

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 «Основы программной инженерии»

Выполнил:
Баратов Семен Григорьевич
2 курс, группа ПИЖ-б-о-22-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

(подпись)

Преподаватель:
Воронкин Р.А., канд. тех. наук, доцент,
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.

Результаты выполнения

1. Создали репозиторий с лицензией MIT, добавили в .gitignore необходимые правила для работы с IDE PyCharm, клонировали репозиторий, организовали репозиторий в соответствии с моделью git-flow.

```
Last login: Tue Oct 24 20:29:02 on ttys000
[itssyoma@MacBook-Air-Sema Основы программной инженерии % git clone https://github.com/itssyoma/megarepo_21.git
Cloning into 'megarepo_21'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
[itssyoma@MacBook-Air-Sema Основы программной инженерии % cd megarepo_21
[itssyoma@MacBook-Air-Sema megarepo_21 % git checkout -b develop
Switched to a new branch 'develop'
[itssyoma@MacBook-Air-Sema megarepo_21 % git branch release
[itssyoma@MacBook-Air-Sema megarepo_21 % git branch develop
fatal: a branch named 'develop' already exists
[itssyoma@MacBook-Air-Sema megarepo_21 % git branch hotfix
[itssyoma@MacBook-Air-Sema megarepo_21 % git branch feature
[itssyoma@MacBook-Air-Sema megarepo_21 %
```

Рисунок 1 – Работа с репозиторием в командной строке.

2. Написали программу user.py.

```
name = input("What is your name? ")
age = input("How old are you? ")
location = input("Where are you live? ")

print(f"This is {name}\nIt is {age}\n(S)he lives in {location}")
```

Рисунок 2 – Код программы.

```
What is your name? Syoma
How old are you? 19
Where are you live? Stavropol
This is Syoma
It is 19
(S)he lives in Stavropol
```

Рисунок 3 – Вывод программы.

3. Написали программу arithmetic.py.

```
user_result = input("Count this: 4 * 100 - 54. What's your answer? ")
result = 4 * 100 - 54
print(f"Your result is: {user_result}\nReal result is: {result}")
```

Рисунок 4 – Код программы.

```
Count this: 4 * 100 - 54. What's your answer? 9394382
Your result is: 9394382
Real result is: 346
```

Рисунок 5– Вывод программы.

4. Написали программу numbers.py.

```
a, b, c, d = map(float, input("Enter 4 numbers: ").split())
res_1 = a + b
res_2 = c + d
res = res_1 / res_2
print("{:.2f}".format(res))
```

Рисунок 6 – Код программы.

```
Enter 4 numbers: 5 7 2 5
1.71
```

Рисунок 7 – Вывод программы.

5. Написали программу по индивидуальному заданию. Даны два числа. Найти среднее арифметическое и среднее геометрическое их модулей.

```
a, b = map(float, input("Enter 2 numbers: ").split())
mean_a = (abs(a) + abs(b)) / 2
mean_g = (abs(a) * abs(b))**0.5
print(f"Arithmetic mean: {mean_a}\nGeometric mean: {mean_g}")
```

Рисунок 8 – Код программы.

```
Enter 2 numbers: -6 4
Arithmetic mean: 5.0
Geometric mean: 4.898979485566356
```

Рисунок 9 – Вывод программы.

6. Написали программу повышенной сложности. Даны цифры двух целых чисел: двузначного a_2 a_1 и однозначного b , где a_1 – число единиц, a_2 – число десятков. Получить цифры числа, равного сумме заданных чисел (известно, что это число двузначное). Слагаемое – двузначное число и число-результат не определять; условный оператор не использовать.

```
a2, a1, b = map(int, input("Enter a2, a1, b: ").split())
c2 = a2 + (a1 + b) // 10
c1 = (a1 + b) % 10
print(f"Number of tens: {c2}\nNumber of ones: {c1}")
```

Рисунок 10 – Код программы.

```
Enter a2, a1, b: 5 4 7
Number of tens: 6
Number of ones: 1
```

Рисунок 11 – Вывод программы.

7. Сделали коммит файлов python.

```
itssyoma@MacBook-Air-Sema megarepo_21 % git status
On branch develop
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    arithmetic.py
    hard.py
    individual.py
    numbers.py
    user.py

nothing added to commit but untracked files present (use "git add" to track)
itssyoma@MacBook-Air-Sema megarepo_21 % git add .
itssyoma@MacBook-Air-Sema megarepo_21 % git commit -m "Upload python files"
[develop bf2cfa3] Upload python files
5 files changed, 21 insertions(+)
create mode 100644 arithmetic.py
create mode 100644 hard.py
create mode 100644 individual.py
create mode 100644 numbers.py
create mode 100644 user.py
```

Рисунок 12 – Работа с репозиторием в командной строке.

8. Слили ветку develop с веткой main.

```
itssyoma@MacBook-Air-Sema megarepo_21 % git merge develop
Merge made by the 'ort' strategy.
 .gitignore | 4 +
 arithmetic.py | 3 +
 hard.py | 4 +
 individual.py | 4 +
 numbers.py | 5 +
 user.py | 5 +
```

Рисунок 13 – Работа с репозиторием в командной строке.

