

**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 №2.11

Дисциплина: «Программирование на Python»

Тема: «Замыкания в языке Python»

Выполнил: студент 2 курса

группы ИВТ-б-о-21-1

Назаров Никита Юрьевич

Выполнение работы.

```
# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
*$py.class

# C extensions
*.so

# Distribution / packaging
.Python
build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
parts/
sdist/
var/
wheels/
pip-wheel-metadata/
share/python-wheels/
*.egg-info/
.installed.cfg
*.egg
MANIFEST

# 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 into it.
*.manifest
*.spec

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

# Unit test / coverage reports
htmlcov/
.tox/
.nox/
.coverage
.coverage.*
.cache
nosetests.xml
coverage.xml
*.cover
*.py,cover
.hypothesis/
.pytest_cache/
```

Рисунок 1 – редактирование gitignore

```

C:\Users\student-09-525\Desktop\2.11>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/student-09-525/Desktop/2.11/.git/hooks]

C:\Users\student-09-525\Desktop\2.11>

```

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

The screenshot shows an IDE with a project explorer on the left, a code editor in the center, and a run console at the bottom. The project explorer shows a folder named '2.11' containing a 'doc' folder, 'primeri', '.gitignore', 'LICENSE', and 'README.md'. The code editor displays a Python file 'prim6_1.py' with the following code:

