

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:  
Матвеев Александр Иванович  
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 Воронкин Роман Александрович

\_\_\_\_\_  
(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 версии 3.x.

Ход работы.

1. Создание нового репозитория с лицензией MIT.

**Create a new repository**

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

*Required fields are marked with an asterisk (\*).*

**Owner \*** **Repository name \***

SashkaHacker / laba10

✔ laba10 is available.

Great repository names are short and memorable. Need inspiration? How about [fluffy-guide](#) ?

**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:**

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

**Add .gitignore**

.gitignore template: **Python**

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

**Choose a license**

License: **MIT License**

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

You are creating a public repository in your personal account.

**Create repository**

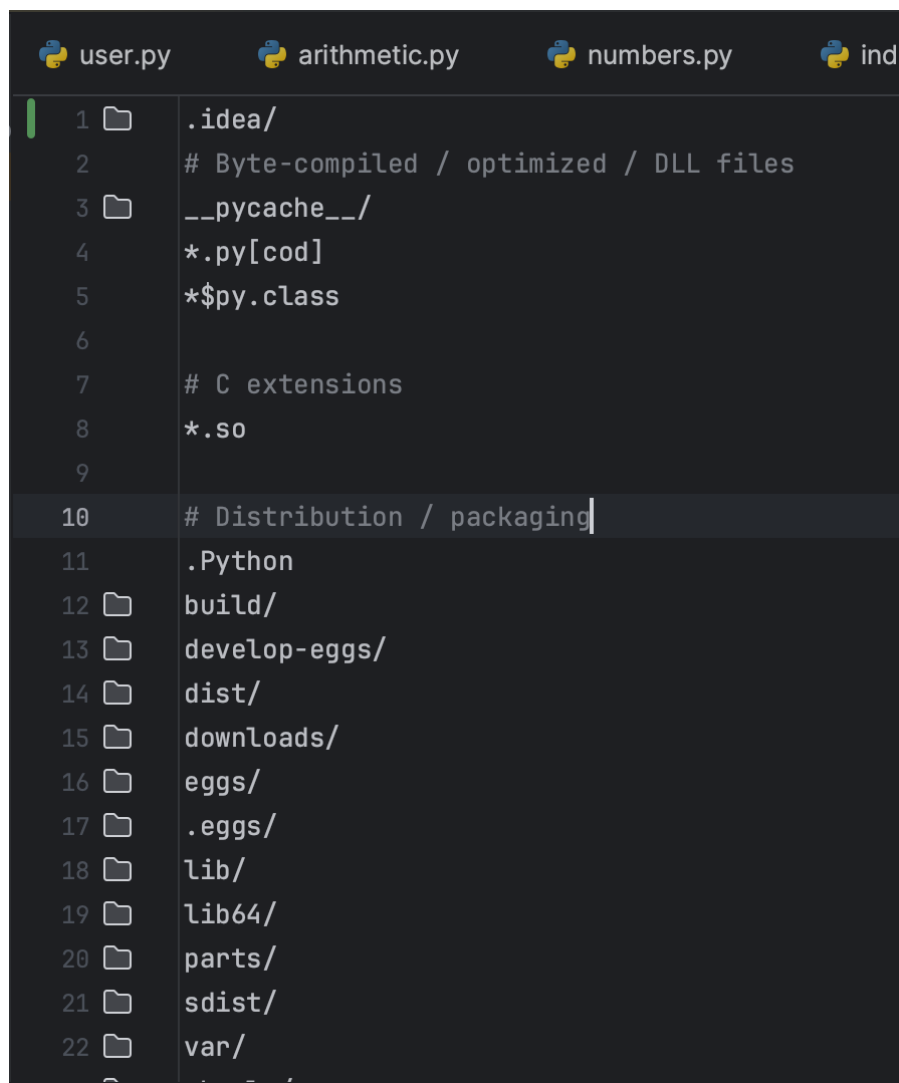
Рисунок 1 – Создание репозитория

2. Клонировал репозиторий на рабочий ПК.

```
[→ GitHub git clone https://github.com/SashkaHacker/laba16.git
Cloning into 'laba16'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
```

Рисунок 2 – Клонирование репозитория

3. Дополнил файл .gitignore необходимыми инструкциями.



```
user.py arithmetic.py numbers.py ind
1  .idea/
2  # Byte-compiled / optimized / DLL files
3  __pycache__/
4  *.py[cod]
5  *$py.class
6
7  # C extensions
8  *.so
9
10 # Distribution / packaging
11 .Python
12 build/
13 develop-eggs/
14 dist/
15 downloads/
16 eggs/
17 .eggs/
18 lib/
19 lib64/
20 parts/
21 sdist/
22 var/
```

Рисунок 3 – Файл .gitignore

4. Выполнил индивидуальное задание лабораторной работы 2.11, оформил функции программы в виде отдельного модуля. Подключил разработанный модуль в основную программу с помощью команды `import`.

