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 №2.3

Дисциплина: «Основы кроссплатформенного программирования»

Тема: «Работа со строками в языке Python»

Выполнил: студент 1 курса

группы ИВТ-б-о-21-1

Назаров Никита Юрьевич

Ставрополь 2022

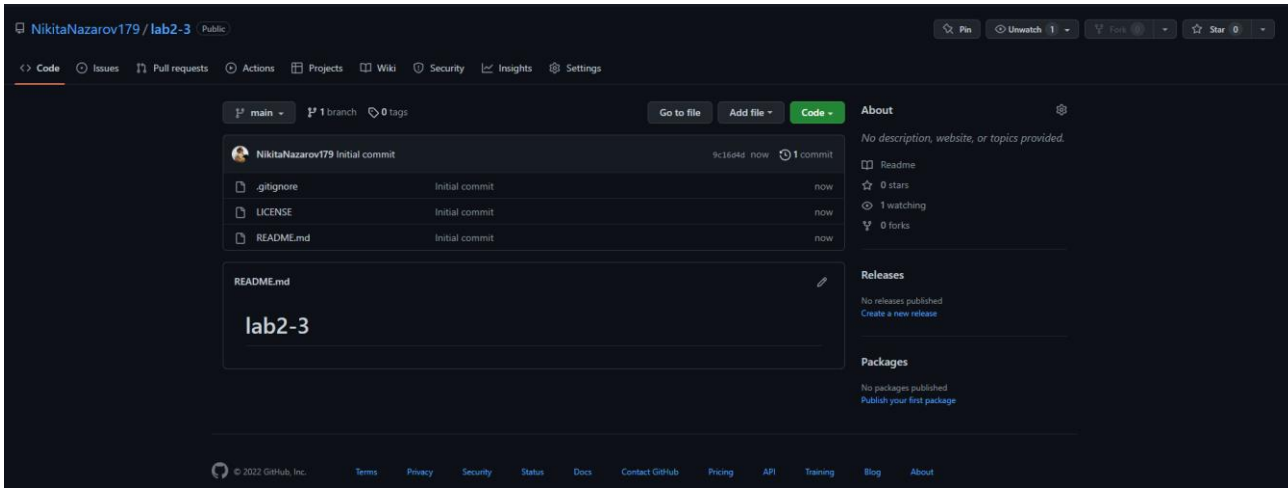

Рисунок 1 – новый репозиторий

```
H:\cross\git\nazarov>git clone https://github.com/NikitaNazarov179/lab2-3.git
Cloning into 'lab2-3'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
```

Рисунок 2 – клонирование репозитория

.gitignore – Блокнот

Файл Правка Формат Вид Справка

```
# Created by .ignore support plugin (hsz.mobi)
### Python template
# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
*$py.class

# C extensions
*.so

# Distribution / packaging
.Python
env/
build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
parts/
sdist/
var/
*.egg-info/
.installed.cfg
*.egg

# PyInstaller
#  Usually these files are written by a python script from a template
#  before PyInstaller builds the exe, so as to inject date/other infos i
*.manifest
*.spec

# Installer logs
pip-log.txt
pip-delete-this-directory.txt

# Unit test / coverage reports
htmlcov/
.tox/
.coverage
.coverage.*
```

Стр 1, стлб 1 | 100% | Windows (CRLF) | UTF-8

Рисунок 3 – редактирование файла gitignore

```
H:\cross\git\nazarov\lab2-3>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? [H:/cross/git/nazarov/lab2-3/.git/hooks]
```

Рисунок 4 – организовал репозиторий в соответствии с моделью git flow

```
H:\Python\python.exe H:/cross/git/nazarov/lab2-3/примеры/1.py
Введите предложение: я люблю СКФУ
Предложение после замены: я_люблю_СКФУ
```

Рисунок 5 – результат работы 1 примера

```
H:\Python\python.exe H:/cross/git/nazarov/lab2-3/примеры/2.py
Введите слово: Программирование
Програмрование

Process finished with exit code 0
```

Рисунок 6 – результат работы 2 примера

```
H:\Python\python.exe H:/cross/git/nazarov/lab2-3/примеры/3.py
Введите предложение: я люблю СКФУ
Введите длину: 15
я   люблю  СКФУ

Process finished with exit code 0
```

