

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 №2.4
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: Работа со списками в языке Python.

Цель: приобретение навыков по работе со списками при написании программ с помощью языка программирования 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. Проработали пример №1.

```
example1.py > ...
1  #!/usr/bin/env python3
2  #-*- coding: utf-8 -*-
3
4  import sys
5
6
7  if __name__ == '__main__':
8
9      # Ввести список одной строкой.
10     A = list(map(int, input().split()))
11
12     # Проверить количество элементов списка.
13     if len(A) != 10:
14         print("Неверный размер списка", file=sys.stderr)
15         exit(1)
16
17     # Найти искомую сумму.
18     s = sum([a for a in A if abs(a) < 5])
19
20     print(s)
21
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ

```
• itssyoma@MacBook-Air-Sema megarepo_24 % /usr/local/bin/python3 "/
ed/Учеба/Основы программной инженерии/megarepo_24/example1.py"
-3 5 3 1 2 4 7 9 1 20
8
• itssyoma@MacBook-Air-Sema megarepo_24 % /usr/local/bin/python3 "/
ed/Учеба/Основы программной инженерии/megarepo_24/example1.py"
1 2 3 4
Неверный размер списка
```

Рисунок 2 – Пример №1. Код и результат выполнения программы.

3. Проработали пример №2.

```
example2.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6
7  if __name__ == '__main__':
8      # Ввести список одной строкой.
9      a = list(map(int, input().split()))
10     # Если список пуст, завершить программу.
11     if not a:
12         print("Заданный список пуст", file=sys.stderr)
13         exit(1)
14
15     # Определить индексы минимального и максимального элементов.
16     a_min = a_max = a[0]
17     i_min = i_max = 0
18     for i, item in enumerate(a):
19         if item < a_min:
20             i_min, a_min = i, item
21         if item >= a_max:
22             i_max, a_max = i, item
23
24     # Проверить индексы и обменять их местами.
25     if i_min > i_max:
26         i_min, i_max = i_max, i_min
27
28     # Посчитать количество положительных элементов.
29     count = 0
30     for item in a[i_min+1:i_max]:
31         if item > 0:
32             count += 1
33
34     print(count)
35
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

```
itssyoma@MacBook-Air-Sema megarepo_24 % /usr/local/bin/python3 "/Users/its:
xample2.py"
1 2 3 4 5 6 7
5
itssyoma@MacBook-Air-Sema megarepo_24 % /usr/local/bin/python3 "/Users/its:
e2.py"
5 4 7 5 1 6 3 4 7 9
4
itssyoma@MacBook-Air-Sema megarepo_24 % /usr/local/bin/python3 "/Users/its:
e2.py"
Заданный список пуст
```

Рисунок 3 – Пример №2. Код и результат выполнения программы.

4. Выполнили индивидуальное задание №1 (вариант 1). Ввести список А из 10 элементов, найти наибольший элемент и переставить его с первым элементом. Преобразованный массив вывести.

```
task1.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6
7  if __name__ == "__main__":
8      A = list(map(int, input("Введите элементы массива через пробел: ").split()))
9      if not A:
10         print("Заданный список пуст", file=sys.stderr)
11         exit(1)
12
13         max_elem = A[0]
14         max_index = 0
15
16         for i, num in enumerate(A):
17             if num > max_elem:
18                 max_elem = num
19                 max_index = i
20
21         A[0], A[max_index] = A[max_index], A[0]
22
23         print("Преобразованный список: ", A)
24
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

```
itssyoma@MacBook-Air-Sema megarepo_24 % /usr/local/bin/python3 "/Users/itssyoma/Yandex.Dis
py"
Введите элементы массива через пробел: 1 2 3 4 5 6 7 10 8 9
Преобразованный список: [10, 2, 3, 4, 5, 6, 7, 1, 8, 9]
itssyoma@MacBook-Air-Sema megarepo_24 % /usr/local/bin/python3 "/Users/itssyoma/Yandex.Dis
py"
Введите элементы массива через пробел:
Заданный список пуст
itssyoma@MacBook-Air-Sema megarepo_24 % /usr/local/bin/python3 "/Users/itssyoma/Yandex.Dis
py"
Введите элементы массива через пробел: 1 2 3 4 5 6 10 7 8 9
Преобразованный список: [10, 2, 3, 4, 5, 6, 1, 7, 8, 9]
```

Рисунок 4 – Задание №1. Решение с помощью циклов.

Код и результат выполнения программы.

