

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 №4
дисциплины «Программирование на Python»

Выполнила:
Михеева Елена Александровна
2 курс, группа ИВТ-б-з-20-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной
техники и автоматизированных
систем», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А....

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

```

[MacBook-Pro-Elena:~ mikheeva$ git clone https://github.com/helendddd/Python_2.1.git
Клонирование в «Python_2.1»...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Получение объектов: 100% (5/5), готово.
MacBook-Pro-Elena:~ mikheeva$ cd Python_2.1
[MacBook-Pro-Elena:Python_2.1 mikheeva$ git init
[Переинициализирован существующий репозиторий Git в /Users/mikheeva/Python_2.1/.git/
MacBook-Pro-Elena:Python_2.1 mikheeva$ git add arithmerick
fatal: спецификатор пути «arithmerick» не соответствует ни одному файлу
[MacBook-Pro-Elena:Python_2.1 mikheeva$ git add Programm
MacBook-Pro-Elena:Python_2.1 mikheeva$ git status
[Текущая ветка: main
[Эта ветка соответствует «origin/main».

Изменения, которые будут включены в коммит:
(используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    новый файл:    Programm/arithmetic.py
    новый файл:    Programm/hard_individual.py
    новый файл:    Programm/individual.py
    новый файл:    Programm/numbers.py
    новый файл:    Programm/user.py

Неотслеживаемые файлы:
(используйте «git add <файл>...», чтобы добавить в то, что будет включено в коммит)
    .DS_Store

[MacBook-Pro-Elena:Python_2.1 mikheeva$ git commit -m "add all programs"
[main f363861] add all programs
5 files changed, 43 insertions(+)
create mode 100644 Programm/arithmetic.py
create mode 100644 Programm/hard_individual.py
create mode 100644 Programm/individual.py
create mode 100644 Programm/numbers.py
create mode 100644 Programm/user.py
[MacBook-Pro-Elena:Python_2.1 mikheeva$ git checkout -b develop_h
Переключились на новую ветку «develop_h»
MacBook-Pro-Elena:Python_2.1 mikheeva$ git checkout main
Переключились на ветку «main»
Ваша ветка опережает «origin/main» на 1 коммит.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
MacBook-Pro-Elena:Python_2.1 mikheeva$ git merge develop_h
Обновление f363861..c42691e
Fast-forward
 .DS_Store          | Bin 0 -> 6148 bytes
 Programm/arithmetic.py | 3 +++
 Programm/hard_individual.py | 5 ++++-
 Programm/individual.py | 3 +++
 Programm/numbers.py | 3 +++
 Programm/user.py | 3 +++
6 files changed, 16 insertions(+), 1 deletion(-)
create mode 100644 .DS_Store
MacBook-Pro-Elena:Python_2.1 mikheeva$

```

```
[MacBook-Pro-Elena:Python_2.1 mikheeva$ git status
Текущая ветка: develop_h
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    новый файл:      .DS_Store
    изменено:       Programm/arithmetic.py

[MacBook-Pro-Elena:Python_2.1 mikheeva$ git add .
[MacBook-Pro-Elena:Python_2.1 mikheeva$ git status
Текущая ветка: develop_h
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    новый файл:      .DS_Store
    изменено:       Programm/arithmetic.py
    изменено:       Programm/hard_individual.py
    изменено:       Programm/individual.py
    изменено:       Programm/numbers.py
    изменено:       Programm/user.py

[MacBook-Pro-Elena:Python_2.1 mikheeva$ git commit -m "Add shebang for all programs"
[develop_h eb7ee6e] Add shebang for all programs
 6 files changed, 15 insertions(+)
 create mode 100644 .DS_Store
[MacBook-Pro-Elena:Python_2.1 mikheeva$ git status
Текущая ветка: develop_h
нечего коммитить, нет изменений в рабочем каталоге
[MacBook-Pro-Elena:Python_2.1 mikheeva$ add Programm/hard_individual.py
-bash: add: command not found
[MacBook-Pro-Elena:Python_2.1 mikheeva$ git add Programm/hard_individual.py
[MacBook-Pro-Elena:Python_2.1 mikheeva$ git status
Текущая ветка: develop_h
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    изменено:       Programm/hard_individual.py

[MacBook-Pro-Elena:Python_2.1 mikheeva$ git commit -m "Correct input"
[develop_h c42691e] Correct input
 1 file changed, 1 insertion(+), 1 deletion(-)
[MacBook-Pro-Elena:Python_2.1 mikheeva$
MacBook-Pro-Elena:Python_2.1 mikheeva$ █
```

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  print("Calculate...")
5  answer = int(input("4 * 100 - 54 = "))
6
7  if answer == 346:
8      print("Correct!")
9  else:
10     print("Wrong! Correct answer is 346!\nYour answer is ", answer)
11

```

Рисунок 1. Программа aithmetic.py

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  name = input("\nWhat is your name? ")
5  age = input("How old are you? ")
6  town = input("Where are you live? ")
7
8  print("\nThis is ", name)
9  print("It is ", age)
10 print("(S)he live in ", town)
11

