

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования «СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Институт цифрового развития

ОТЧЁТ

по лабораторной работе №2.4

Дисциплина: «Основы кроссплатформенного программирования»

Тема: «Работа со списками в языке Python»

Выполнил: студент 1 курса,
группы ИВТ-б-о-21-1
Богдан Александр Анатольевич

Ставрополь 2022

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

Практическая часть:

Создал репозиторий на GitHub

```
D:\WAR\gi>git clone https://github.com/Alekbs/Lab7
Cloning into 'Lab7'...
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.
```

Рисунок 1. Клонирование репозитория

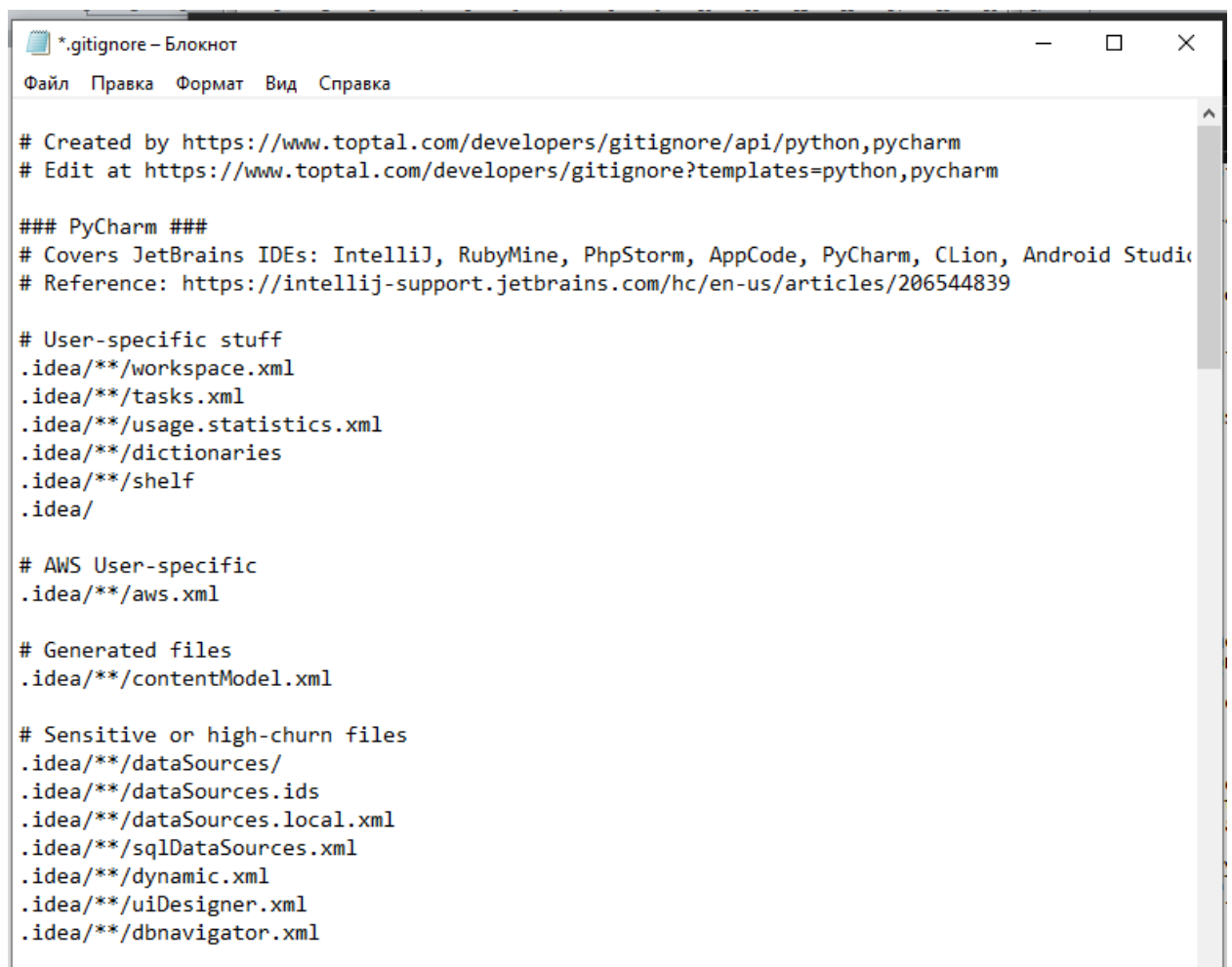


Рисунок 2. Изменение файла .gitignore

```

D:\WAR\gi\Lab7>git flow init

Which branch should be used for bringing forth production releases?
  - develop
  - main
Branch name for production releases: [main] main

Which branch should be used for integration of the "next release"?
  - develop
Branch name for "next release" development: [develop] develop

How to name your supporting branch prefixes?
Feature branches? [feature/] feat
Bugfix branches? [bugfix/] bug
Release branches? [release/] release
Hotfix branches? [hotfix/] hot
Support branches? [support/] support
Version tag prefix? [] pre
Hooks and filters directory? [D:/WAR/gi/Lab7/.git/hooks] hooks

```

Рисунок 3. Организация репозитория в соответствии с git-flow

Создак проект PyCharm в папке репозитория.

The screenshot shows the PyCharm IDE interface. The left sidebar displays the project structure for 'D:\WAR\gi\Lab7\Project', including a 'venv' directory and files 'main.py' and 'pri1.py'. The main editor window shows the code for 'pri1.py', which is a Python script that takes a list of numbers as input and prints the sum of those numbers. The code includes comments in Russian. The bottom console window shows the command to run the script: 'D:\WAR\gi\Lab7\Project\venv\Scripts\python.exe D:\WAR\gi\Lab7\Project\pri1.py', and the output: '1 0 1 0 1 0 1 0 1 0'.

