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_3
дисциплины
«Основы программной инженерии»

Выполнил:
Разворотников Денис Сергеевич
2 курс, группа ПИЖ-б-о-21-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

_____
(подпись)

Проверил:
_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь 2022

Цель: приобретение навыков по работе со строками при написании программ с помощью языка программирования Python version .3.x

1. Был создан общедоступного репозиторий в GitHub в котором были добавлены gitignore, правила для работы с IDE PyCharm с ЯП Python и лицензия MIT, репозиторий был клонировал на локальный сервер и организован в соответствие с моделью ветвления git-flow.

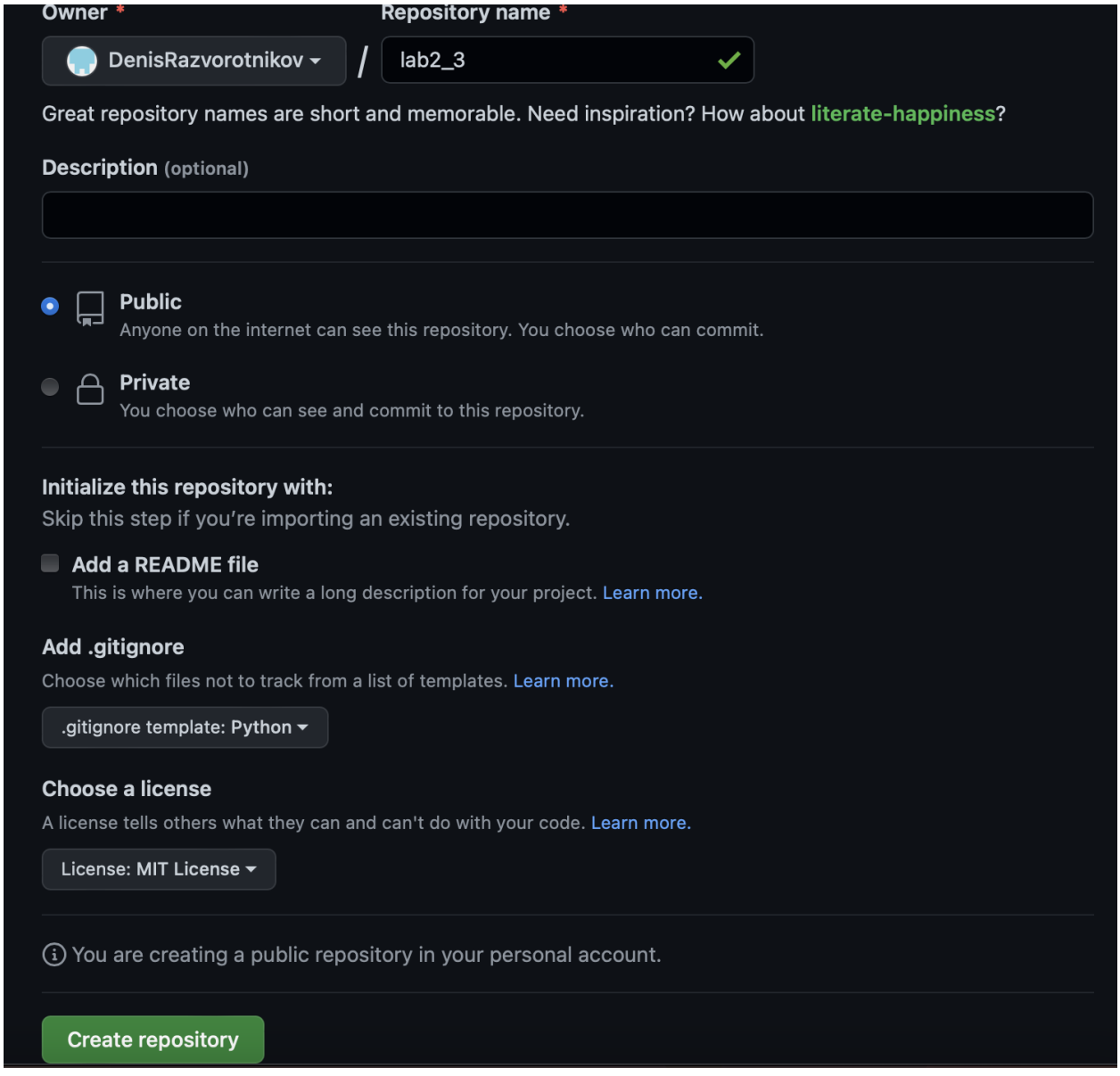

Рисунок 1.1 – Создание общедоступного репозитория

```
denisrazvorotnikov@Air-Denis lab2_3 % git branch -a
* develop
  main
  remotes/origin/HEAD -> origin/main
  remotes/origin/main
```

Рисунок 1.2 – Репозиторий был организован в соответствии с моделью ветвления git-flow

2. Была создана папка pycharm в которую были помещены примеры.

```
/Users/denisrazvorotnikov/Desktop/lab2_2/PY/bin/python /Users/d
Введите предложение: hy my friends
Предложение после замены:  {'hy_my_friends'}

Process finished with exit code 0
```

