

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций
«Работа со списками в языке Python»

Отчет по лабораторной работе № 2.4

по дисциплине
«Программирование на Python»

Выполнил студент группы ИВТ-б-о-22-1

Сумин Никита Сергеевич.

« » _____ 2023г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____

(подпись)

Ставрополь 2023

Цель работы: приобретение навыков по работе со списками при написании программ с помощью языка программирования Python версии 3.x.

Порядок выполнения работы:

1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT.

2. Выполните клонирование созданного репозитория.

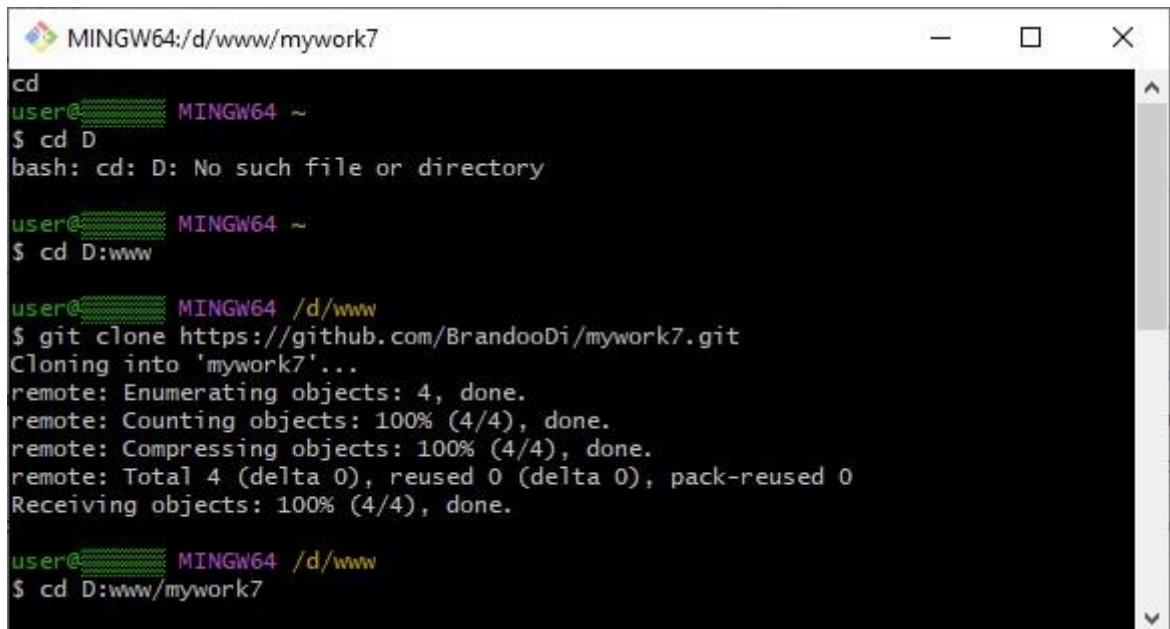
A screenshot of a terminal window titled 'MINGW64:/d/www/mywork7'. The terminal shows a series of commands and their outputs. The user starts by running 'cd', then '\$ cd D', which results in an error 'bash: cd: D: No such file or directory'. Next, the user runs '\$ cd D:www'. Then, the user runs '\$ git clone https://github.com/BrandooDi/mywork7.git'. The output shows the cloning process: 'Cloning into 'mywork7'...', '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', and 'Receiving objects: 100% (4/4), done.'. Finally, the user runs '\$ cd D:www/mywork7'.

Рисунок 1 - Клонирование репозитория

3. Организуйте свой репозиторий в соответствие с моделью ветвления git-flow.

```
MINGW64:/d/www/mywork7
$ git checkout -d develop
fatal: git checkout: --detach does not take a path argument 'develop'

user@MINGW64 /d/www/mywork7 (main)
$ git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]
How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/] Release branches? [release/] Hotfix branches? [hotfix/]
Support branches? [support/] Version tag prefix? []
```

Рисунок 2 - Ветвление по модели git-flow 4.

Создайте проект PyCharm в папке репозитория.

| Имя | Дата изменения | Тип | Размер |
|-----------------|------------------|--------------------|--------|
| .git | 13.09.2022 17:38 | Папка с файлами | |
| PyCharm Project | 13.09.2022 17:43 | Папка с файлами | |
| .gitignore | 13.09.2022 17:38 | Текстовый докум... | 2 КБ |
| LICENSE | 13.09.2022 17:38 | Файл | 2 КБ |

Рисунок 3 - Проект PyCharm

5. Проработайте примеры лабораторной работы. Создайте для каждого примера отдельный модуль языка Python. Зафиксируйте изменения в репозитории.