Рисунок 7 – результат работы 3 примера

```
H:\Python\python.exe "H:/cross/git/nazarov/lab2-3/инд. задания/1.py"
Введите предложение:
asdhf+gjoper*wehf+ertkp34
2
1

Process finished with exit code 0
```

Рисунок 8 – индивидуальное задание 1

```
H:\Python\python.exe "H:/cross/git/nazarov/lab2-3/инд. задания/2.py"
Введите предложение:
про твк апро укопку ро ул аз про
нет твк анет укопку ро ул аз нет

Process finished with exit code 0
```

Рисунок 9 – индивидуальное задание 2

```
H:\Python\python.exe "H:/cross/git/nazarov/lab2-3/инд. задания/3.py"
процессор

Process finished with exit code 0
```

Рисунок 10 – индивидуальное задание 3

```
H:\Python\python.exe "H:/cross/git/nazarov/lab2-3/инд. задания/повыш.py"
Введите предложение:
sdfjgh oroeit f fgohker rg
1

Process finished with exit code 0
```

Рисунок 11 – задание повышенной сложности

```
H:\cross\git\nazarov\lab2-3>git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

H:\cross\git\nazarov\lab2-3>git merge develop
Updating 8cea738..7e14395
Fast-forward
 .idea/.gitignore                                   |  3 --
 .../1.py"                                          | 20 ++++++++
 .../2.py"                                          |  8 +++
 .../3.py"                                          | 10 ++++
 .../\320\277\320\276\320\262\321\213\321\210.py"   |  9 ++++
 .../2.py"                                          | 13 +++++
 .../3.py"                                          | 57 ++++++++++++++++++++++
 7 files changed, 117 insertions(+), 3 deletions(-)
 delete mode 100644 .idea/.gitignore
 create mode 100644 "\320\270\320\275\320\264. \320\267\320\260\320\264\320\260\320\275\320\270\321\217/1.py"
 create mode 100644 "\320\270\320\275\320\264. \320\267\320\260\320\264\320\260\320\275\320\270\321\217/2.py"
 create mode 100644 "\320\270\320\275\320\264. \320\267\320\260\320\264\320\260\320\275\320\270\321\217/3.py"
 create mode 100644 "\320\270\320\275\320\264. \320\267\320\260\320\264\320\260\320\275\320\270\321\217/\320\277\320\276
\320\262\321\213\321\210.py"
 create mode 100644 "\320\277\321\200\320\270\320\274\320\265\321\200\321\213/2.py"
 create mode 100644 "\320\277\321\200\320\270\320\274\320\265\321\200\321\213/3.py"

H:\cross\git\nazarov\lab2-3>
```

Рисунок 12 – слияние веток

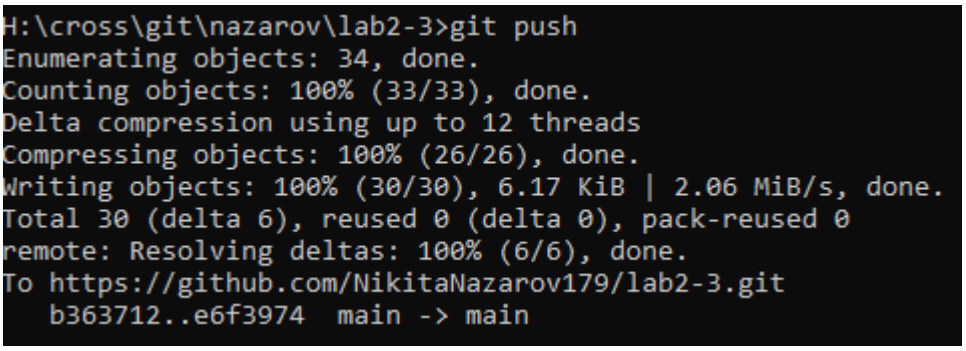

Рисунок 13 – пуш веток на уд.репозиторий

Контр. вопросы и ответы на них:

1. **Что такое строки в языке Python?**

Строки в Python - упорядоченные последовательности символов, используемые для хранения и представления текстовой информации, поэтому с помощью строк можно работать со всем, что может быть представлено в текстовой форме.