Рисунок 2.1 – Результат работы первого примера

```
/Users/denisrazvorotnikov/Desktop/lab2_2/PY/bin/python
Введите слово: denis
deis

Process finished with exit code 0
```

Рисунок 2.2 – Результат работы второго примера

```
/Users/denisrazvorotnikov/Desktop/lab2_2/PY/bin/python /Users/den
Введите предложение: hi friends
Введите длину: 12
hi   friends

Process finished with exit code 0
```

Рисунок 2.3 – Результат работы третьего примера

3. Было выполнено три индивидуальных задания

Задание 1

Дан текст. Подсчитать общее число вхождений в него символов «+» и «–».

```
#!/usr/bin/env python3
# -- coding: utf-8 -

if __name__ == '__main__':
s = input()
a = s.count("+")
b = s.count("-")
print(a + b)
```

```
/Users/denisrazvorotnikov/Desktop/lab2_2/PY/bin/python
+fds- +fds
3

Process finished with exit code 0
```

Рисунок 3.1 – Результат выполнения программы

Задание 2

Дано слово из четного числа букв. Поменять местами первую букву со второй, третью – с четвертой и т. д.

```
#!/usr/bin/env python3
# -- coding: utf-8 --
import sys

if __name__ == '__main__':
    s = input()
    a = list(s)
    if len(a) % 2 == 0:
        a[0], a[1] = s[1], s[0]
        a[2], a[3] = s[3], s[2]
        print(''.join(a))
    else:
        print(
            "Нечетное число букв в слове",
            file=sys.stderr
        )
```

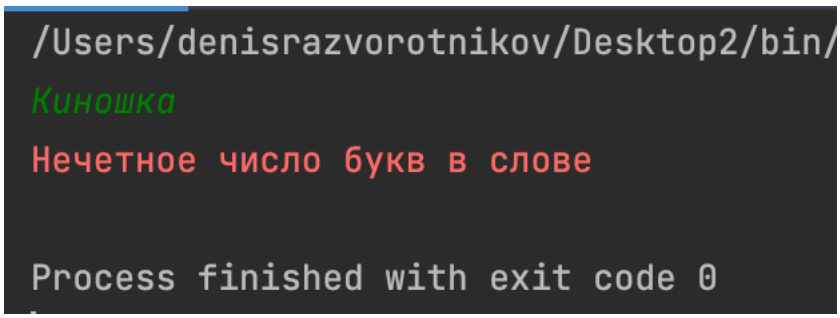

Рисунок 3.2 – Результат работы программы во втором случае

Задание 3

Дано слово из 12 букв. Переставить его буквы следующим способом: первая, двенадцатая, вторая, одиннадцатая, ..., пятая, восьмая, шестая, седьмая.

```
#!/usr/bin/env python3
# -- coding: utf-8 -

if __name__ == '__main__':
a = input()
print(a[: :-1])
```

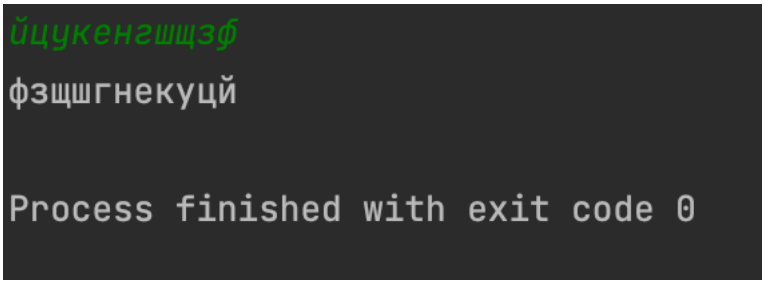

Рисунок 3.3 – Результат работы программы с предложением на русском

Задание повышенной сложности

24. Дано предложение. Напечатать все его различные слова.

```python
#!/usr/bin/env python3
# -- coding: utf-8 -

if __name__ == '__main__':
stroka = input('Введите предложение:\n')
if stroka[-1] == '.':
    stroka = stroka[0:-1]
    stroka_set = set(stroka.split(' '))
    print(stroka_set)
else:
    stroka_set = set(stroka.split(' '))
    print(stroka_set)
```

```
/Users/denisrazvorotnikov/Desktop2/bi
Введите предложение:
Предложение из букв
{'из', 'Предложение', 'букв'}
```

