

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций
«Исследование возможностей Git для работы с
локальными репозиториями»**

**Отчет по лабораторной работе № 2.4
по дисциплине «Основы программной инженерии»**

Выполнил студент группы
ПИЖ-б-о-21-1

Трушева В. О. .« » 2022г.

Подпись студента _____

Работа защищена «
» _____ 20__ г.

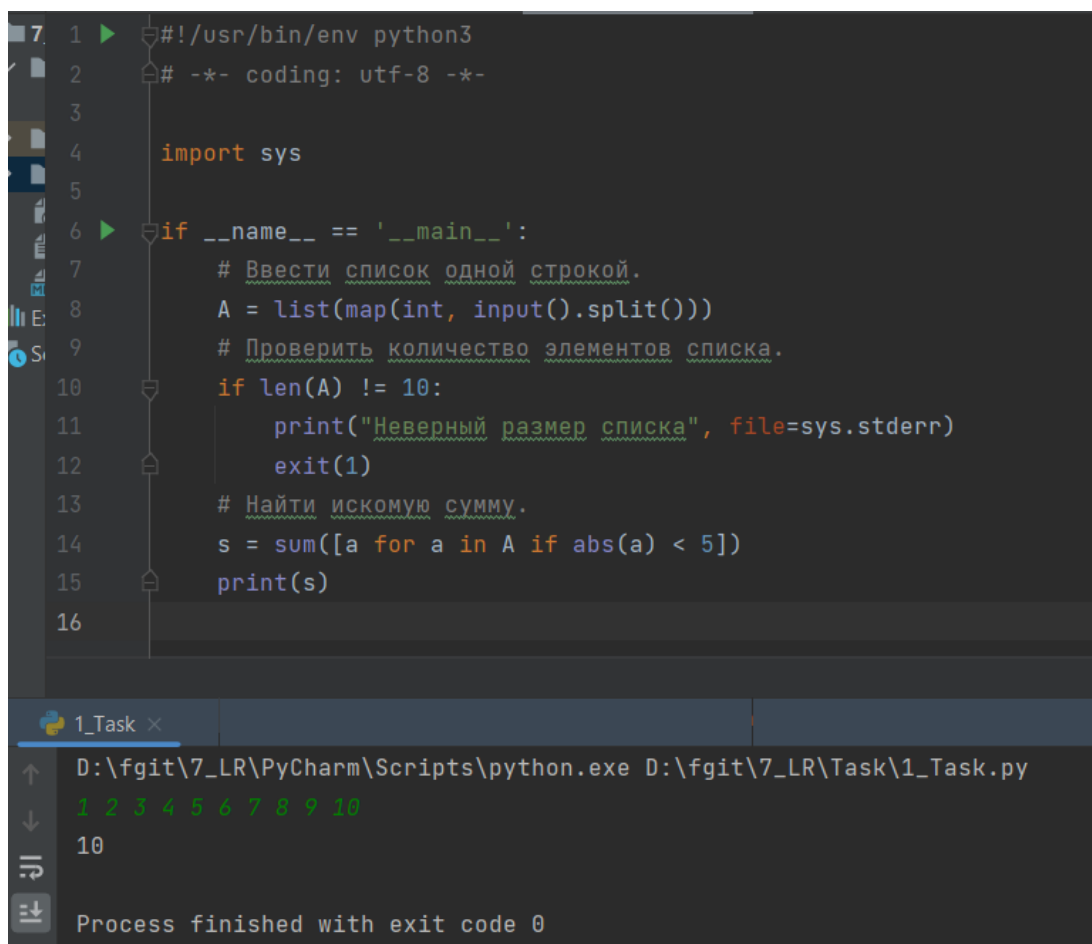
Проверила Воронкин Р.А.

(подпись)

Ставрополь 2022

Методика и порядок выполнения работы

1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.4. Дополните файл .gitignore необходимыми правилами для работы с IDE PyCharm.
5. Организуйте свой репозиторий в соответствие с моделью ветвления git-flow.
6. Создайте проект PyCharm в папке репозитория.
7. Проработайте примеры лабораторной работы. Создайте для каждого примера отдельный модуль языка Python. Зафиксируйте изменения в репозитории.