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  def mul5(a):
6      def helper(b):
7          return a * b
8      return helper
9
10
11 if __name__ == '__main__':
12     print(mul5(5)(6))
13

```

The run console at the bottom shows the command executed: `C:\Python37\python.exe C:/Users/student-09-525/Desktop/2.11/primeri/prim6_1.py`, the output `30`, and the message `Process finished with exit code 0`.

Рисунок 3 – проработал 1 пример

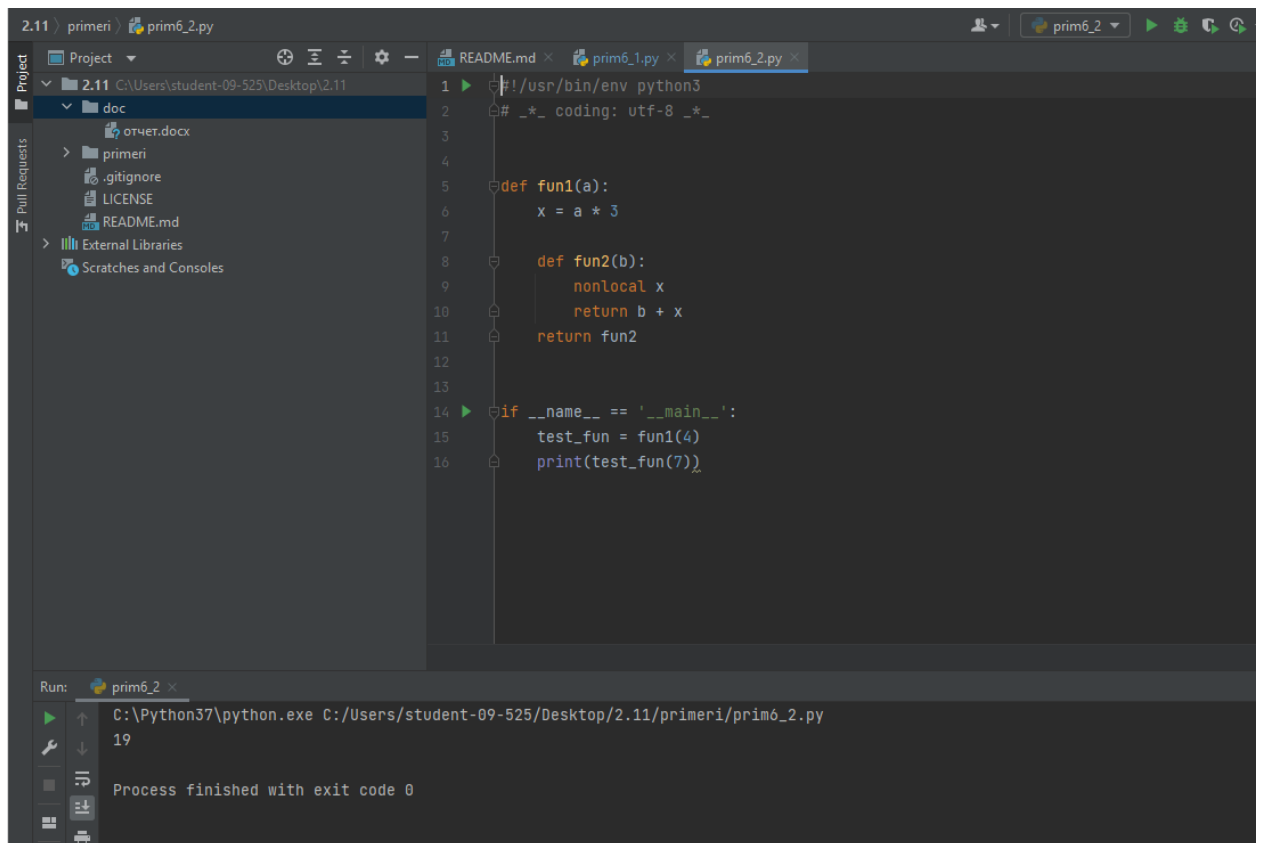


Рисунок 4 – проработал 2 пример

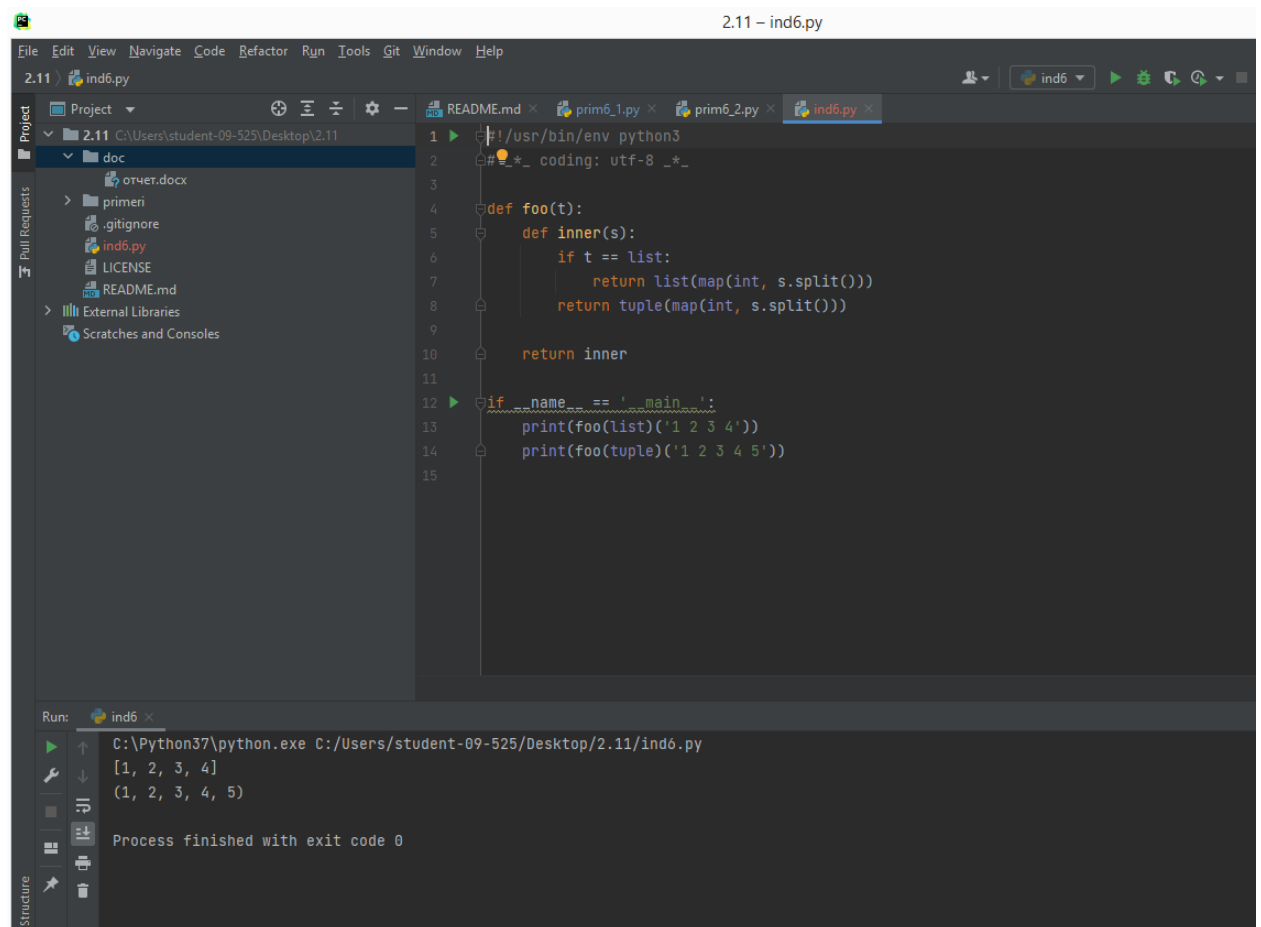
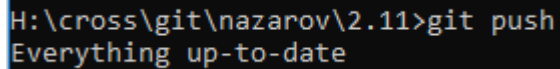


Рисунок 5 – выполнил индивидуальное задание



```
H:\cross\git\nazarov\2.11>git push
Everything up-to-date
```

Рисунок 6 – пуш изменений

Контр. вопросы и ответы на них:

1. Что такое замыкание?

Замыкание – это функция, в теле которой присутствуют ссылки на переменные, объявленные вне тела этой функции.

2. Как реализованы замыкания в языке программирования Python?

Замыканием в языке Python называется функция, вложенная в другую функцию и использующая переменные внешней функции.

3. Что подразумевает под собой область видимости Local?

Переменные с областью видимости Local (локальные переменные) могут быть использованы только внутри того блока кода, где она была объявлена.

4. Что подразумевает под собой область видимости Enclosing?

Для вложенных функций переменные из функции более высокого уровня имеют данную область видимости.

5. Что подразумевает под собой область видимости Global?

Область видимости Global означает, что данная переменная может быть использована (видна) во всём модуле (файле с расширением .py).

6. Что подразумевает под собой область видимости Build-in?

Это переменный уровня интерпретатора. Для их использования не нужно импортировать модули.