Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра инфокоммуникаций

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №2.1**

**дисциплины «Программирование на Python»**

**Вариант\_\_\_**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнил:  Иващенко Олег Андреевич  2 курс, группа ИВТ-б-о-22-1,  09.03.02 «Информационные и вычислительные машины», направленность (профиль) «Программное обеспечение средств вычислительной техники и автоматизированных систем»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | | Руководитель практики:  Воронкин Роман Александрович, доцент кафедры инфокоммуникаций  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | |  | |

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_ Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ставрополь, 2023 г.

**Тема**: «Основы языка Python»

**Цель**: Исследование процесса установки и базовых возможностей языка Python версии 3.x.

Порядок выполнения работы

Таблица 1 – Код программы name.py

|  |
| --- |
| if \_\_name\_\_ == "\_\_main\_\_":  name = input('Enter your name: ')  age = input('Enter your age: ')  residance = input('Enter your place of residance: ')    print("\nName: {0}".format(name))  print("Age: {0}".format(age))  print("Place of residence: {0}".format(residance)) |

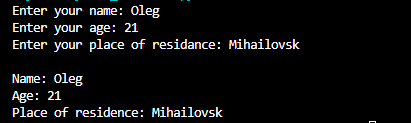


Рисунок 1 – Вывод программы name.py

Таблица 2 – Код программы arithmetic.py

|  |
| --- |
| if \_\_name\_\_ == "\_\_main\_\_":  answer = input("4 \* 100 - 54 = ")  print("\nCorrect answer: 346")  if (int(answer) == 346):  print("Your answer is correct ({0})".format(answer))  else:  print("Yout answer isn't correct ({0})".format(answer)) |

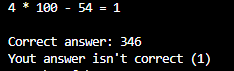


Рисунок 2 – Вывод программы arithmetic.py в случае неправильного ответа

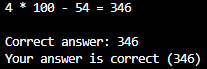


Рисунок 3 – Вывод программы arithmetic.py в случае правильного ответа

Таблица 3 – Код программы numbers.py

|  |
| --- |
| if \_\_name\_\_ == "\_\_main\_\_":  first\_number = input('Enter first number: ')  second\_number = input('Enter second number: ')  third\_number = input('Enter third number: ')  fourth\_number = input('Enter fourth number: ')    first\_sum = int(first\_number) + int(second\_number)  second\_sum = int(third\_number) + int(fourth\_number)  print("\nResult of division: %.2f" % (first\_sum / second\_sum)) |

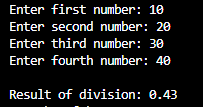


Рисунок 4 – Вывод программы numbers.py

Индивидуальное задание. Возраст Тани – лет, а возраст Мити – лет. Найти их средний возраст, а также определить, на сколько отличается возраст каждого ребенка от среднего значения.

Таблица 4 – Код программы individual.py

|  |
| --- |
| if \_\_name\_\_ == "\_\_main\_\_":  age\_x = input("Enter Tanya's age: ")  age\_y = input("Enter Mitya's age: ")  average\_age = (int(age\_x) + int(age\_y)) / 2  difference\_age\_x = int(age\_x) - average\_age  difference\_age\_y = int(age\_y) - average\_age    print("\nAverage age: {0}".format(average\_age))  print("Difference with average age and:")  print("Tanya's age: {0}".format(difference\_age\_x))  print("Mitya's age: {0}".format(difference\_age\_y)) |

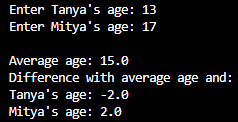


Рисунок 5 – Результат вывода программы indivivual.py

Задание повышенной сложности: Даны целые числа h, m, s (0 < h <= 23, 0<= m <= 59, 0 <= s <= 59), указывающие момент времени: «часов, минут, секунд». Определить угол (в градусах) между положением часовой стрелки в начале суток и в указанный момент времени.

