

Северо-Кавказский федеральный университет

Кафедра инфокоммуникаций СКФУ

Отчет

По лабораторной работе №2

По предмету: «Основы
кроссплатформенного программирования»

Исполнитель:

Студента группы ИТС-б-з-22-1

Направление подготовки 11.03.02
Инфокоммуникационные
технологии и системы связи

Пальников Станислав Петрович

(Ф.И.О.)

Руководитель дисциплины:

Воронкин Роман Александрович

Ставрополь, 2023

Ответы на контрольные опросы:

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

А) Запустите скачанный установочный файл. Б). Выберите способ установки. В данном окне предлагается два варианта Install Now и Customize installation. При выборе Install Now, Python установится в папку по указанному пути. Помимо самого интерпретатора будет установлен IDLE (интегрированная среда разработки), pip (пакетный менеджер) и документация, а также будут созданы соответствующие ярлыки и установлены связи файлов, имеющие расширение .py с интерпретатором Python. Customize installation – это вариант настраиваемой установки. Опция Add python 3.5 to PATH нужна для того, чтобы появилась возможность запускать интерпретатор без указания полного пути до исполняемого файла при работе в командной строке. В). Отметьте необходимые опции установки (доступно при выборе Customize installation). На этом шаге нам предлагается отметить дополнения, устанавливаемые вместе с интерпретатором Python. Рекомендуется выбрать все опции. Documentation – установка документаций. pip – установка пакетного менеджера pip. Tcl/Tk and IDLE – установка интегрированной среды разработки (IDLE) и библиотеки для построения графического интерфейса (tkinter). Г) Выберите место установки (доступно при выборе Customize installation) Помимо указания пути, данное окно позволяет внести дополнительные изменения в процесс установки с помощью опций: Install for all users – Установить для всех пользователей. Если не выбрать данную опцию, то будет предложен вариант инсталляции в папку пользователя, устанавливающего интерпретатор. Associate files with Python – Связать файлы, имеющие расширение .py, с Python. При выборе данной опции будут внесены изменения в Windows, позволяющие запускать Python скрипты по двойному щелчку мыши. Create shortcuts for installed applications – Создать ярлыки для запуска приложений. Add Python to environment variables – Добавить пути до интерпретатора Python в переменную PATH. Precompile standard library – Провести прекомпиляцию стандартной библиотеки. Последние два пункта связаны с загрузкой компонентов для отладки, их мы устанавливать не будем. Д) После успешной установки вас ждет следующее сообщение.

Установка Python в Linux Чаще всего интерпретатор Python уже входит в состав дистрибутива. Это можно проверить набрав в терминале `$ python` или `$ python3`. В первом случае, вы запустите Python 2 во втором – Python 3. В

будущем, скорее всего, во всех дистрибутивах Linux, включающих Python, будет входить только третья версия. Если у вас, при попытке запустить Python, выдается сообщение о том, что он не установлен, или установлен, но не тот, что вы хотите, то у вас есть два пути: а) собрать Python из исходников; б) взять из репозитория. Для установки из репозитория в Ubuntu воспользуйтесь командой `$ sudo apt-get install python3`.

2. В чем отличие пакета Anaconda от пакета Python, скачиваемого с официального сайта?

Для удобства запуска примеров и изучения языка Python, настоятельно рекомендуется установить на свой ПК пакет Anaconda. Этот пакет включает в себя интерпретатор языка Python (есть версии 2 и 3), набор наиболее часто используемых библиотек и удобную среду разработки и исполнения, запускаемую в браузере.

3. Как осуществить проверку работоспособности пакета Anaconda?