Пример 1. Ввести список A из 10 элементов, найти сумму элементов, меньших по модулю 5, и вывести ее на экран.

```
"D:\www\mywork7\PyCharm Project\venv\Scripts\python.exe" "D:/www/mywork7/PyCharm Project/pr1.py"  
0 1 2 3 4 5 6 7 8 9  
10  
  
Process finished with exit code 0
```

Рисунок 4 – Результат выполнения программы

Пример 2. Написать программу, которая для целочисленного списка определяет, сколько положительных элементов располагается между его максимальным и минимальным элементами.

```
"D:\www\mywork7\PyCharm Project\venv\Scripts\python.exe" "D:/www/mywork7/PyCharm Project/pr2.py"  
1 13 34 45 56 109  
4  
  
Process finished with exit code 0
```

Рисунок 5 - Результат выполнения программы

6. Выполните индивидуальные задания:

Задание 1. В списках U, D, V содержатся значения утренней, дневной и вечерней температуры

соответственно за каждый день недели. Подсчитать среднее значение дневной температуры за каждый день.

```
"D:\www\mywork7\PyCharm Project\venv\Scripts\python.exe" "D:/www/mywork7/PyCharm Project/zd1.py"
Введите значения U: 23 24 25 26 26 26 26
Введите значения D: 32 33 43 54 24 24 32
Введите значения B: 32 25 54 34 27 26 20
Среднее значение температуры по дням недели: [29.0, 20.666666666666668, 40.666666666666664, 38.0, 25.666666666666668, 25.333333333333332, 26.666666666666668]
Process finished with exit code 0
|
```

Рисунок 6 - Результат выполнения программы

Задание 2 . В списке, состоящем из вещественных элементов, вычислить:

1. количество положительных элементов списка;
2. сумму элементов списка, расположенных после последнего элемента, равного нулю.

```
"D:\www\mywork7\PyCharm Project\venv\Scripts\python.exe" "D:/www/mywork7/PyCharm Project/zd2.py"
Введите элементы списка: 2 3 10 34 54 105
Введите минимальное значение диапазона (A): 2
Введите максимальное значение диапазона (B): 105
Количество чисел в диапазоне от A до B = 4
Сумма чисел после максимального элемента, введённого списка 0
[105.0]
Process finished with exit code 0
|
```

Рисунок 7 - Результат выполнения программы

Контрольные вопросы:

1. Что такое списки в языке Python?

Список (list) – это структура данных для хранения объектов различных типов.

2. Как осуществляется создание списка в Python?

Для создания списка нужно заключить элементы в квадратные скобки:
`my_list = [1, 2, 3, 4, 5]`

Список может выглядеть так: `my_list = ['один', 'два', 'три', 'четыре', 'пять']`.

Можно смешивать типы содержимого: `my_list = ['один', 10, 2.25, [5, 15], 'пять']`.

3. Как организовано хранение списков в оперативной памяти?

Список является изменяемым типом данных. При его создании в памяти резервируется область, которую можно условно назвать некоторым “контейнером”, в котором хранятся ссылки на другие элементы данных в памяти. В отличие от таких типов данных как число или строка, содержимое “контейнера” списка можно менять.

4. Каким образом можно перебрать все элементы списка?

Перебор элементов списка состоит в том, что мы в цикле просматриваем все элементы этого списка.

Читать элементы списка можно с помощью следующего цикла:

```
my_list = ['один', 'два', 'три', 'четыре', 'пять']  
for elem in my_list: print(elem)
```

5. Какие существуют арифметические операции со списками?

Для объединения списков можно использовать оператор сложения (+).
Список можно повторить с помощью оператора умножения (*).

6. Как проверить есть ли элемент в списке?

Для того, чтобы проверить, есть ли заданный элемент в списке Python необходимо использовать оператор in.

7. Как определить число вхождений заданного элемента в списке?

Метод count можно использовать для определения числа сколько раз данный элемент встречается в списке.

8. Как осуществляется добавление (вставка) элемента в список?

Метод append можно использовать для добавления элемента в список.

Метод `insert` можно использовать, чтобы вставить элемент в список.

9. Как выполнить сортировку списка?

Для сортировки списка нужно использовать метод `sort()`, в порядке возрастания будет(`list1.sort()`). Для сортировки списка в порядке убывания необходимо вызвать метод `sort` с аргументом `reverse()`=`True`(`list1.reverse()`).

10. Как удалить один или несколько элементов из списка?

Удалить элемент можно, написав его индекс в методе `pop`.

11. Что такое списковое включение и как с его помощью осуществлять обработку списков?

List Comprehensions чаще всего на русский язык переводят как абстракция списков или списковое включение, является частью синтаксиса языка, которая предоставляет простой способ построения списков.

12. Как осуществляется доступ к элементам списков с помощью срезов? `list[::]`

13. Какие существуют функции агрегации для работы со списками?

Для работы со списками Python предоставляет следующие функции:

- `len(L)` - получить число элементов в списке `L` .
- `min(L)` - получить минимальный элемент списка `L` .
- `max(L)` - получить максимальный элемент списка `L` .
- `sum(L)` - получить сумму элементов списка `L` , если

список `L` содержит только числовые значения

14. Как создать копию списка?

Для создания копии списка необходимо использовать либо метод `copy`, либо использовать оператор среза.

15. Самостоятельно изучите функцию sorted языка Python. В чем ее отличие от метода sort списков?

Функция `sorted()` в Python возвращает отсортированный список из элементов в итерируемом объекте. `list.sort()` на 13% быстрее, чем `sorted()`.

Отличие заключается в том, что метод `list.sort()` определён только для списков, в то время как `sorted()` работает со всеми итерируемыми объектами.

Вывод: приобрел навыки по работе со списками при написании программ с помощью языка программирования Python версии 3.x.