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  import sys
4
5  if __name__ == '__main__':
6      # Ввести список одной строкой.
7      A = list(map(int, input().split()))
8      # Проверить количество элементов списка.
9      if len(A) != 10:
10         print("Неверный размер списка", file=sys.stderr)
11         exit(1)
12     # Найти искомую сумму.
13     s = sum([a for a in A if abs(a) < 5])
14     print(s)

```

```

D:\WAR\gi\Lab7\Project\venv\Scripts\python.exe D:\WAR\gi\Lab7\Project\pri1.py
1 0 1 0 1 0 1 0 1 0

```

Рисунок 4. Пример 1

```
1  #!/usr/bin/env python3
2  #- coding: utf-8 -*-
3  import sys
4
5  if __name__ == '__main__':
6      # Ввести список одной строкой.
7      a = list(map(int, input().split()))
8      # Если список пуст, завершить программу.
9      if not a:
10         print("Заданный список пуст", file=sys.stderr)
11         exit(1)
12      # Определить индексы минимального и максимального элементов.
13      a_min = a_max = a[0]
14      i_min = i_max = 0
15      for i, item in enumerate(a):
16         if item < a_min:
17             i_min, a_min = i, item
18
19         if item >= a_max:
20             i_max, a_max = i, item
21      # Проверить индексы и обменять их местами.
22      if i_min > i_max:
23         i_min, i_max = i_max, i_min
24      # Посчитать количество положительных элементов.
25      count = 0
26      for item in a[i_min+1:i_max]:
27         if item > 0:
28             count += 1
29      print(count)
```

if __name__ == '__main__' > for item in a[i_min+1:i_max] > if item > 0

in: pri2 ×

D:\WAR\gi\Lab7\Project\venv\Scripts\python.exe D:\WAR\gi\Lab7\Project\pri2.py

-44 1 2 3 -3 44

3

Рисунок 5. Пример 2

