

**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

Горшков Виталий Игоревич

Ставрополь 2022

Выполнение работы:

1. Создал репозиторий в GitHub «rep 2.2» в который добавил .gitignore, который дополнил правила для работы с IDE PyCharm с ЯП Python, выбрал лицензию MIT, клонировал его на лок. сервер и организовал в соответствии с моделью ветвления git-flow.

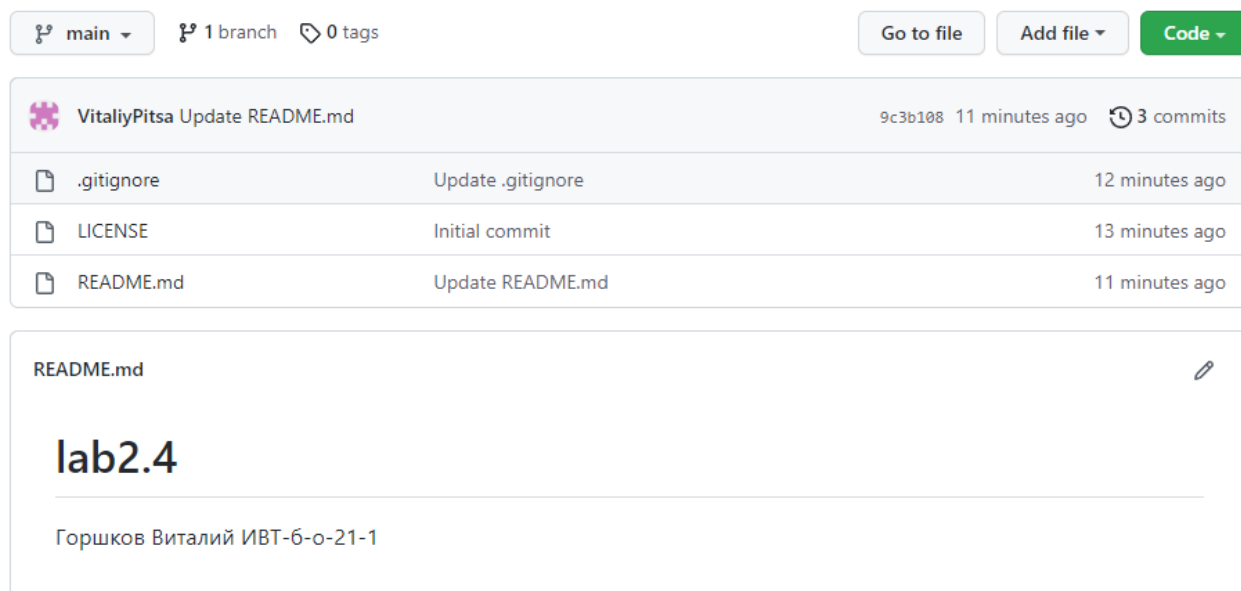


Рисунок 1 Создание репозитория

```
C:\Users\Виталий>cd /d C:\lab2.4

C:\lab2.4>git clone https://github.com/VitaliyPitsa/lab2.4.git
Cloning into 'lab2.4'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 11 (delta 2), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (11/11), 5.00 KiB | 465.00 KiB/s, done.
Resolving deltas: 100% (2/2), done.

C:\lab2.4>cd /d C:\lab2.4\lab2.4
```

Рисунок 2 Клонирование репозитория

```
C:\lab2.4\lab2.4>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? []
Hooks and filters directory? [C:/lab2.4/lab2.4/.git/hooks]
```

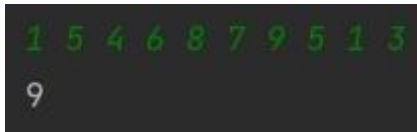
Рисунок 3 Организация репозитория в соответствии с моделью ветвления
git-flow

