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 №16**

**дисциплины «Основы программной инженерии»**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнил:  Мелтонян Одиссей  2 курс, группа ПИЖ-б-о-22-1,  09.03.04 «Программная инженерия», направленность (профиль) «Разработка и сопровождение программного обеспечения», очная форма обучения  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | | Проверил:  Кандидат технических наук, доцент кафедры инфокоммуникаций Воронкин Р. А  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | |  | |

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_ Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ставрополь, 2024 г

Тема: Модули и пакеты

Цель работы: приобретение навыков по работе с модулями и пакетами языка программирования Python

Ход работы:

1. Изучен теоретический материал работы.

2. Создан общедоступный репозиторий на GitHub, в котором была использована лицензия MIT и язык программирования Python.

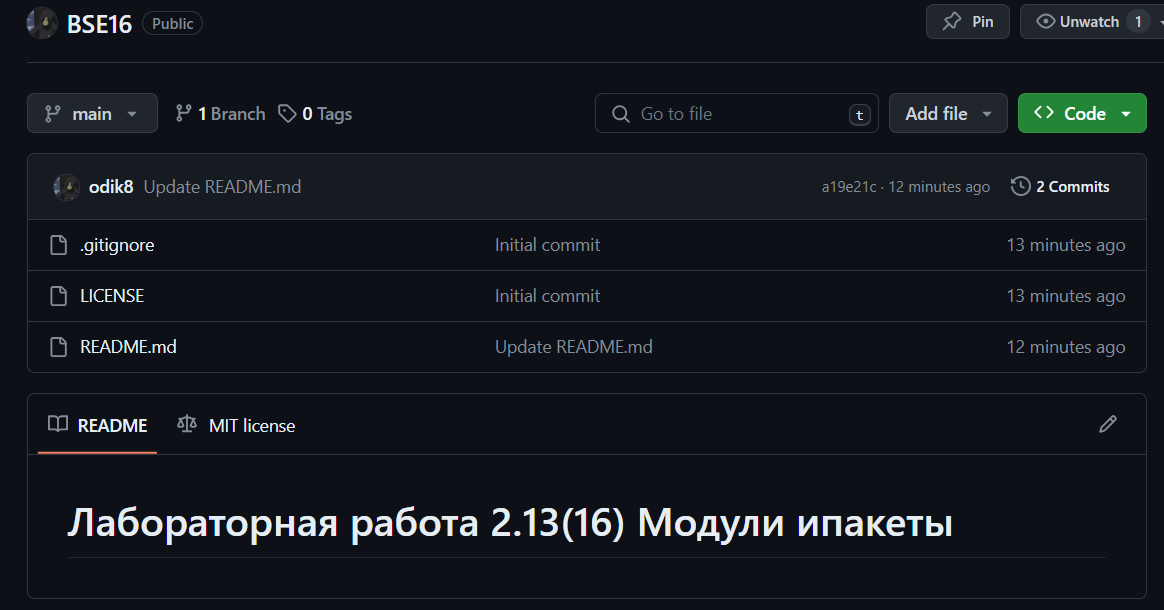


Рисунок 1 – Созданный репозиторий

3. Выполнено клонирование созданного репозитория.

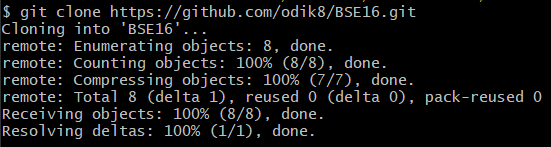


Рисунок 2 – Клонирование репозитория

4. Дополнен файл .gitignore необходимыми правилами для работы с IDE PyCharm.

5. Организован репозиторий в соответствие с моделью ветвления git-flow.

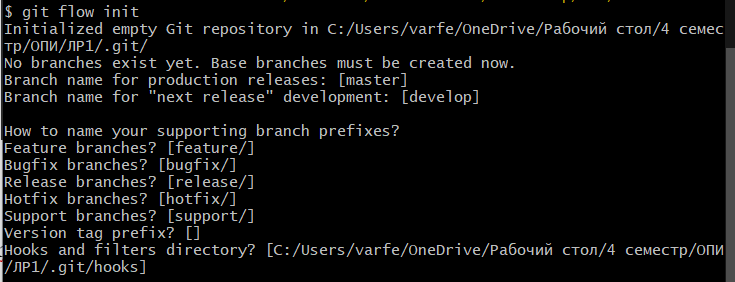


Рисунок 3 – Инициализация git-flow

6. Создан проект PyCharm в папке репозитория.