```
6 # Ввести список одной строкой.  
7 A = list(map(int, input().split()))  
8 # Проверить количество элементов списка.  
9 if len(A) != 10:  
10     print("Неверный размер списка", file=sys.stderr)  
11     exit(1)  
12 s = 1  
13  
14 for item in A:  
15     if item > 0:  
16         s *= item  
17 s1 = 0  
18 for i in A:  
19     if i == 1:  
20         s1 += 1  
21  
22 if s == 1:  
23     if s1 == 0:  
24         print("0")  
25     else:  
26         print("1")  
27 else:  
28     print(s)  
  
if __name__ == '__main__':  
    for i in A:  
        if i == 1
```

Run: pri2 x ind1 x
D:\WAR\gi\Lab7\Project\venv\Scripts\python.exe D:/WAR/gi/Lab7/Project/ind1.py
-2 -1 -1 3 -2 -2 4 -1 -1 1
12

Рисунок 6. Индивидуальное задание 1 (Вариант 2)

```
1 #!/usr/bin/env python3  
2 # -*- coding: utf-8 -*-  
3 import sys  
4  
5 if __name__ == '__main__':  
6     # Ввести список одной строкой.  
7     A = list(map(int, input().split()))  
8  
9     # Сумма положительных  
10    sum = 0  
11    for item in A:  
12        if item > 0:  
13            sum += item  
14  
15    # Поиск минимального и максимального по модулю  
16    min_A = abs(A[1])  
17    max_A = abs(A[1])  
18    Min, Max = 0, 0  
  
if __name__ == '__main__':  
    if Max - Min == 1
```

Run: pri2 x ind2 x
D:\WAR\gi\Lab7\Project\venv\Scripts\python.exe D:/WAR/gi/Lab7/Project/ind2.py
2 3 -4 1 2
Sum = 8
Product = 0
Sorted list: [3, 2, 2, 1, -4]

Рисунок 7. Индивидуальное задание 2 (Вариант 2)

Ответы на 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.

3. Как организовано хранение списков в оперативной памяти?

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?

Можно развернуть порядок элементов в списке с помощью метода reverse.

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.

Если требуется, чтобы элемент отсутствовал в списке, необходимо использовать оператор not 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 `insert` можно использовать, чтобы вставить элемент в список.

Метод `append` можно использовать для добавления элемента в список.

9. Как выполнить сортировку списка?

Для сортировки списка нужно использовать метод `sort`.

10. Как удалить один или несколько элементов из списка?

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

Элемент можно удалить с помощью метода `remove`.

Оператор `del` можно использовать для тех же целей.

Можно удалить все элементы из списка с помощью метода `clear`.

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.

В языке Python есть две очень мощные функции для работы с коллекциями: `map` и `filter`. Они позволяют использовать функциональный стиль программирования, не прибегая к помощи циклов, для работы с такими типами как `list`, `tuple`, `set`, `dict` и т.п. Списковое включение позволяет обойтись без этих функций.

12. Как осуществляется доступ к элементам списков с помощью срезов?

Слайсы (срезы) являются очень мощной составляющей Python, которая позволяет быстро и лаконично решать задачи выборки элементов из списка. Выше уже был пример использования слайсов, здесь разберем более подробно работу с ними. Создадим список для экспериментов:

```
a = [i for i in range(10)].
```

Слайс задается тройкой чисел, разделенных запятой: start:stop:step. Start – позиция с которой нужно начать выборку, stop – конечная позиция, step – шаг. При этом необходимо помнить, что выборка не включает элемент определяемый stop.

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.

Для функций min и max элементы списка должны быть сравнимы между собой.

14. Как создать копию списка?

Поэтому 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 sort () очень похожа на sorted (), но в отличие от sorted она ничего не возвращает и не вносит изменений в исходную

последовательность. Более того, `sort()` является методом класса `list` и может использоваться только со списками.

Вывод: в результате выполнения работы были приобретены навыки по работе со списками при написании программ с помощью языка программирования Python версии 3.x.