Таблица 5 – Код программы inc\_def.py

|  |
| --- |
| if \_\_name\_\_ == "\_\_main\_\_":  hours = input("Enter hour: ")  minutes = input("Enter minutes: ")  seconds = input("Enter seconds: ")  hours = int(hours) \* 30  minutes = int(minutes) / 2  seconds = int(seconds) / 120  degrees = hours + minutes + seconds  if degrees > 360:  degrees = degrees - 360  print("Clockwise angle: %.2f" % (degrees)) |

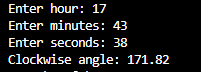


Рисунок 6 – Вывод программы inc\_def.py

Контрольные вопросы

1. Опишите основные этапы установки Python в Windows и Linux.

Для Windows: Скачать установщик с официального сайта Python, запустить установщик, выбрать нужные параметры, установив опцию «Add Python to PATH».

Для Linux: В большинстве дистрибутивов Linux Python предустановлен. Но если он отсутствует, то он устанавливается с помощью пакетного менеджера (sudo apt-get install python3). После установки можно проверить версию, используя команду python3 --version.

1. В чём отличие пакета Anaconda от пакета Python, скачиваемого с официального сайта?

Anaconda представляет собой дистрибутив, ориентированный на науку о данных и анализ данных. Он включает в себя не только интерпретатор Python, но и множество предустановленных библиотек и инструментов для научных вычислений, машинного обучения и обработки данных.

Anaconda использует свой собственный инструмент управления пакетами – conda, в то время, как Python использует pip.

1. Как осуществить проверку работоспособности пакета Anaconda?

Запустить инструмент Anaconda Navigator, запустить Jupyter Notebook, который откроет интерфейс Jupiter в веб-браузере, создать новый блокнот, ввести какой-либо простейший код с выводом. При запуске кода должен появиться вывод.

1. Как задать используемый интерпретатор языка Python в IDE PyCharm?

Запустить проект в PyCharm, перейти в настройки проекта, в меню настроек найти раздел «Project: [название проекта]», выбрать его и найти подраздел «Python Interpreter».

1. Как осуществить запуск программы с помощью IDE PyCharm?

Открыть PyCharm, выбрать нужный проект, в верхней части программы найти кнопку с зелёной стрелкой (Run) и выбрать из выпадающего списка конфигурации запуска.

1. В чём суть интерактивного и пакетного режимов работы Python?

Интерактивный режим Python предоставляет возможность вводить команды Python непосредственно в интерпретаторе и видеть результаты выполнения каждой команды. Для его запуска нужно ввести в терминал или командную строку команду «python».

Пакетный режим Python предполагает запуск программы Python из файла, который в себе содержит последовательность команд. Этот режим читает и выполняет команды из файла поочерёдно, начиная с первой строки и заканчивая последней.

1. Почему язык программирования Python называется языком динамической типизации?

Динамическая типизация означает, что тип данных переменной определяется автоматически во время выполнения программы, а не в момент компиляции. Также тип переменной может изменяться в процессе выполнения программы. Иными словами, в Python отсутствует явное указание типа данных для переменных.

1. Какие существуют основные типы в языке программирования Python?

В Python существует несколько основных встроенных типов данных, которые предоставляют различные способы хранения и использования данными:

* Числовые типы (int, float, complex);
* Строковые типы (str);
* Списки (list);
* Кортежи (tuple);
* Множества (set);
* Словари (dict);
* Логический тип (bool).

1. Как создаются объекты в памяти? Каково их устройство? В чём заключается процесс объявления новых переменных и работа операции присваивания?

При создании объекта Python выделяет память для хранения его данных и метаданных. После выделения память инициализируются данные объекта в соответствии с его типом. Например, для целого числа (int) выделяется память под целочисленное значение.

Переменная в Python является ссылкой на объект. При объявлении переменной происходит создание объекта и присвоение переменной ссылки на этот объект.

В Python операция присваивания связывает переменную с объектом. Если переменная уже указывает на какой-то объект, то новое присваивание изменит эту связь.

1. Как получить список ключевых слов в Python?

В Python существует встроенный модуль «keyword», который предоставляет функциональность для работы с ключевыми словами языка. Пример использования модуля:

|  |
| --- |
| import keyword  keywords\_list = keyword.kwlist  print(keywords\_list) |

Этот код выведет список ключевых слов в Python.

1. Каково назначение функций id() и type()?

Функция id() используется для просмотра, на объект с каким идентификатором ссылается данная переменная.