```
task1_lc.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6
7  if __name__ == "__main__":
8      A = list(map(int, input("Введите элементы массива через пробел: ").split()))
9      if not A:
10         print("Заданный список пуст", file=sys.stderr)
11         exit(1)
12
13     res = [max(A) if i == 0 else A[0] if num == max(A) else num for i, num in enumerate(A)]
14     print("Преобразованный список: ", res)
15
16
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ **ТЕРМИНАЛ** ПОРТЫ

itssyoma@MacBook-Air-Sema megarepo_24 % /usr/local/bin/python3 "/Users/itssyoma/Yandex.Disk.localiz ed/Учеба/Основы программной инженерии/megarepo_24/task1_lc.py"
Введите элементы массива через пробел: 1 2 3 4 5 6 10 7 8 9
Преобразованный список: [10, 2, 3, 4, 5, 6, 1, 7, 8, 9]
itssyoma@MacBook-Air-Sema megarepo_24 % /usr/local/bin/python3 "/Users/itssyoma/Yandex.Disk.localiz ed/Учеба/Основы программной инженерии/megarepo_24/task1_lc.py"
Введите элементы массива через пробел:
Заданный список пуст

Рисунок 5 – Задание №1. Решение с помощью спискового включения.

Код и результат выполнения программы.

5. Выполнили индивидуальное задание №2 (вариант 1). В списке, состоящем из вещественных элементов, вычислить: сумму отрицательных элементов списка, произведение элементов списка, расположенных между максимальным и минимальным элементами. Упорядочить элементы списка по возрастанию.

```
task2.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6
7  if __name__ == "__main__":
8      A = list(map(float, input("Введите элементы массива через пробел: ").split()))
9      if not A:
10         print("Заданный список пуст", file=sys.stderr)
11         exit(1)
12
13     sum_negative = 0
14     min_index = A.index(min(A))
15     max_index = A.index(max(A))
16
17     if min_index < max_index:
18         rng = [i for i in range(min_index+1, max_index)]
19     else:
20         rng = [i for i in range(max_index+1, min_index)]
21
22     product_between = 1
23     for i, num in enumerate(A):
24         if num < 0:
25             sum_negative += A[i]
26         if i in rng:
27             product_between *= num
28
29     A.sort()
30
31     print("Сумма отрицательных элементов: ", sum_negative)
32     print("Произведение элементов между максимальным и минимальным: ", product_between)
33     print("Упорядоченный список: ", A)
34
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ **ТЕРМИНАЛ** ПОРТЫ

ed/Учеба/Основы программной инженерии/megarepo_24/task2.py"
Введите элементы массива через пробел: 2 3 4 1 2 3 4 10 6 -1 -5 -7
Сумма отрицательных элементов: -13.0
Произведение элементов между максимальным и минимальным: 30.0
Упорядоченный список: [-7.0, -5.0, -1.0, 1.0, 2.0, 2.0, 3.0, 3.0, 4.0, 4.0, 6.0, 10.0]
itssyoma@MacBook-Air-Sema megarepo_24 % /usr/local/bin/python3 "/Users/itssyoma/Yandex.Disk.localiz ed/Учеба/Основы программной инженерии/megarepo_24/task2.py"
Введите элементы массива через пробел:
Заданный список пуст

Рисунок 6 – Задание №2. Код и результат выполнения программы.

Ответы на контрольные вопросы

1. Что такое списки в языке Python?

Список в языке Python – это упорядоченная коллекция элементов, которая может содержать объекты разных типов.

2. Как осуществляется создание списка в Python?

Для создания списка в Python используется квадратные скобки [], внутри которых перечисляются элементы списка, разделенные запятой.

3. Как организовано хранение списков в оперативной памяти?

Списки в оперативной памяти хранятся как последовательность ссылок на объекты, которые могут быть расположены в разных областях памяти.

4. Каким образом можно перебрать все элементы списка?

Для перебора всех элементов списка можно использовать цикл for или while, функцию map или списковое включение.

5. Какие существуют арифметические операции со списками?

Со списками можно выполнять операции сложения (+), умножения (*), вычитания (-), деления (/), а также проверять на равенство (==) и неравенство (!=).

6. Как проверить есть ли элемент в списке?

Для проверки наличия элемента в списке можно использовать оператор in.

7. Как определить число вхождений заданного элемента в списке?

Чтобы определить число вхождений заданного элемента в список, можно использовать метод count.

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

Для добавления элемента в список можно использовать метод append или insert.

9. Как выполнить сортировку списка?

Список можно отсортировать с помощью метода sort или функции sorted.

10. Как удалить один или несколько элементов из списка?

Чтобы удалить один или несколько элементов из списка, можно использовать методы `remove`, `pop` или `del`.

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

Списковое включение (`list comprehension`) - это синтаксическая конструкция, позволяющая создавать новый список на основе существующего, применяя к его элементам определенное выражение или функцию.

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

Доступ к элементам списков с помощью срезов осуществляется с помощью квадратных скобок и указания индексов начала и конца интересующей нас части списка.

13. Какие существуют функции агрегации для работы со списками?

Функции агрегации для работы со списками включают в себя `sum`, `min`, `max`, `len`, `any`, `all` и др.

14. Как создать копию списка?

Чтобы создать копию списка, можно использовать метод `copy` или срез `[:]`.

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

Функция `sorted` возвращает новый отсортированный список, не изменяя исходный список, в отличие от метода `sort`, который сортирует список на месте.