9. Отправили все изменения на удаленный репозиторий.

Ответы на контрольные вопросы

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

- Скачать установочный файл Python с официального сайта python.org.
- Запустить установочный файл и следовать инструкциям мастера установки.
- Выбрать опцию "Add Python to PATH" (добавить Python в переменную окружения PATH) для удобства использования.
- Нажать кнопку "Install Now" и дождаться завершения установки.
- Проверить успешность установки, выполнив команду `python --version` в командной строке.

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

- Пакет Anaconda включает в себя не только сам язык Python, но и множество научных библиотек и инструментов, таких как NumPy, Pandas, Matplotlib и другие.
- Anaconda предоставляет собственное окружение (conda environment) для управления пакетами и их зависимостями, что делает установку и использование научных библиотек более удобным.
- В отличие от стандартного пакета Python, Anaconda уже содержит множество научных библиотек, что позволяет сразу приступить к работе с анализом данных и машинным обучением.

3. Как осуществить проверку работоспособности пакета Anaconda?

- Запустить командную строку или терминал.
- Ввести команду `conda --version` и нажать Enter.
- Если выводится версия conda, значит Anaconda успешно установлена и готова к использованию.

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

- Открыть настройки PyCharm (File -> Settings).
- В разделе "Project Interpreter" выбрать нужный интерпретатор Python из списка или добавить новый интерпретатор.
- Нажать кнопку "Apply" или "OK" для сохранения изменений.

5. Как осуществить запуск программы с помощью IDE PyCharm?

- Открыть файл программы в PyCharm.
- Нажать кнопку "Run" или использовать сочетание клавиш Ctrl+Shift+F10.
- Программа будет запущена и ее вывод будет отображен в окне консоли PyCharm.

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

- Интерактивный режим позволяет вводить и выполнять команды Python непосредственно в командной строке или интерпретаторе Python. Результаты вычислений выводятся сразу же.
- Пакетный режим предполагает написание программ на языке Python в текстовых файлах, которые затем выполняются интерпретатором Python. Результаты выводятся после выполнения всей программы.

7. Почему язык программирования Python называется языком динамической типизации?

Язык программирования Python называется языком динамической типизации, потому что типы переменных определяются автоматически во время выполнения программы. Это означает, что переменная может содержать значения разных типов в разные моменты времени, и тип переменной может быть изменен в процессе выполнения программы.

8. Какие существуют основные типы в языке программирования Python?

- int (целое число)
- float (вещественное число)

- str (строка)
- bool (логическое значение True или False)
- list (список)
- tuple (кортеж)
- dict (словарь)
- set (множество)

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

Объекты в памяти создаются при выполнении операции присваивания. При объявлении новой переменной выделяется память под объект нужного типа, и переменная ссылается на этот объект. В процессе работы программы объекты могут создаваться и удаляться по мере необходимости. Организация памяти и устройство объектов зависит от реализации интерпретатора Python.

10. Как получить список ключевых слов в Python?

Список ключевых слов в Python можно получить с помощью функции `keyword.kwlist` из модуля `keyword`. Например:

```
import keyword  
print(keyword.kwlist)
```

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

Функция `id()` возвращает уникальный идентификатор объекта, который может быть использован для сравнения объектов на идентичность. Функция `type()` возвращает тип объекта.

12. Что такое изменяемые и неизменяемые типы в Python.

В Python существуют изменяемые и неизменяемые типы данных. Изменяемые типы данных могут быть изменены после создания, например, списки и словари. Неизменяемые типы данных не могут быть изменены после создания, например, числа, строки и кортежи.

13. Чем отличаются операции деления и целочисленного деления?

Операция деления / возвращает вещественный результат, даже если оба операнда являются целыми числами. Операция целочисленного деления // возвращает результат целочисленного деления без остатка.

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

Для работы с комплексными числами в Python используется встроенный тип `complex`. Комплексные числа могут быть созданы с помощью функции `complex()` или записаны в виде `a + bj`, где `a` и `b` - вещественные числа.

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

Библиотека (модуль) `math` предоставляет функции для математических вычислений, такие как тригонометрические функции, логарифмы, степени и др. Модуль `cmath` предоставляет аналогичные функции для работы с комплексными числами.

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

Параметр `sep` в функции `print()` определяет разделитель между элементами, которые нужно вывести. Параметр `end` определяет символ, который будет добавлен в конец вывода. По умолчанию `sep` равен пробелу, а `end` равен символу новой строки.

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.

Метод `format()` используется для форматирования строк в Python. Он позволяет вставлять значения переменных в строку и задавать формат

вывода. Другие средства для форматирования строк в Python включают оператор % и f-строки.

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

Ввод целочисленной переменной можно осуществить с помощью функции `int(input())`, а ввод вещественной переменной - с помощью функции `float(input())`. Например:

```
x = int(input("Введите целое число: "))
```

```
y = float(input("Введите вещественное число: "))
```