Рисунок 3.4 – Результат работы программы

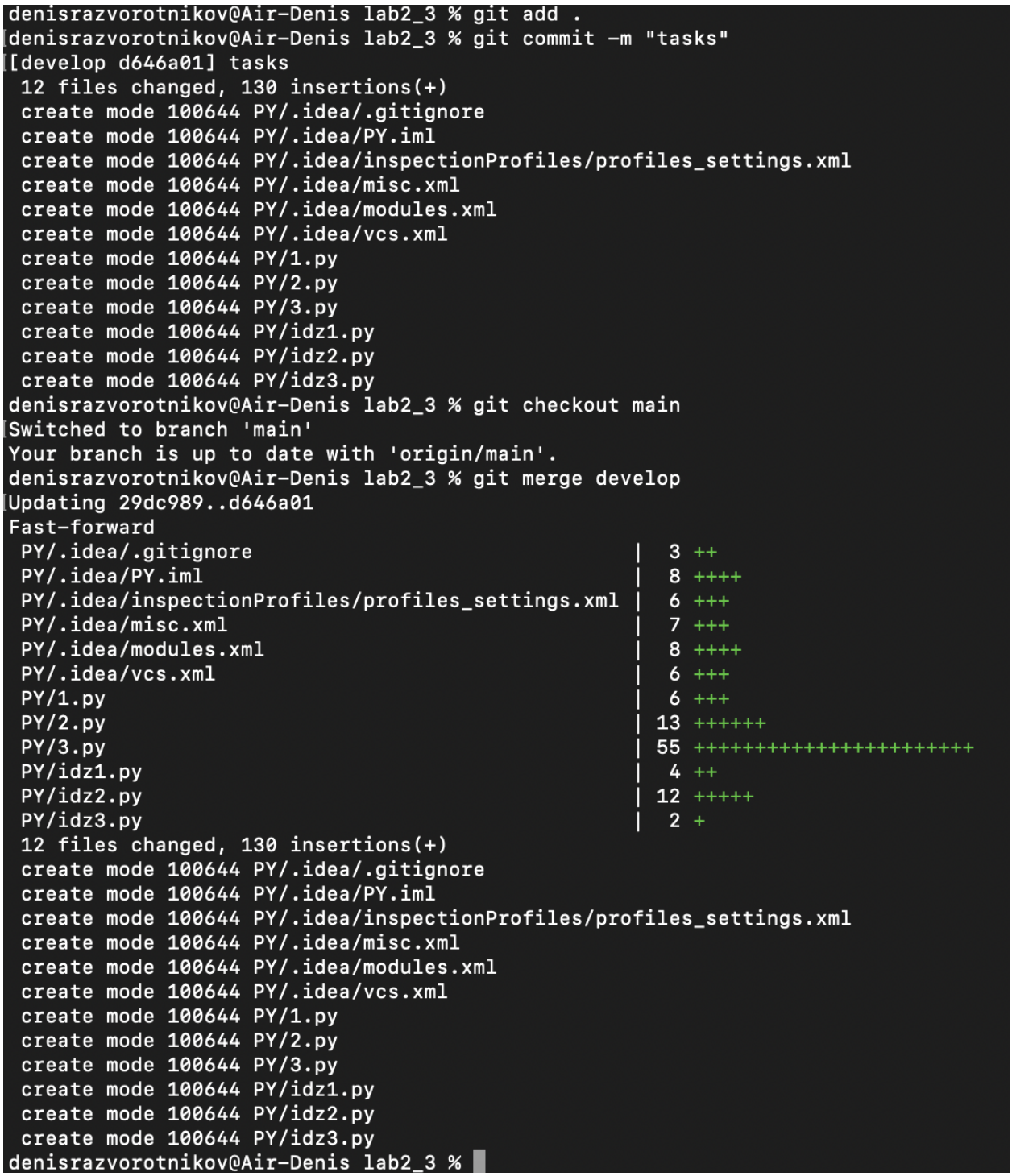

Рисунок 3.5 – Коммит изменений и слияние ветки develop с main

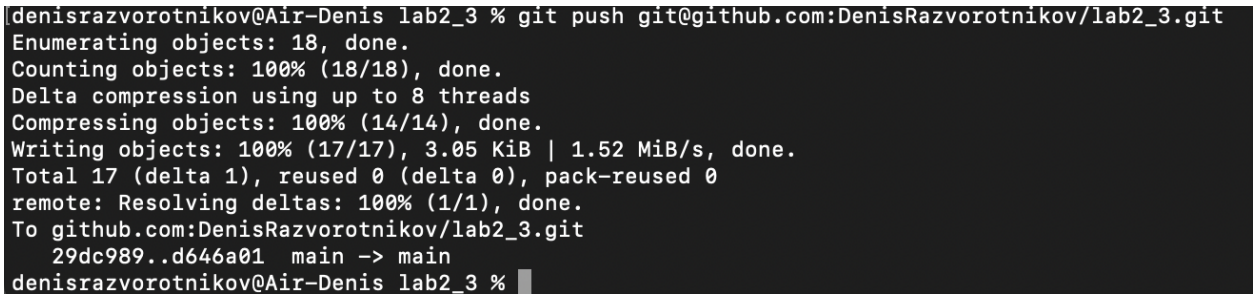

Рисунок 3.6 – Пуш на удаленный сервер

Вывод: в результате лабораторной работы были приобретены навыки по работе со строками при написании программ с помощью языка программирования Python version .3.x

Контрольные вопросы

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 одного символа 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 заглавной буквы?

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 букв?

str.split()

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

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 в строке?

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>.