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

1_Task x

D:\fgit\7_LR\PyCharm\Scripts\python.exe D:\fgit\7_LR\Task\1_Task.py

1 2 3 4 5 6 7 8 9 10

10

Process finished with exit code 0

Рисунок 1 – Пример 1

```
4 import sys
5
6 if __name__ == '__main__':
7     # ввести список одной строкой.
8     a = list(map(int, input().split()))
9     # если список пуст, завершить программу.
10    if not a:
11        print("заданный список пуст", file=sys.stderr)
12        exit(1)
13
14    # определить индексы мин и макс элементов.
15    a_min = a_max = a[0]
16    i_min = i_max = 0
17    for i, item in enumerate(a):
18        if item < a_min:
19            i_min, a_min = i, item
20        if item >= a_max:
21            i_max, a_max = i, item
22
23    # проверить индексы и обменять их местами.
24    if i_min > i_max:
25        i_min, i_max = i_max, i_min
26
27    # посчитать количество положительных элементов.
28    count = 0
29    for item in a[i_min+1:i_max]:
30        if item > 0:
31            count += 1
32
33 if __name__ == '__main__': > for item in a[i_min+1:i_max] > if item > 0
```

un: 2_Task x

D:\fgit\7_LR\PyCharm\Scripts\python.exe D:\fgit\7_LR\Task\2_Task

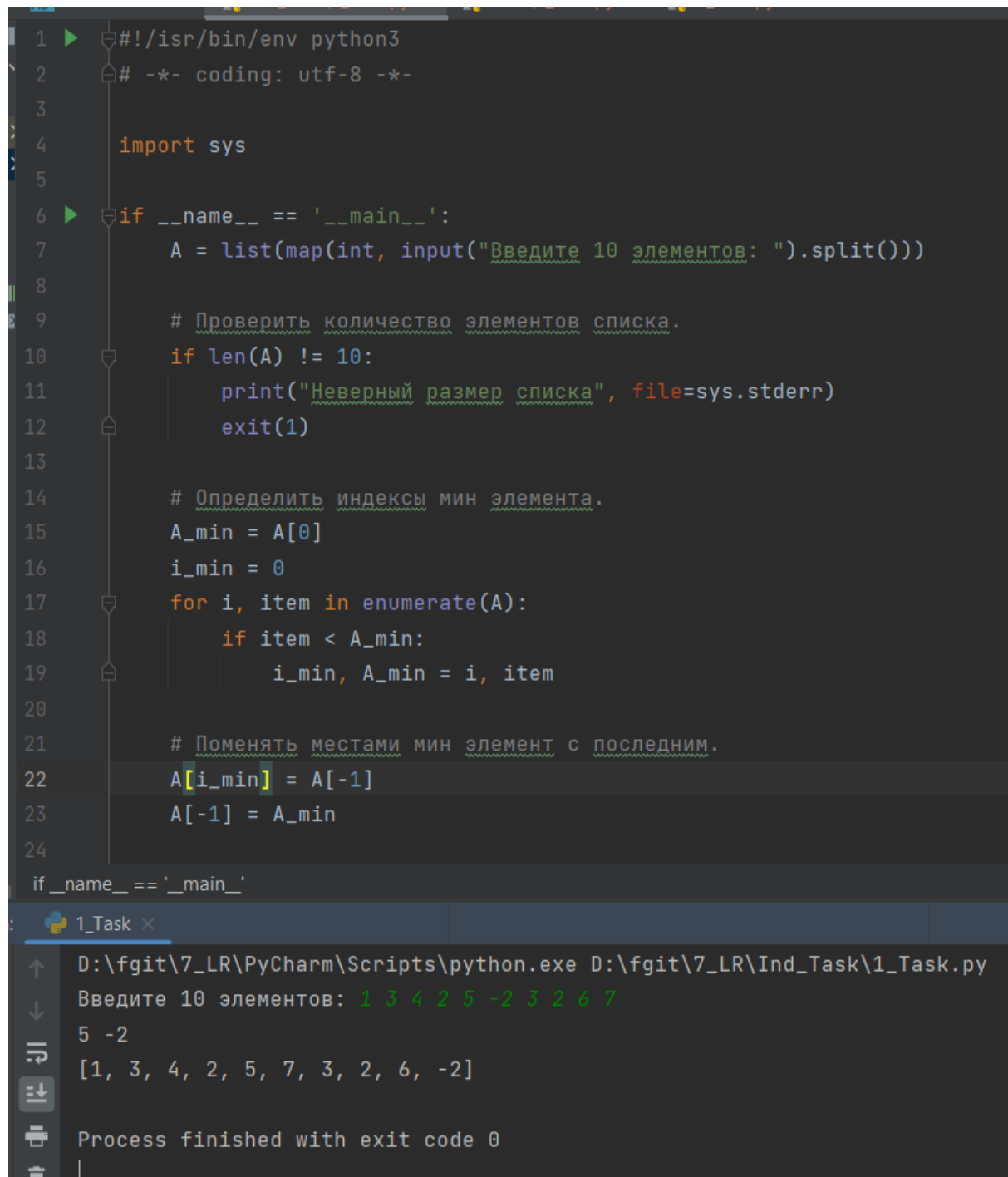
1 4 5 8 7 9 8 8

2

Рисунок 2 – Пример 2

8. Приведите в отчете скриншоты результатов выполнения каждой из программ примеров при различных исходных данных вводимых с клавиатуры.