```

Рисунок 2. Программа user.py

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  first_number = int(input("Enter 1st number... "))
5  second_number = int(input("Enter 2nd number... "))
6  third_number = int(input("Enter 3rd number... "))
7  fourth_number = int(input("Enter 4th number... "))
8
9  sum_1 = first_number + second_number
10 sum_2 = third_number + fourth_number
11
12 answer = sum_1 / sum_2
13
14 print("(1st + 2nd) / (3rd + 4th) = ", f"{answer:.2f}")
15

```

Рисунок 3. Программа numbers.py

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  from math import pi
5
6  R = float(input("Enter outer radius... "))
7  r = float(input("Enter inner radius... "))
8
9  print("Ring area is ", f"{pi * (R - r):.2f}")
10

```

Рисунок 4. Программа individual.py

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  # Вариант №6
5  # С начала суток часовая стрелка повернулась на y градусов
6  # (0 ≤ y < 360, – вещественное число).
7  # Определить число полных часов и число полных минут, прошедших с начала суток
8
9  y = float(input("Enter degrees... "))
10
11 if y < 0 or y > 360:
12     print("Incorrect data entry!")
13 else:
14     print("Hours passed: ", f"{y//15:.0f}")
15     print("Minutes passed: ", f"{y//0.25:.0f}")
16

```

Рисунок 5. Программа hard_individual.py

Ответы на вопросы:

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

На windows:

- Скачать последнюю версию Python для Windows на официальный сайт Python

- Запустить установщик.

- Открыть командную строку и ввести python --version или python -V.

Должна отобразиться версия Python.

На Linux:

- Открыть терминал и ввести sudo apt-get install -y python3 (для Ubuntu и подобных дистрибутивов).

– Ввести `python3 --version` или `python3 -V`. Должна отобразиться версия Python.

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

Anaconda — это дистрибутив Python, который включает в себя не только интерпретатор Python, но и множество научных библиотек, инструментов для анализа данных, визуализации и управления пакетами (например, `conda`). Anaconda предлагает собственное окружение, которое упрощает установку и управление зависимостями.

Anaconda предоставляет расширенный дистрибутив Python с предустановленными научными библиотеками и инструментами для анализа данных, также использует свой менеджер пакетов `conda`. В отличие от этого, официальный пакет Python включает базовый интерпретатор и требует использования стандартных инструментов управления пакетами, таких как `pip`. Anaconda ориентирована на научные вычисления и машинное обучение, тогда как пакет Python подходит для более общего программирования.

3. Как осуществить проверку работоспособности пакета Anaconda?

Необходимо перейти в папку `Scripts` и ввести в командной строке команду `«ipython notebook»`. В результате запустится веб-сервер и среда разработки в браузере. Далее нужно создать ноутбук для разработки, для этого нажать на кнопку `New` и в появившемся списке выбрать `Python`. В результате будет создана новая страница в браузере с ноутбуком. Вводим в первой ячейке команду `print("Hello, World!")` и нажимаем `Alt+Enter` на клавиатуре. Ниже ячейки должна появиться соответствующая надпись.

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

Необходимо открыть проект в PyCharm. Перейти в настройки проекта, выбрать `"Project: [name project]" -> "Python Interpreter"`. В настройках `"Python Interpreter"` выбрать `"Add..."` для добавления нового интерпретатора. Затем

выбрать существующий интерпретатор Python или добавить новый, указав путь к интерпретатору.

5. Как осуществить запуск программы с помощью IDE PyCharm?

Открыть файл с Python-кодом, убедиться, что правильно выбран интерпретатор, запустить программу.

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

Интерактивный режим: запускается командой `python` в терминале. Позволяет вводить команды построчно и видеть результат немедленно.

Пакетный режим: запускается скриптом. Весь код выполняется от начала до конца. Используется для запуска программ и скриптов

7. Почему язык программирования Python называется языком динамической типизации?

Потому-что в Python тип переменной определяется во время выполнения программы, а не на этапе компиляции. Переменные могут изменять свой тип в процессе выполнения.

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?

Объекты создаются динамически во время выполнения программы при помощи оператора =.

При инициализации переменной, на уровне интерпретатора, происходит следующее:

- создается целочисленный объект;
- данный объект имеет некоторый идентификатор, значение и тип;
- посредством оператора “=” создается ссылка между переменной и объектом (переменная ссылается на объект).

Переменные — это ссылки на объекты в памяти. Процесс объявления переменных: имя_переменной = значение.

10. Как получить список ключевых слов в Python?

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

keyword - Модуль, благодаря которому можно получить список всех ключевых слов (keyword.kwlist) а также проверить, является ли значение ключевым (keyword.iskeyword())