

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 \_\_\_\_\_

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.

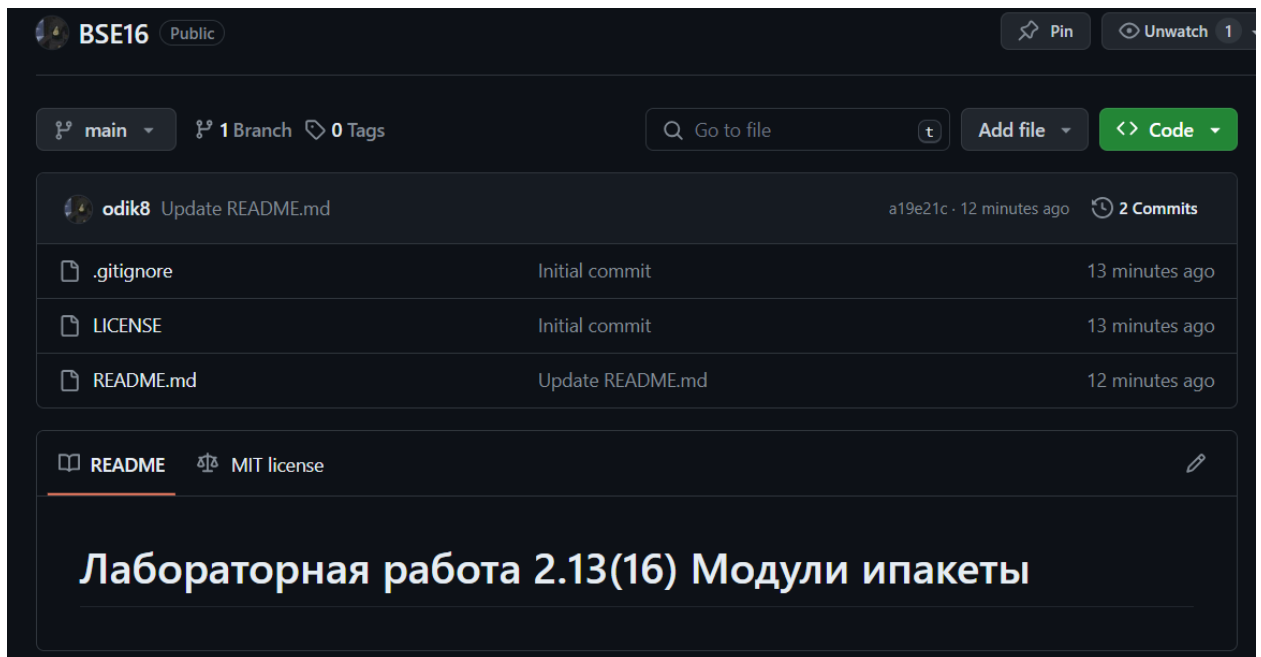


Рисунок 1 – Созданный репозиторий

3. Выполнено клонирование созданного репозитория.

```
$ git clone https://github.com/odik8/BSE16.git
Cloning into 'BSE16'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (8/8), done.
Resolving deltas: 100% (1/1), done.
```

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.

```

$ git flow init
Initialized empty Git repository in C:/Users/varfe/OneDrive/Рабочий стол/4 семес
тр/ОПИ/ЛР1/.git/
No branches exist yet. Base branches must be created now.
Branch name for production releases: [master]
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/varfe/OneDrive/Рабочий стол/4 семестр/ОПИ
/ЛР1/.git/hooks]

```

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

```

1      #!/usr/bin/env python
2
3      from modul1 import func
4
5  ▶    if __name__ == "__main__":
6         cnt = func()
7         k = int(input("Введите значение: "))
8         = result = cnt(k)
9         print(result)

```

Рисунок 4 – Главный код

```

2 usages new *
1 def func():
    new *
2     def helper(b):
3         return b + 3
4
5     return helper
6

```

Рисунок 5 – Код модуля

```

C:\Users\varfe\AppData\Local\
Введите значение: 2
5

```

Рисунок 6 – Результат выполнения программы

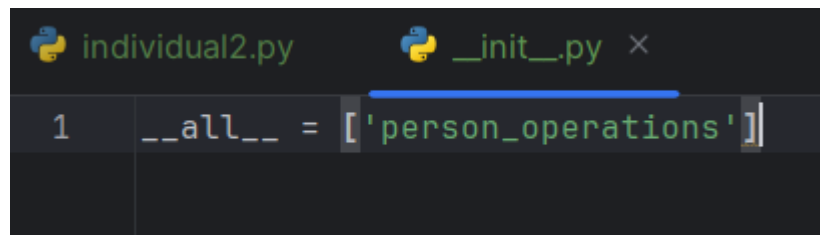
Задание 2. Выполнить индивидуальное задание лабораторной работы 2.8, оформив все классы программы в виде отдельного пакета. Разработанный пакет должен быть подключен в основную программу с помощью одного из вариантов команды `import`. Настроить соответствующим образом переменную `__all__` в файле `__init__.py`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

```

1  #!/usr/bin/env python
2  from modulespackage import person_operations
3
4
5  usage new *
6  def main():
7      list_of_people = []
8
9      while True:
10         print("1. Добавить человека")
11         print("2. Найти человека по номеру телефона")
12         print("3. Вывести список людей")
13         print("4. Выйти")
14         choice = input("Выберите действие (1/2/3/4): ")
15
16         match choice:
17             case '1':
18                 person_operations.add_person(list_of_people)
19             case '2':
20                 phone_to_find = input("Введите номер телефона для поиска: ")
21                 found_person = person_operations.find_person_by_phone(list_of_people, phone_to_find)
22                 person_operations.print_person_info(found_person)
23             case '3':
24                 for _ in list_of_people: person_operations.print_person_info(_)
25             case '4':
26                 print("Программа завершена.")
27                 break
28             case _:
29                 print("Некорректный ввод. Попробуйте снова.")
30
31  if __name__ == "__main__":

```

Рисунок 7 – Главный код программы



```

individual2.py  __init__.py x
1  __all__ = ['person_operations']

```

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 *
```