2. **Какие существуют способы задания строковых литералов в языке Python?**

Строки в апострофах и в кавычках, экранированные последовательности, "сырые" строки, строки в тройных апострофах или кавычках

3. **Какие операции и функции существуют для строк?**

Сложение, дублирование, длина строки, длина строки, извлечение среза и т. д.

4. **Как осуществляется индексирование строк?**

Доступ к символам в строках основан на операции индексирования – после строки или имени переменной, ссылающейся на строку, в квадратных скобках указываются номера позиций необходимых символов.

5. **Как осуществляется работа со срезами для строк?**

Есть три формы срезов. Самая простая форма среза: взятие одногосимвола строки, а именно, S[i] — это срез, состоящий из одного символа, который имеет номер i, при этом считая, что нумерация начинается с числа 0.То есть если S = ‘Hello’, то S[0]=='H', S[1]=='e', S[2]=='l', S[3]=='l', S[4]=='o'.

Если указать отрицательное значение индекса, то номер будет отсчитываться с конца, начиная с номера -1.

Срез с двумя параметрами: S[a:b] возвращает подстроку из b-a символов,

начиная с символа с индексом a, то есть до символа с индексом b, не включая его.

6. Почему строки Python относятся к неизменяемому типу данных?

Строки — один из типов данных, которые Python считает неизменяемыми, что означает невозможность их изменять. Python дает возможность изменять (заменять и перезаписывать) строки.

7. Как проверить то, что каждое слово в строке начинается с заглавнойбуквы?

string.istitle()

8. Как проверить строку на вхождение в неё другой строки?

string.find()

9. Как найти индекс первого вхождения подстроки в строку?

s.partition(<sep>)

10. Как подсчитать количество символов в строке?

len(s)

11. Как подсчитать то, сколько раз определённый символ встречается в строке?

s.count(<sub>)

12. Что такое f-строки и как ими пользоваться?

Эти строки улучшают читаемость кода, а также работают быстрее чем другие способы форматирования. F-строки задаются с помощью литерала «f» перед кавычками. Пример: print(f"Меня зовут {name} Мне {age} лет.")

13. Как найти подстроку в заданной части строки?

s.find(значение, начало, конец)

14. Как вставить содержимое переменной в строку, воспользовавшись методом format()?

print('{}'.format(s))

15. Как узнать о том, что в строке содержатся только цифры?

s.isdigit()

16. Как разделить строку по заданному символу?

str.split()

17. Как проверить строку на то, что она составлена только из строчных

б

укв?

s.isalpha()

18. Как проверить то, что строка начинается со строчной буквы?

s.istitle()

19. Можно ли в Python прибавить целое число к строке?

Нет

20. Как «перевернуть» строку?

s.reverse()

21. Как объединить список строк в одну строку, элементы которой

разделены дефисами?

str.split('-')

22. Как привести всю строку к верхнему или нижнему регистру?

s.upper()s.lower

23. Как преобразовать первый символ строки к верхнему регистру?

s.capitalize()

24. Как проверить строку на то, что она составлена только из прописных букв?

s.isupper()

25. В какой ситуации вы воспользовались бы методом splitlines() ?

s.splitlines() делит s на строки и возвращает их в списке. Любой из следующих символов или последовательностей символов считается границей строки.

26. Как в заданной строке заменить на что-либо все вхождения некоей подстроки?

s.replace(old, new)

27. Как проверить то, что строка начинается с заданной последовательности символов, или заканчивается заданной последовательностью символов?

str.startswith() и str.endswith()

28. Как узнать о том, что строка включает в себя только пробелы?

s. isspace()

29. Что случится, если умножить некую строку на 3?

Asd*3 = AsdAsdAsd

30. Как привести к верхнему регистру первый символ каждого слова встроке?

s.title()

31. Как пользоваться методом partition()?

Метод partition() разбивает строку при первом появлении строки аргумента и возвращает кортеж, содержащий часть перед разделителем, строку аргумента и часть после разделителя.

32. В каких ситуациях пользуются методом rfind()?

s.rfind(<sub>) возвращает индекс последнего вхождения подстроки <sub> в s , который соответствует началу <sub>.