Для выполнения проверки работоспособности Anaconda необходимо вначале запустить командный процессор с поддержкой виртуальных окружений Anaconda. В Windows это можно сделать выбрав следующий пункт главного меню системы Пуск Anaconda3 (64-bit) Anaconda Prompt. В появившейся командной строке необходимо ввести `> jupyter notebook` в результате чего отобразится процесс загрузки веб-среды Jupyter Notebook после чего запустится веб-сервер и среда разработки в браузере. Создайте ноутбук для разработки, для этого нажмите на кнопку New (в правом углу окна) и в появившемся списке выберите Python. В результате будет создана новая страница в браузере с ноутбуком. Введите в первой ячейке команду и нажмите `Alt+Enter` на клавиатуре. Ниже ячейки должна появиться соответствующая надпись. `print("Hello, World!")`

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

1. Перейдите к Settings.
2. Перейдите в раздел Project Interpreter .
3. Нажмите на кнопку плюс (+) и выберите Local опцию.
4. Найдите интерпретатор, в Linux он используется в `/usr/bin/python` или `/usr/bin/pythonX.Y`, где X и Y - это версия. В Windows он используется в `C:\python`.
5. На последнем шаге сохраните настройки, и вы настроите интерпретатор.

5. Как осуществить запуск программы с помощью IDE PyCharm?

Запустите PyCharm и выберите Create New Project в появившемся окне.

Укажите путь до проекта Python и интерпретатор, который будет использоваться для запуска и отладки.

Добавьте Python файл в проект.

Введите код программы

Запустите программу.

6. В чем суть интерактивного и пакетного режимов работы Python?

В интерактивный режим можно войти, набрав в командной строке `> python` или `> python3`. В результате Python запустится в интерактивном режиме и будет ожидать ввод команд пользователя.

Если же у вас есть файл с исходным кодом на Python, и вы его хотите запустить, то для этого нужно в командной строке вызвать интерпретатор Python и в качестве аргумента передать ваш файл. Например, для файла с именем `test.py` процедура запуска будет выглядеть так: `> python test.py`

7. Почему язык программирования Python называется языком динамической типизации?

Неявная типизация означает, что при объявлении переменной вам не нужно указывать её тип, при явной – это делать необходимо. Тип переменной определяется непосредственно при выполнении программы.

8. Какие существуют основные типы в языке программирования Python?

1) None (неопределенное значение переменной)

2) Логические переменные (Boolean Type)

3) Числа (Numeric Type) 1. `int` – целое число 2. `float` – число с плавающей точкой 3. `complex` – комплексное число 4. Списки (Sequence Type) 1. `list` – список 2. `tuple` – кортеж 3. `range` – диапазон 5) Строки (Text Sequence Type) 1. `str`

6) Бинарные списки (Binary Sequence Types) 1. `bytes` – байты 2. `bytearray` – массивы байт 3. `memoryview` – специальные объекты для доступа к внутренним данным объекта через `protocol buffer`

7) Множества (Set Types) 1. set – множество 2. frozenset – неизменяемое множество

8) Словари (Mapping Types) 1. dict – словарь

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

Для того, чтобы объявить и сразу инициализировать переменную необходимо написать её имя, потом поставить знак равенства и значение, с которым эта переменная будет создана.

Например: `b = 5`

Целочисленное значение 5 в рамках языка Python по сути своей является объектом. Объект, в данном случае – это абстракция для представления данных, данные – это числа, списки, строки и т.п. При этом, под данными следует понимать как непосредственно сами объекты, так и отношения между ними (об этом чуть позже). Каждый объект имеет три атрибута – это идентификатор, значение и тип. Идентификатор – это уникальный признак объекта, позволяющий отличать объекты друг от друга, а значение – непосредственно информация, хранящаяся в памяти, которой управляет интерпретатор. При инициализации переменной, на уровне интерпретатора, происходит следующее: создается целочисленный объект 5 (можно представить, что в этот момент создается ячейка и 5 кладется в эту ячейку); данный объект имеет некоторый идентификатор, значение: 5, и тип: целое число; посредством оператора “=” создается ссылка между переменной `b` и целочисленным объектом 5 (переменная `b` ссылается на объект 5).

10. Как получить список ключевых слов в Python?