Вариант - 3. Условие. Ввести список A из 10 элементов, найти наименьший элемент и переставить его с последним элементом. Преобразованный массив вывести.



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6  if __name__ == '__main__':
7      A = list(map(int, input("Введите 10 элементов: ").split()))
8
9      # Проверить количество элементов списка.
10     if len(A) != 10:
11         print("Неверный размер списка", file=sys.stderr)
12         exit(1)
13
14     # Определить индексы мин элемента.
15     A_min = A[0]
16     i_min = 0
17     for i, item in enumerate(A):
18         if item < A_min:
19             i_min, A_min = i, item
20
21     # Поменять местами мин элемент с последним.
22     A[i_min] = A[-1]
23     A[-1] = A_min
24
25 if __name__ == '__main__':
```

1_Task x

D:\fgit\7_LR\PyCharm\Scripts\python.exe D:\fgit\7_LR\Ind_Task\1_Task.py

Введите 10 элементов: 1 3 4 2 5 -2 3 2 6 7

5 -2

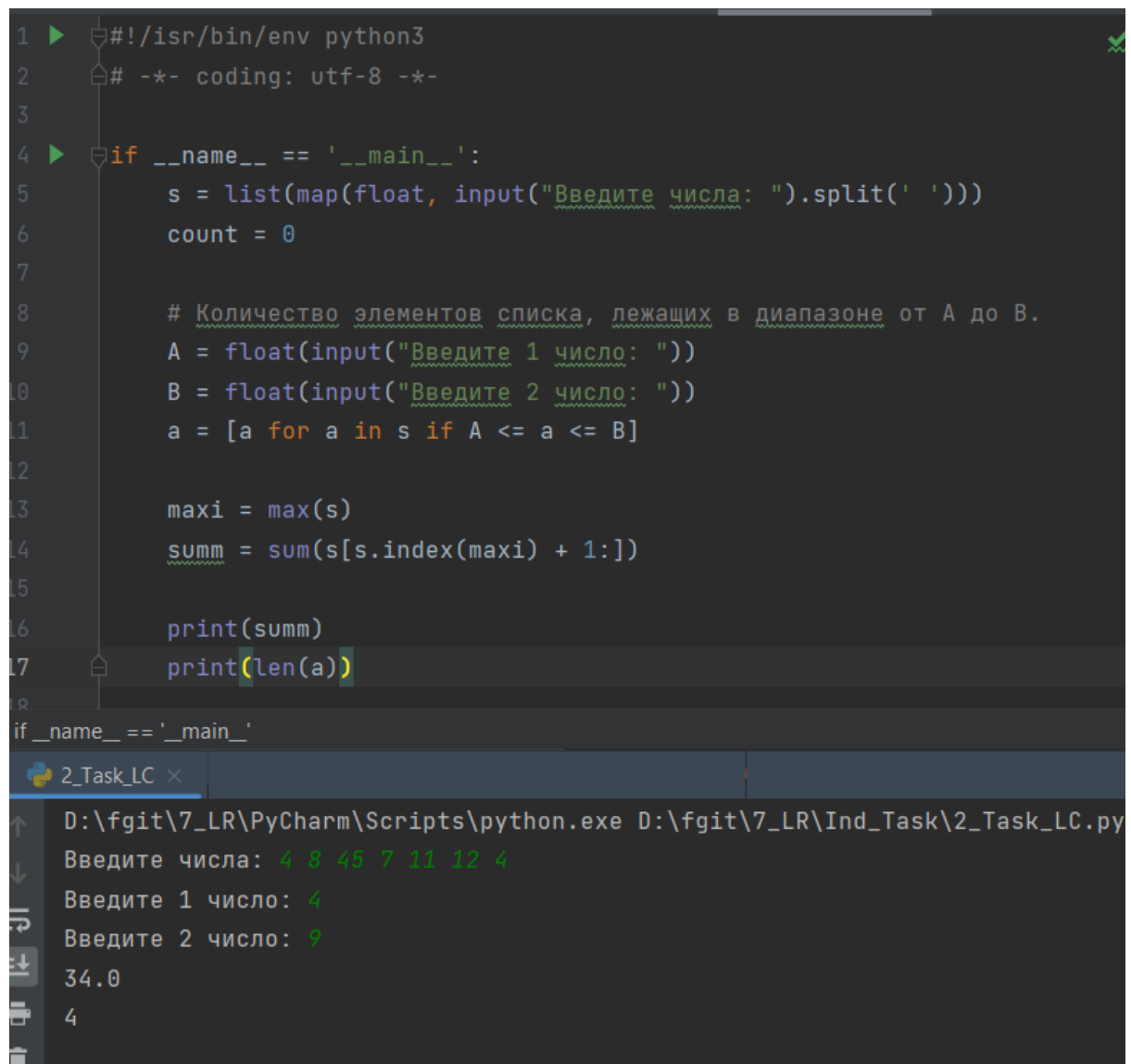
[1, 3, 4, 2, 5, 7, 3, 2, 6, -2]

Process finished with exit code 0

Рисунок 3 – Индивидуальный пример 1

Вариант - 12. Условие. В списке, состоящем из вещественных элементов, вычислить:

1. Количество элементов списка, лежащих в диапазоне от A до B;
2. Сумму элементов списка, расположенных после максимального элемента.



