

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**
**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

«Основы ветвления Git»

Отчет по лабораторной работе № 2.11

по дисциплине «Основы программной инженерии»

Выполнил студент группы ПИЖ-б-о-21-1

Образцова М. Д.. «27» декабря 2022г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____

(подпись)

Ставрополь 2022

МЕТОДИКА И ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Создала репозиторий в GitHub «OPI_LR_15» в который добавила .gitignore для работы с IDE PyCharm с ЯП Python, выбрала лицензию MIT.

Owner * Repository name *

obrMaria / OPI_LR_15

Great repository names are short and memorable. Need inspiration? How about [turbo-octo-spoon](#)?

Description (optional)

☒ Public
Anyone on the internet can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☒ Add a README file
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: Python

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: MIT License

This will set `main` as the default branch. Change the default name in your [settings](#).

i You are creating a public repository in your personal account.

Creating repository...

Активна

Рисунок 1 – Создание репозитория

```
M@DESKTOP-UVM9NOL MINGW64 ~/desktop/OPI_LR_15 (main)
$ 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:/Users/M/desktop/OPI_LR_15/.git/hooks]
```

Рисунок 2 – Организация репозитория в соответствии с моделью ветвления git-flow

```

M@DESKTOP-UVM9NOL MINGW64 ~/desktop/OPI_LR_15 (develop)
$ git add .

M@DESKTOP-UVM9NOL MINGW64 ~/desktop/OPI_LR_15 (develop)
$ git status
On branch develop
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   .gitignore

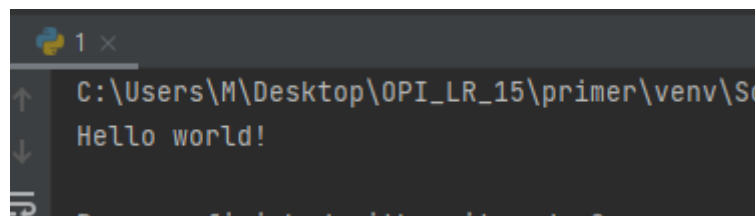
M@DESKTOP-UVM9NOL MINGW64 ~/desktop/OPI_LR_15 (develop)
$ git commit -m "gitignore"
[develop e769cde] gitignore
1 file changed, 140 insertions(+), 2 deletions(-)

M@DESKTOP-UVM9NOL MINGW64 ~/desktop/OPI_LR_15 (develop)
$ git

```

Рисунок 3 – Изменение .gitignore

1. Примеры проработаны.

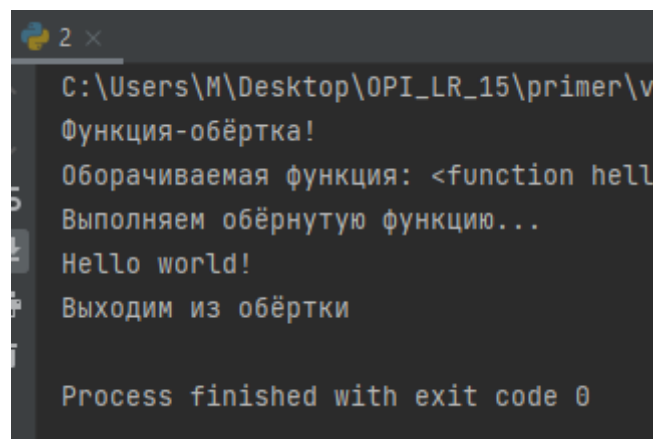


```

C:\Users\M\Desktop\OPI_LR_15\primer\venv\Scripts
Hello world!

```

Рисунок 4 – Результат выполнения программы 1-го примеров



```

C:\Users\M\Desktop\OPI_LR_15\primer\venv\Scripts
Функция-обёртка!
Оборачиваемая функция: <function hello...>
Выполняем обёрнутую функцию...
Hello world!
Выходим из обёртки
Process finished with exit code 0

```

Рисунок 5 – Результат выполнения программы 1-го примеров

2. (20 вариант). Индивидуальное задание.

Объявите функцию, которая принимает строку, удаляет из нее

все подряд идущие пробелы и переводит ее в нижний регистр – малые буквы. Результат (строка) возвращается функцией. Определите декоратор, который строку, возвращенную функцией, переводит в азбуку Морзе, используя следующий словарь для замены русских букв и символа пробела на соответствующие последовательности из точек и тире:

```
morze = {'a': '.-', 'б': '-...-', 'в': '.--', 'г': '--.', 'д': '-..', 'е':  
'..', 'ё': '...', 'ж': '...-', 'з': '--...', 'и': '...', 'й': '----', 'к': '-.-',  
'л': '.-..', 'м': '--', 'н': '-.', 'о': '---', 'п': '-.-.', 'р': '.-.', 'с':  
'...', 'т': '-', 'у': '..-', 'ф': '...-', 'х': '....', 'ц': '-.-.', 'ч': '--  
-', 'ш': '----', 'щ': '---.', 'ъ': '-.-.-', 'ы': '-.-.-', 'ь': '-.-.-', 'э':  
'...-', 'ю': '...-', 'я': '...-', ' ': '-----'}
```

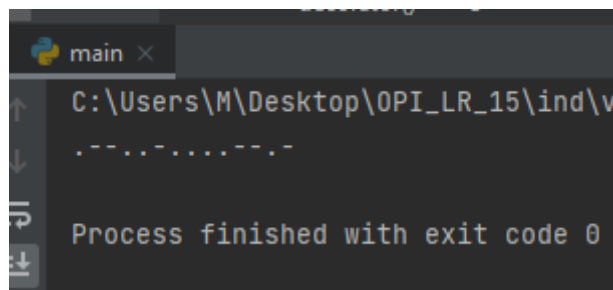


Рисунок 6 – Вывод программы индивидуального задания

Контрольные вопросы

1. Что такое декоратор?

Декоратор – это функция, которая позволяет обернуть другую функцию для расширения её функциональности без непосредственного изменения её кода.

2. Почему функции являются объектами первого класса?

Потому что с ними можно работать как с переменными, могут быть переданы как аргумент процедуры, могут быть возвращены как результат выполнения процедуры, могут быть включены в другие структуры данных.

3. Каково назначение функций высших порядков?

Основной задачей функций высших порядков является возможность принимать в качестве аргументов и возвращать другие функции.

4. Как работают декораторы?

Они берут декорируемую функцию в качестве аргумента и позволяет совершать с ней какие-либо действия до и после того, что сделает эта функция, не изменяя её.

5. Какова структура декоратора функций?

Функция `decorator` принимает в качестве аргумента функцию `func`, внутри функции `decorator` другая функция `wrapper`. В конце декоратора происходит возвращение функции `wrapper`.

6. Самостоятельно изучить как можно передать параметры декоратору, а не декорируемой функции?

Достаточно обернуть функцию декоратор в другую функцию, которая будет принимать аргументы. И сделать вывод функций `wrapper` и `decorator`.