Список ключевых слов можно получить непосредственно в программе, для этого нужно подключить модуль `keyword` и воспользоваться командой `keyword.kwlist`.

11. Каково назначение функций `id()` и `type()`?

Для того, чтобы посмотреть на объект с каким идентификатором ссылается данная переменная, можно использовать функцию `id()`. `type()` используется для определения типа переменной.

12. Что такое изменяемые и неизменяемые типы в Python.

К неизменяемым (immutable) типам относятся: целые числа (int), числа с плавающей точкой (float), комплексные числа (complex), логические переменные (bool), кортежи (tuple), строки (str) и неизменяемые множества (frozen set). Неизменяемость типа данных означает, что созданный объект больше не изменяется. Например, если мы объявим переменную $k = 15$, то будет создан объект со значением 15, типа int и идентификатором, который можно узнать с помощью функции id().

К изменяемым (mutable) типам относятся: списки (list), множества (set), словари (dict). Если тип данных изменяемый, то можно менять значение объекта.

13. Чем отличаются операции деления и целочисленного деления?

Если в качестве операндов некоторого арифметического выражения используются только целые числа, то результат тоже будет целое число. Исключением является операция деления, результатом которой является вещественное число. При совместном использовании целочисленных и вещественных переменных, результат будет вещественным.

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

Для создания комплексного числа можно использовать функцию `complex(a, b)`, в которую, в качестве первого аргумента, передается действительная часть, в качестве второго – мнимая. Либо записать число в виде $a + bj$.

15. 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 предварительно нужно импортировать.

`math.ceil(x)` Возвращает ближайшее целое число большее, чем x .

`math.fabs(x)` Возвращает абсолютное значение числа.

`math.factorial(x)` Вычисляет факториал x .

`math.floor(x)` Возвращает ближайшее целое число меньшее, чем x .

`math.exp(x)` Вычисляет $e^{**}x$.

`math.log2(x)` Логарифм по основанию 2.

`math.log10(x)` Логарифм по основанию 10.

`math.log(x[, base])` По умолчанию вычисляет логарифм по основанию e , дополнительно можно указать основание логарифма.

`math.pow(x, y)` Вычисляет значение x в степени y .

`math.sqrt(x)` Корень квадратный от x .

`math.cos(x)` Косинус от x .

`math.sin(x)` Синус от x .

`math.tan(x)` Тангенс от x .

`math.acos(x)` Арккосинус от x .

`math.asin(x)` Арксинус от x .

`math.atan(x)` Арктангенс от x .

`math.pi` Число π .

`math.e` Число e .

Модуль `cmath` – предоставляет функции для работы с комплексными числами.

`cmath.polar(x)` - преобразование к полярным координатам. Возвращает пару (r, ϕ) .

`cmath.rect(r, phi)` - преобразование из полярных координат.

`cmath.exp(x)` - e^x .

`cmath.log(x[, base])` - логарифм x по основанию $base$. Если $base$ не указан, возвращается натуральный логарифм.

`cmath.log10(x)` - десятичный логарифм.

`cmath.sqrt(x)` - квадратный корень из x .

`cmath.acos(x)` - арккосинус x .

`cmath.asin(x)` - арксинус x .

`cmath.atan(x)` - арктангенс x .

`cmath.cos(x)` - косинус x .

`cmath.sin(x)` - синус x .

`cmath.tan(x)` - тангенс x .

`cmath.acosh(x)` - гиперболический арккосинус x .

`cmath.asinh(x)` - гиперболический арксинус x .

`cmath.atanh(x)` - гиперболический арктангенс x .

`cmath.cosh(x)` - гиперболический косинус x .

`cmath.sinh(x)` - гиперболический синус x .

`cmath.tanh(x)` - гиперболический тангенс x .

`cmath.isfinite(x)` - True, если действительная и мнимая части конечны.

`cmath.isinf(x)` - True, если либо действительная, либо мнимая часть бесконечна.

`cmath.isnan(x)` - True, если либо действительная, либо мнимая часть NaN.