267 lines (213 sloc) | 5.26 KB

```
1 # Created by https://www.toptal.com/developers/gitignore/api/python,pycharm
2 # Edit at https://www.toptal.com/developers/gitignore?templates=python,pycharm
3 .idea/
4 ### PyCharm ###
5 # Covers JetBrains IDEs: IntelliJ, RubyMine, PhpStorm, AppCode, PyCharm, CLion, Android Studio, WebStorm and Rider
6 # Reference: https://intellij-support.jetbrains.com/hc/en-us/articles/206544839
7
8 # User-specific stuff
9 .idea/**/workspace.xml
10 .idea/**/tasks.xml
11 .idea/**/usage.statistics.xml
12 .idea/**/dictionaries
13 .idea/**/shelf
14
15 # AWS User-specific
16 .idea/**/aws.xml
17
18 # Generated files
19 .idea/**/contentModel.xml
20
21 # Sensitive or high-churn files
22 .idea/**/dataSources/
23 .idea/**/dataSources.ids
24 .idea/**/dataSources.local.xml
25 .idea/**/sqlDataSources.xml
26 .idea/**/dynamic.xml
27 .idea/**/uiDesigner.xml
28 .idea/**/dbnavigator.xml
29
30 # Gradle
31 .idea/**/gradle.xml
32 .idea/**/libraries
33
34 # Gradle and Maven with auto-import
35 # When using Gradle or Maven with auto-import, you should exclude module files,
36 # since they will be recreated, and may cause churn. Uncomment if using
37 # auto-import.
```

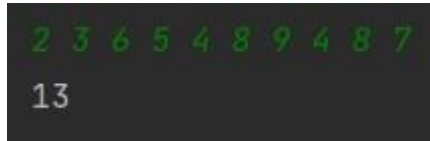
Рисунок 1.4 Изменение .gitignore

2. Создал проект PyCharm в папке репозитория, проработал примеры ЛР.



```
1 5 4 6 8 7 9 5 1 3
9
```

Рисунок 4 Рез-т выполнения программы



```
2 3 6 5 4 8 9 4 8 7
13
```

Рисунок 5 Рез-т выполнения программы

3. (8 вариант). Выполнил 2 индивидуальных задания.

```
C:\Users\Виталий\PycharmProjects\p  
введите список 1 0 2 3 4  
(1, 0)
```

Рисунок 6 Вывод программы индивидуального задания 1

```

C:\Users\Виталий\PycharmProject>
введите список 1 0 3 4 5
0
12
[0, 4, 1, 3, 5]

```

Рисунок 3.2 Вывод программы индивидуального задания 2

4. Сделал коммит, выполнил слияние с веткой main, и запустил изменения в уд. репозиторий.

```

C:\lab2.4\lab2.4>git add .

C:\lab2.4\lab2.4>git commit -m "1commit"
[develop ef00ac1] 1commit
 4 files changed, 71 insertions(+)
 create mode 100644 individual1.py
 create mode 100644 individual2.py
 create mode 100644 primer1.py
 create mode 100644 primer2.py

C:\lab2.4\lab2.4>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

```

Рисунок 4.1 коммит изменений и переход на ветку main

```

C:\lab2.4\lab2.4>git merge develop
Updating 9c3b108..ef00ac1
Fast-forward
 individual1.py | 8 ++++++++
 individual2.py | 9 ++++++++
 primer1.py     | 20 ++++++
 primer2.py     | 34 ++++++
 4 files changed, 71 insertions(+)
 create mode 100644 individual1.py
 create mode 100644 individual2.py
 create mode 100644 primer1.py
 create mode 100644 primer2.py

```


Рисунок 4.2 Слияние ветки main с develop

```








C:\lab2.4\lab2.4>git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.54 KiB | 1.54 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/VitaliyPitsa/lab2.4.git
 9c3b108..ef00ac1  main -> main

```


Рисунок 4.3 Пуш изменений на удаленный сервер

 VitaliyPitsa 1 commit

ef00ac1 1 minute ago 4 commits

 .gitignore	Update .gitignore	3 hours ago
 LICENSE	Initial commit	3 hours ago
 README.md	Update README.md	3 hours ago
 individual1.py	1 commit	1 minute ago
 individual2.py	1 commit	1 minute ago
 primer1.py	1 commit	1 minute ago
 primer2.py	1 commit	1 minute ago

README.md



lab2.4

Горшков Виталий ИБТ-6-о-21-1

Рисунок 4.4 Изменения на удаленном сервере

Контр. вопросы и ответы на них:

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.