это означает, что написанная в коде функция не существует до тех пор, пока интерпретатор Python не встретит и не выполнит ее. Функции можно вкладывать внутрь операторов `if`, циклов `while` и даже других функций. При типовом применении операторы `def` помещаются в файлы модулей и естественным образом выполняются для генерации функций, когда файл модуля, где они находятся, импортируется в первый раз.

Когда интерпретатор Python достигает оператора `def` и выполняет его, генерируется новый объект, который присваивается имени функции. Как и со всеми присваиваниями, имя функции становится ссылкой на объект функции. Объекты функций также могут иметь произвольные определяемые пользователем атрибуты, присоединяемые к ним для регистрации данных.

Помимо этого, функции можно также создавать с помощью `lambda`-выражений – средства, которое позволяет встраивать определения функций в места, где оператор `def` синтаксически не допускается.

Когда функция вызывается, вызывающий код приостанавливается до тех пор, пока функция не завершит свою работу и не возвратит управление обрат-

но. Функции, которые вычисляют значение, возвращают его вызывающему коду посредством оператора `return` (`return` без значения просто возвращает управление в вызывающий код и отправляет стандартный результат `None`).

`yield` – выражение, но оно обычно используется как оператор и отправляет результирующий объект вызывающему коду, при этом запоминая место, где он остановился. В функциях, известных как генераторы, можно также использовать `yield`, чтобы посылать обратно значение и предохранять их состояние, так что они смогут возобновлять работу позже для производства серии результатов.

Оператор `global` объявляет переменные уровня модуля, предназначенные для присваивания. По умолчанию все объявляемые переменные в функции являются локальными и существуют только во время ее выполнения. Чтобы присвоить значение переменной из включающего модуля, необходимо расширить ее область видимости посредством оператора `global` внутри функции.

Оператор `nonlocal` объявляет переменные включающей функции, предназначенные для присваивания. Проще говоря, он позволяет функции присваивать значения переменной, которая уже существует в области видимости функции более высокого порядка, которая включает в себя текущую функцию [2].

Задание

Решить лабораторные работы № 9 и 11, используя функции.

Контрольные вопросы

- 1 Что такое функции?
- 2 Для чего используются функции?
- 3 Какие параметры можно передавать в функции?

Список литературы

1 **Федоров, Д. Ю.** Программирование на языке высокого уровня Python: учебное пособие / Д. Ю. Федоров. – 5-е изд., перераб. и доп. – Москва: Юрайт, 2024. – 227 с.

2 **Чернышев, С. А.** Основы программирования на Python: учебное пособие / С. А. Чернышев. – 2-е изд., перераб. и доп. – Москва: Юрайт, 2024. – 349 с.