`cmath.pi` - π .

`cmath.e` - e .

16. Каково назначение именованных параметров `sep` и `end` в функции `print()`?

Через параметр `sep` можно указать отличный от пробела разделитель строк.

Параметр `end` позволяет указывать, что делать, после вывода строки. По умолчанию происходит переход на новую строку. Однако это действие можно отменить, указав любой другой символ или строку.

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

Функция `format()` в Python используется для создания форматированной строки из строки шаблона и предоставленных значений.

В Python существует 5 способов форматирования строк:

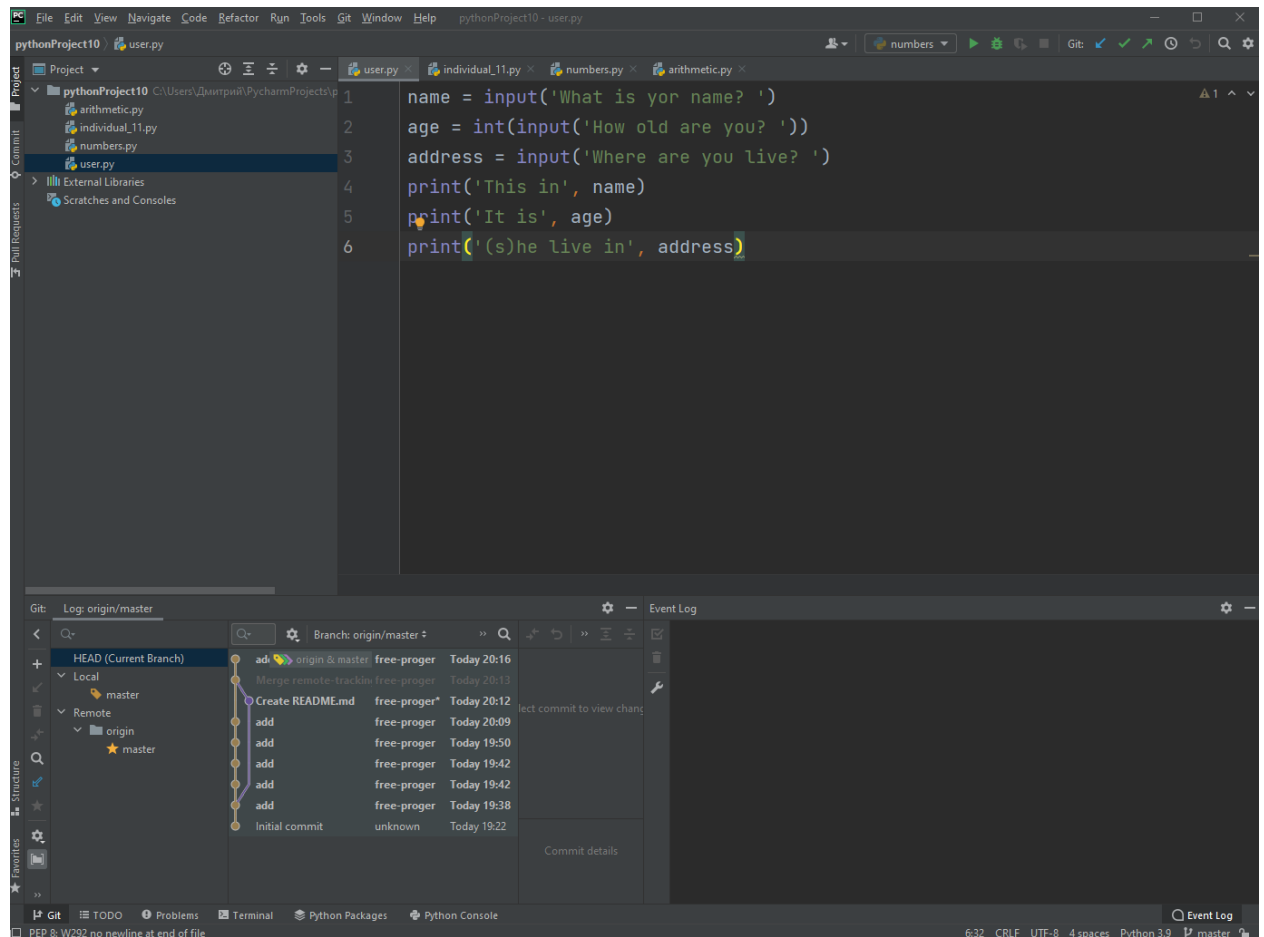
- Конкатенация;
- Оператора `%`;
- Модуль `Template`;
- Метод `format()`;
- f-строки.