```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == '__main__':
5     s = list(map(float, input("Введите числа: ").split(' ')))
6     count = 0
7
8     # Количество элементов списка, лежащих в диапазоне от A до B.
9     A = float(input("Введите 1 число: "))
10    B = float(input("Введите 2 число: "))
11    a = [a for a in s if A <= a <= B]
12
13    maxi = max(s)
14    summ = sum(s[s.index(maxi) + 1:])
15
16    print(summ)
17    print(len(a))
18
if __name__ == '__main__':
```

2_Task_LC x

D:\fgit\7_LR\PyCharm\Scripts\python.exe D:\fgit\7_LR\Ind_Task\2_Task_LC.py

Введите числа: 4 8 45 7 11 12 4

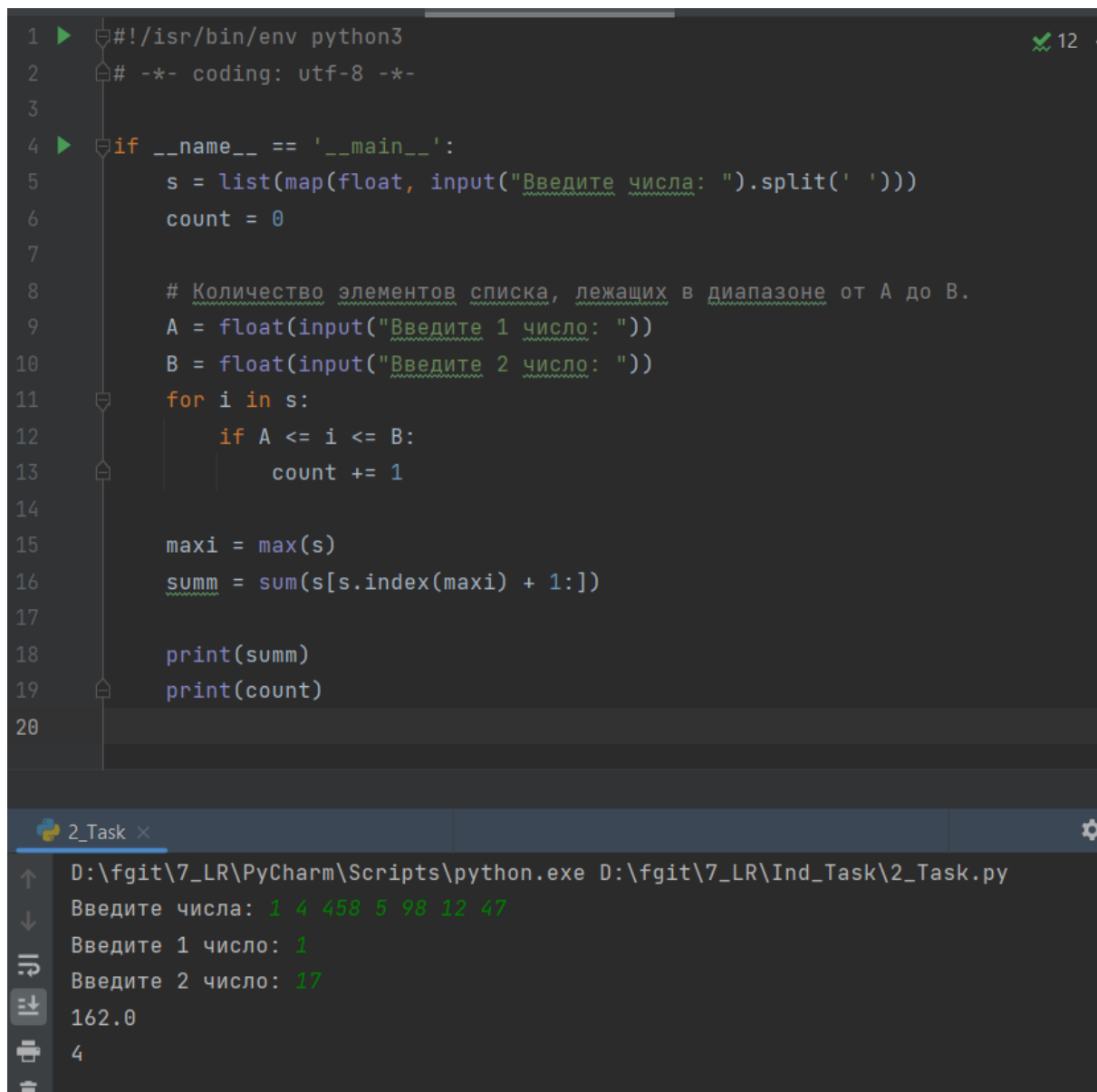
Введите 1 число: 4

Введите 2 число: 9

34.0

4

Рисунок 4 – Индивидуальный пример 2