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

def func():
    def inner(k):
        return k + 3
    return inner
```

Рисунок 4 – Модуль module1.py

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

import module1

if __name__ == '__main__':
    cnt = module1.func()
    print(cnt(int(input())))
    print(cnt(int(input())))
```

Рисунок 5 – Файл individual1.py

5. Выполнил индивидуальное задание лабораторной работы 2.8, оформив все классы программы в виде отдельного пакета. Разработанный пакет подключил в основную программу с помощью команды `import`. Настроил переменную `__all__` в файле `__init__.py` пакета.

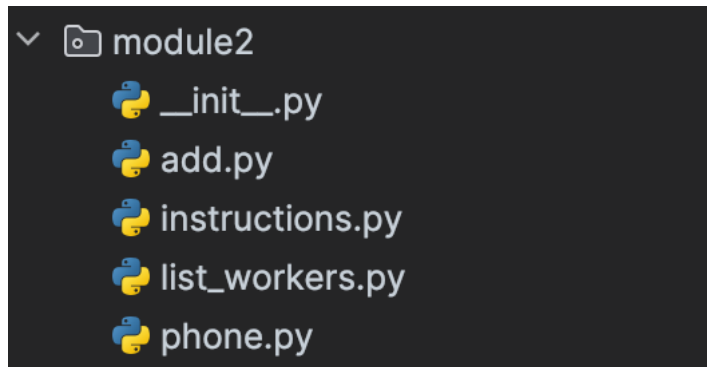


Рисунок 6 – Структура пакета module2

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

from module2 import *

__all__ = ['add', 'instructions', 'list_workers', 'phone']
```

Рисунок 7 – Файл \_\_init\_\_.py

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

import sys
from module2 import *

def main():
    lst = []

    # ВВОД ДАННЫХ
    while True:
        # Запросить команду из терминала.
        command = input(">>> ").lower()

        # Выполнить действие в соответствии с командой.
        match command:
            case 'exit':
                break
            case 'add':
                add.add(lst)
            case 'phone':
                phone.phone(lst)
            case 'help':
                instructions.instruction()
            case 'list':
                list_workers.workers(lst)
            case _:
                print(f"Неизвестная команда {command}", file=sys.stderr)

if __name__ == "__main__":
    main()
```

Рисунок 8 – Файл individual2.py

Контрольные вопросы:

1. В: Что является модулем языка Python?

О: 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.

2. В: Какие существуют способы подключения модулей в языке Python?

О: Импорт полного модуля: Используя инструкцию `import`, например `import math`, после чего вы можете использовать объекты модуля с префиксом `math.`, например `math.sin()`.

Импорт отдельных объектов из модуля: Можно импортировать конкретные функции, классы или переменные, используя `from ... import ...`, например `from math import sin, cos`, после чего вы можете обращаться к `sin` и `cos` напрямую.

Импорт всех объектов из модуля: Используя `from module import *`, вы импортируете все публичные имена из модуля, что не рекомендуется из-за возможных конфликтов имен.

Импорт с псевдонимом: Вы можете использовать `import module as alias`, например `import numpy as np`, что позволяет вам использовать псевдоним `np` для доступа к объектам модуля.

3. В: Что является пакетом языка Python?

О: Пакет в Python – это каталог, включающий в себя другие каталоги и модули, но при этом дополнительно содержащий файл `__init__.py`. Пакеты используются для формирования пространства имен, что позволяет работать с модулями через указание уровня вложенности (через точку).

4. В: Каково назначение файла `__init__.py` ?

О: Он указывает, что директория содержит Python пакет, и позволяет импортировать модули из этой директории.

5. В: Каково назначение переменной `__all__` файла `__init__.py` ?

О: `__all__` определяет список имён модулей или объектов, которые будут импортированы при использовании `from package import *`.