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

`a = int(input())` – целочисленное

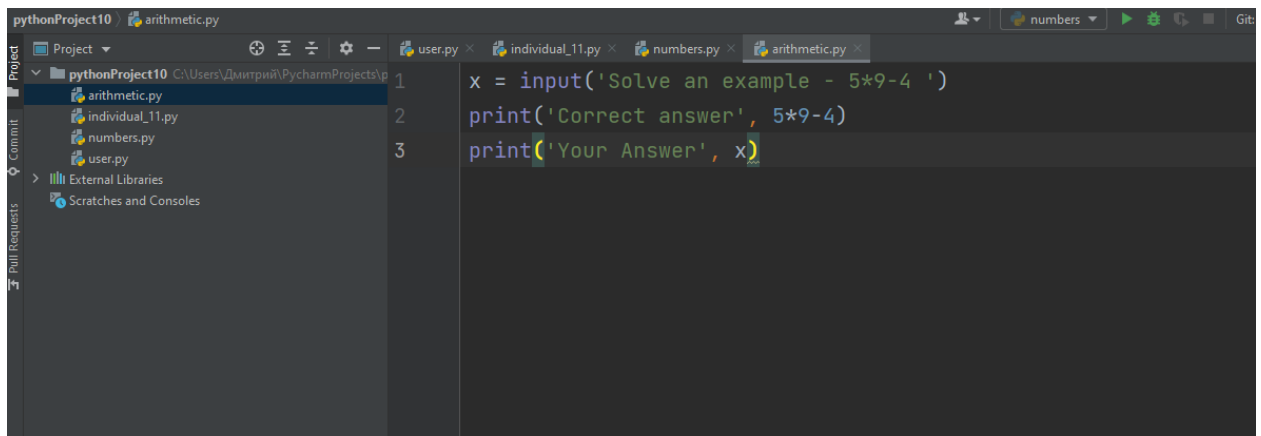
`a = float(input())` - вещественное

Напишите программу (файл `user.py`), которая запрашивала бы у пользователя:



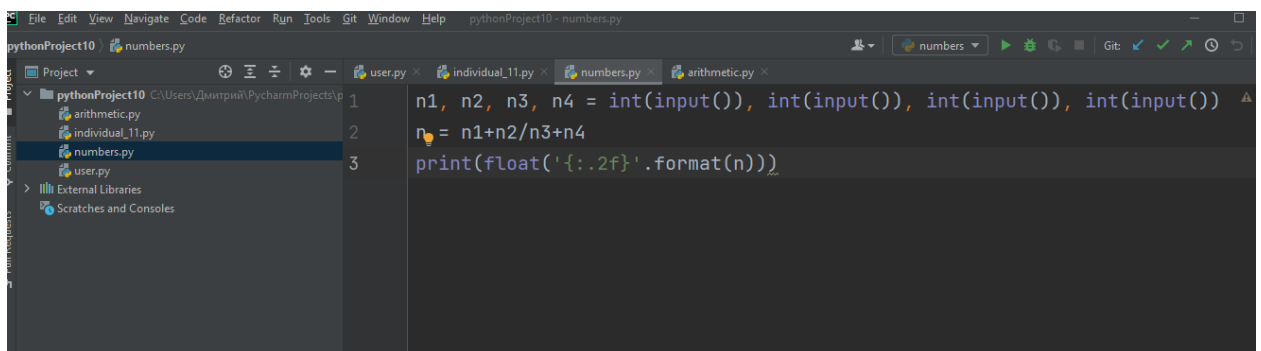
Напишите программу (файл `arithmetic.py`), которая предлагала бы пользователю решить пример $4 * 100 - 54$. Потом выводила бы на экран правильный ответ и ответ пользователя.