```
1  ▶  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  ▶  if __name__ == '__main__':
5      s = list(map(float, input("Введите числа: ").split(' ')))
6      count = 0
7
8      # Количество элементов списка, лежащих в диапазоне от A до B.
9      A = float(input("Введите 1 число: "))
10     B = float(input("Введите 2 число: "))
11     for i in s:
12         if A <= i <= B:
13             count += 1
14
15     maxi = max(s)
16     summ = sum(s[s.index(maxi) + 1:])
17
18     print(summ)
19     print(count)
20
```

2_Task x

D:\fgit\7_LR\PyCharm\Scripts\python.exe D:\fgit\7_LR\Ind_Task\2_Task.py

Введите числа: 1 4 458 5 98 12 47

Введите 1 число: 1

Введите 2 число: 17

162.0

4

Рисунок 5 – Индивидуальный пример 2

9. Приведите в отчете скриншоты работы программ решения индивидуальных заданий.
10. Зафиксируйте сделанные изменения в репозитории.
11. Добавьте отчет по лабораторной работе в формате PDF в папку doc репозитория. Зафиксируйте изменения.
12. Выполните слияние ветки для разработки с веткой main / master.
13. Отправьте сделанные изменения на сервер GitHub.

14. Отправьте адрес репозитория GitHub на электронный адрес преподавателя.

Вопросы для защиты работы

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?

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?

```
for elem in my_list:
```

5. Какие существуют арифметические операции со списками?

+, *

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?

```
list.count(„элемент“)
```

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

Метод insert можно использовать, чтобы вставить элемент в список.

9. Как выполнить сортировку списка?

```
list.sort()
```

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[<начало среза>:<конец среза>:<шаг>]`

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?

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