Функция type() используется для определения типа переменной.

1. Что такое изменяемые и неизменяемые типы в Python?

К неизменяемым (immutable) типам относятся:

* Целые числа (int);
* Числа с плавающей запятой (float);
* Комплексные числа (complex);
* Логические переменные (bool);
* Кортежи (tuple);
* Строки (str);
* Неизменяемые множества (frozen set).

К изменяемым (mutable) типам относятя:

* Списки (list);
* Множества (set);
* Словари (dict).

Неизменяемость типа данных означает, что созданный объект больше не изменяется.

1. Чем отличаются операции деления и целочисленного деления?

Деление (/) возвращает число с плавающей точкой, даже если оба операнда – целые числа.

Целочисленное деление (//) возвращает целое число, отбрасывая дробную часть и округляя число вниз.

1. Какие имеются средства в языке Python для работы с комплексными числами?

Для работы с комплексными числами в Python есть встроенный тип данных «complex». Комплексные числа в Pyhton представляются в виде суммы действительной и мномой частей, обе части являются вещественными.

Комплексное число создаётся с помощью функции complex(a, b), в которую в качестве первого аргумента передаётся действительная часть, в качестве второго – мнимая. Также можно записать число в виде «a + bj» (например, z = 1 + 2j).

Комплексные числа можно складывать, вычитать, умножать, делить и взводить в степень. У комплексного числа можно извлечь действительную (z.real) и мнимую (z.imag) части. Для получения комплексносопряжённого числа используется метод conjugate().

1. Каково назначение и основные функции библиотеки (модуля) math? По аналогии с модулем math изучите самостоятельно назначение и основные функции модуля cmath.

В стандартную поставку Python входит библиотека math, в которой содержится большое количество часто используемых математических функций. Для работы с данным модулем его нужно импортировать (import math).

Основные функции модуля:

* math.ceil(x) – округляет число в большую сторону;
* math.fabs(x) – возвращает абсолютное значение числа;
* math.factorial(x) – возвращает факториал x;
* math.floor(x) – округляет число в меньшую сторону;
* math.exp(x) – вычисляет e\*\*x;
* math.log2(x) – возвращает логарифм по основанию 2;
* math.log10(x) – возвращает логарифм по основанию 10;
* math.log(x, [base]) – по умолчанию возвращает логарифм по основанию e, дополнительно можно указать основание логарифма;
* math.pow(x, y) – возвращает x в степени y;
* math.sqrt(x) – возвращает квадратный корень от x;
* math.cos(x) – возвращает косинус от x;
* math.sin(x) – возвращает синус от x;
* math.tan(x) – возвращает тангенс от x;
* math.acos(x) – возвращает арккосинус от x;
* math.asin(x) – возвращает арксинус от x;
* math.atan(x) – возвращает арктангенс от x;
* math.pi – возвращает число Пи;
* math.e – возвращает число e.

Модуль cmath представляет собой аналогичные math функции для комплексных чисел.

1. Каково назначение именных параметров sep и end в функции print()?

Параметр sep устанавливает отличный от пробела разделитель строк.

Параметр end указывает, что выводить в конце строки.

1. Каково назначение метода format()? Какие ещё существуют средства для форматирования строк в Python? Примечание: в дополнение к рассмотренных средствам изучите самостоятельно работу с f-строками в Python.

Метод format() в Python предназначен для форматирования строк. Этот метод позволяет вставлять значения переменных в строки и контролировать их представление.

Помимо метода format() в Python существуют другие способы форматирования строк:

* f-строки – синтаксис f-строк позволяет вставлять значения переменных прямо в строку с префиксом «f» или «F»;
* Метод «%» - этот метод использует символ «%» для форматирования строк, подобно форматированию строк в языке программирования C.
* Метод join() – позволяет объединять строки из списка.

1. Каким образом осуществить ввод в консоли значения целочисленной и вещественной переменной в языке Python?

Для ввода в Python существует метод input(). С помощью него можно записать в переменную значение, а при помощи функций int() и float() можно выбрать, какой это тип данных должен использоваться.

**Выводы**: В процессе выполнения лабораторной работы был исследован процесс установки Python, были написаны 5 программ, изучены основы программирования на языке Python.