Подумайте, нужно ли здесь преобразовывать строку в число



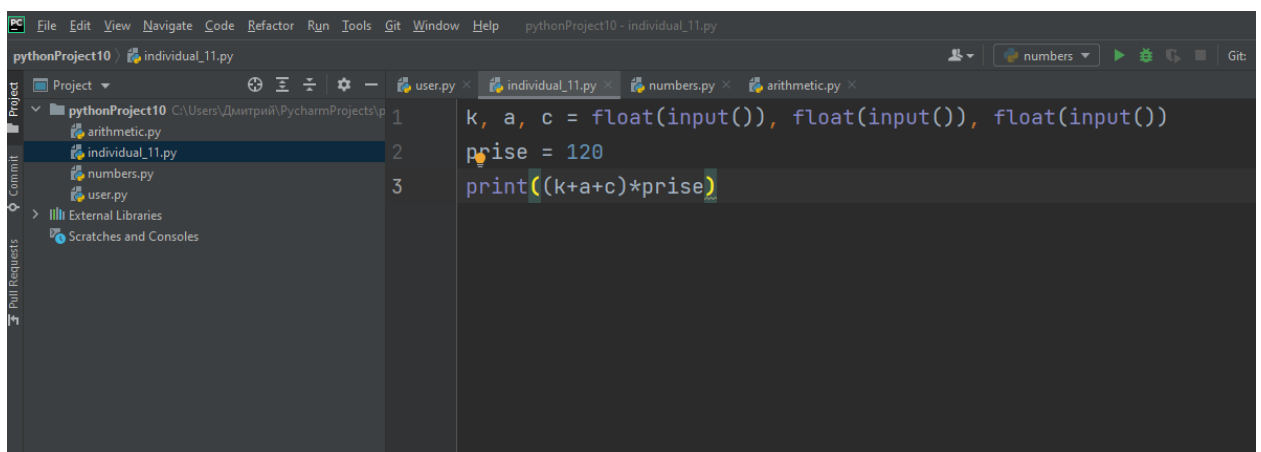
```
pythonProject10 > arithmetic.py
1 x = input('Solve an example - 5*9-4 ')
2 print('Correct answer', 5*9-4)
3 print('Your Answer', x)
```

Запросите у пользователя четыре числа (файл numbers.py). Отдельно сложите первые два и отдельно вторые два. Разделите первую сумму на вторую. Выведите результат на экран так, чтобы ответ содержал две цифры после запятой.



```
pythonProject10 > numbers.py
1 n1, n2, n3, n4 = int(input()), int(input()), int(input()), int(input())
2 n = n1+n2/n3+n4
3 print(float('{:.2f}'.format(n)))
```

Напишите программу (файл individual.py) для решения индивидуального задания. Вариант индивидуального задания уточните у преподавателя.



```
pythonProject10 > individual_11.py
1 k, a, c = float(input()), float(input()), float(input())
2 prise = 120
3 print((k+a+c)*prise)
```

Выполните коммит файлов user.py, arithmetic.py, numbers.py и individual.py в репозиторий git в ветку для разработки.

master 1 branch 0 tags Go to file Add file Code

free-proger add 9c41c3e 1 hour ago 9 commits

.idea	Initial commit	2 hours ago
arithmetic.py	add	2 hours ago
individual_11.py	add	1 hour ago
numbers.py	add	1 hour ago
user.py	add	2 hours ago

Help people interested in this repository understand your project by adding a README. Add a README

About

<https://github.com/free-proger/Python.git>

Activity

0 stars

1 watching

0 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published