7. Выполнены индивидуальные задания.

Задание 1

Выполнить индивидуальное задание лабораторной работы 2.11, оформив все функции программы в виде отдельного модуля. Разработанный модуль должен быть подключен в основную программу с помощью одного из вариантов команды import. Номер варианта уточнить у преподавателя

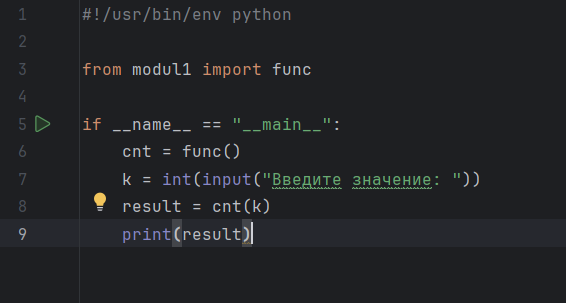


Рисунок 4 – Главный код

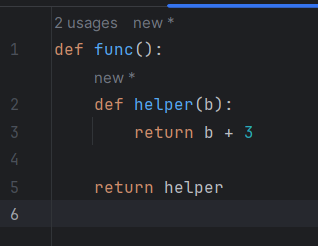


Рисунок 5 – Код модуля

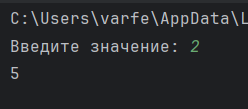


Рисунок 6 – Результат выполнения программы

Задание 2. Выполнить индивидуальное задание лабораторной работы 2.8, оформив все классы программы в виде отдельного пакета. Разработанный пакет должен быть подключен в основную программу с помощью одного из вариантов команды import . Настроить соответствующим образом переменную \_\_all\_\_ в файле \_\_init\_\_.py пакета. Номер варианта уточнить у преподавателя.

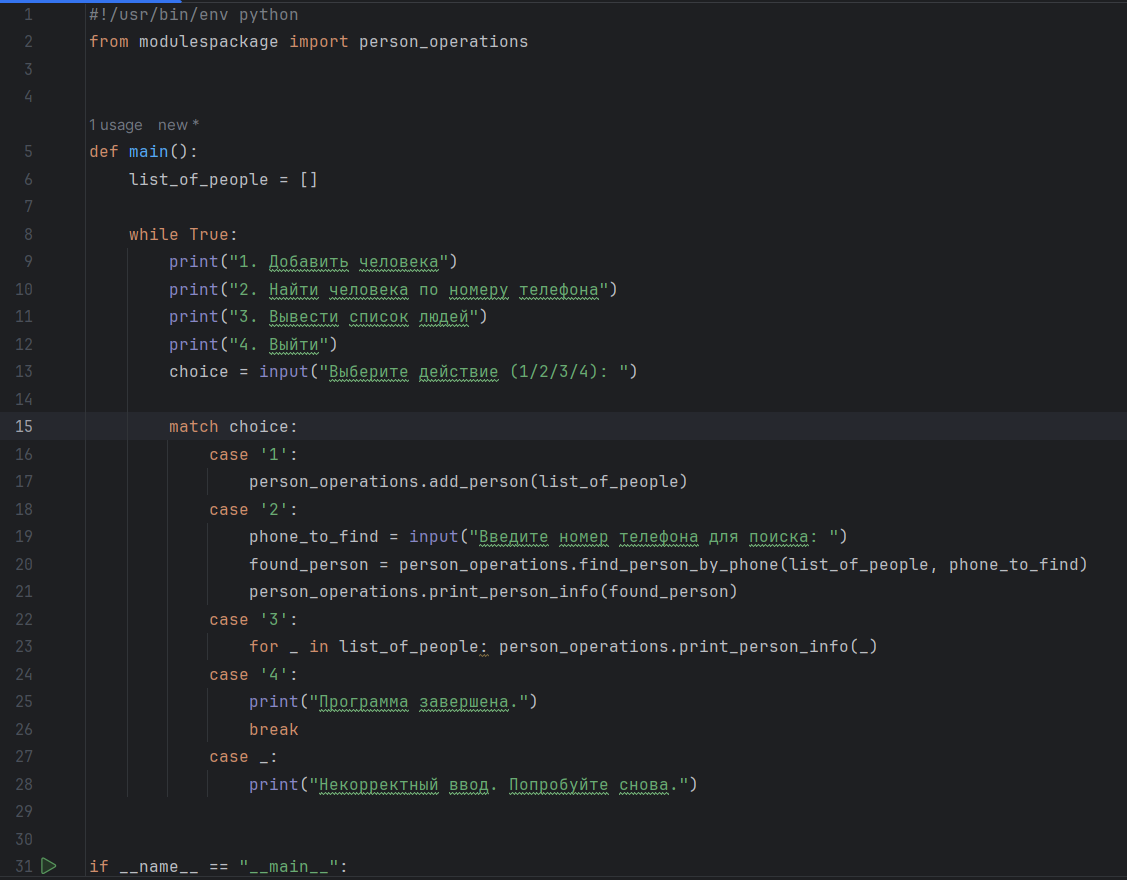


Рисунок 7 – Главный код программы

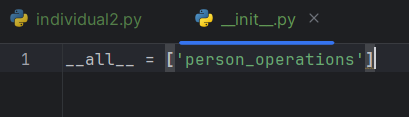


Рисунок 8 – Файл \_\_init\_\_.py

Код вспомогательного модуля

from datetime import datetime  
  
  
def get\_birthdate():  
 while True:  
 try:  
 date\_str = input("Введите дату рождения в формате ДД.ММ.ГГГГ: ")  
 birthdate = datetime.strptime(date\_str, "%d.%m.%Y").date()  
 return birthdate  
 except ValueError:  
 print("Ошибка Неправильный формат даты. Попробуйте снова.")  
  
  
def add\_person(list\_of\_people):  
 print("\nДобавление нового человека:")  
 last\_name = input("Введите фамилию: ")  
 first\_name = input("Введите имя: ")  
 phone\_number = input("Введите номер телефона: ")  
 birthdate = get\_birthdate()  
  
 person = {  
 'фамилия': last\_name,  
 'имя': first\_name,  
 'номер телефона': phone\_number,  
 'дата рождения': birthdate  
 }  
  
 list\_of\_people.append(person)  
 list\_of\_people.sort(key=lambda x: x['дата рождения'])  
 print("Человек добавлен\n")  
  
  
def find\_person\_by\_phone(people, phone):  
 for person in people:  
 match person['номер телефона']:  
 case phone:  
 return person  
 return None  
  
  
def print\_person\_info(list\_of\_people):  
 match list\_of\_people:  
 case []:  
 print("Человек не найден.\n")  
 case {'фамилия': f, 'имя': i, 'номер телефона': nt, 'дата рождения': dr}:  
 print("\nИнформация о человеке:")  
 print(f"Фамилия: {f}")  
 print(f"Имя: {i}")  
 print(f"Номер телефона: {nt}")  
 print(f"Дата рождения: {dr.strftime('%d.%m.%Y')}\n")

8. Зафиксированы сделанные изменения в репозитории.

9. Добавлен отчет по лабораторной работе в формате PDF в папку doc репозитория.

10. Выполнено слияние ветки для разработки с веткой master/main.

11. Отправлены сделанные изменения на сервер GitHub.

12. Отправлен адрес репозитория GitHub на электронный адрес преподавателя.

Контрольные вопросы:

1. Что является модулем языка Python? – Под модулем в Python понимается файл с расширением .py. Модули предназначены для того, чтобы в них хранить часто используемые функции, классы, константы и т. п. Можно условно разделить модули и программы: программы предназначены для непосредственного запуска, а модули для импортирования их в другие программы.

2. Какие существуют способы подключения модулей в языке Python? – Самый простой способ импортировать модуль в Python это воспользоваться конструкцией:

import имя\_модуля

Если вы хотите задать псевдоним для модуля в вашей программе, можно воспользоваться вот таким синтаксисом:

import имя\_модуля as новое\_имя

Используя любой из вышеперечисленных подходов, при вызове функции из импортированного модуля, вам всегда придется указывать имя модуля (или псевдоним). Для того, чтобы этого избежать делайте импорт через конструкцию from имя\_модуля import имя\_объекта

3. Что является пакетом языка Python? – Пакет в Python – это каталог, включающий в себя другие каталоги и модули, но при этом дополнительно содержащий файл \_\_init\_\_.py .

4. Каково назначение файла \_\_init\_\_.py ? – пустой файл для того, чтобы Python распознавал папку как пакет и позволяет нам использовать его объекты внутри других частей проекта.

5. Каково назначение переменной \_\_all\_\_ файла \_\_init\_\_.py ? – Файл \_\_init\_\_.py может быть пустым или может содержать переменную \_\_all\_\_ , хранящую список модулей, который импортируется при загрузке через конструкцию:

from имя\_пакета import \*