

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:
Плугатырев Владислав Алексеевич
2 курс, группа ПИЖ-б-о-22-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка и
сопровождение программного
обеспечения», очная форма обучения

(подпись)

Доцент кафедры инфокоммуникаций
Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: Модули и пакеты в языке Python.

Цель работы: приобретение навыков по работе с модулями и пакетами при написании программ с помощью языка программирования Python версии 3.x.

Порядок выполнения работы

1. Создал репозиторий.

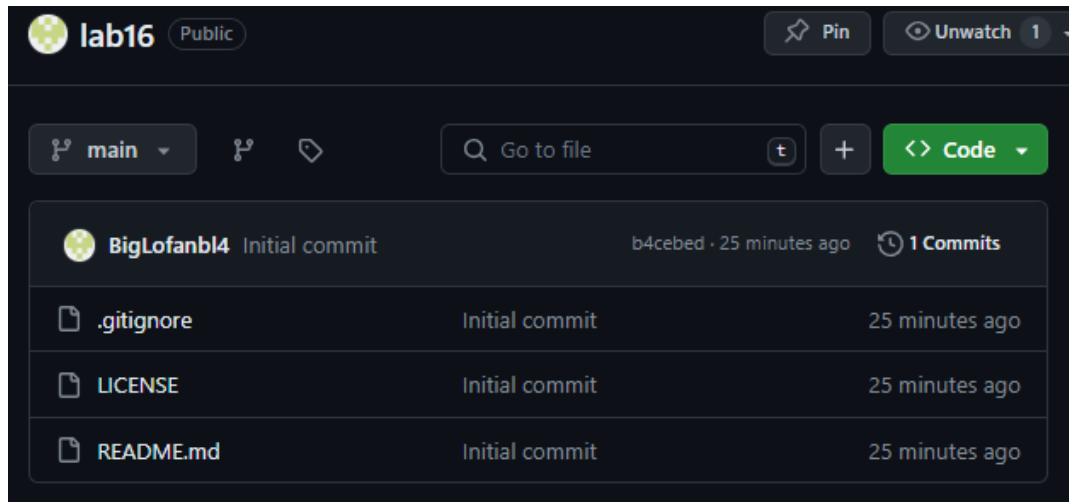


Рисунок 1.1 – Созданные репозиторий

2. Выполнил индивидуальное задание лабораторной работы 2.11, оформив функции программы в виде отдельного модуля. Подключил созданный модуль.

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

def func():
    def form_str(lst):
        res = "<ol>\n"
        for i in lst:
            res += f"<li>{i}</li>\n"
        res += "</ol>"
        return res

    return form_str
```

Рисунок 2.1 – Код созданного модуля

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

import module1 as md

if __name__ == "__main__":
    lst = ["str1", "str2", "str3"]
    a = md.func()
    print(a(lst))
```

Рисунок 2.2 – Подключение и использование модуля

3. Выполнил индивидуальное задание лабораторной работы 2.8, оформив все классы программы в виде отдельного пакета. Настроил переменную «__all__» в файле «__init__.py» пакета.

```
managment > display_people.py > show
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  def show(people):
5      if people:
6          line = "+-{}-+-{}-+-{}-+-{}-+-{}-+ ".format(
7              "-" * 4, "-" * 30, "-" * 30, "-" * 20, "-" * 20
8          )
9          print(line)
10         print(
11             "| {:^4} | {:^30} | {:^30} | {:^20} | {:^20} | ".format(
12                 "№", "Фамилия", "Имя", "Знак зодиака", "Дата рождения"
13             )
14         )
15         print(line)
16
17         for idx, person in enumerate(people, 1):
18             print(
19                 "| {:>4} | {:<30} | {:<30} | {:<20} | {:>20} | ".format(
20                     idx,
21                     person.get("surname", ""),
22                     person.get("name", ""),
23                     person.get("zodiac", ""),
24                     " ".join(person.get("birthday", "")),
25                 )
26             )
27             print(line)
28     else:
29         print("Список пуст")
30
```

Рисунок 3.1 – Код модуля «display_people.py»

```
get_instructions.py U X
managment > get_instructions.py > help
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  def help():
5      print("add - добавление нового человека")
6      print("info - данные о человеке по его фамилии")
7      print("exti - завершение программы")
8      print("list - вывод информации о всех людях")
```

Рисунок 3.2 – Код модуля «get_instructions.py»

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

def create():
    person = {}
    person["surname"] = input("Введите фамилию: ")
    person["name"] = input("Введите имя: ")
    person["zodiac"] = input("Введите знак зодиака: ")
    person["birthday"] = input("Дата рождения (число.месяц.год):").split(".")
    return person
```

Рисунок 3.3 – Код модуля «get_person.py»

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

def sel(surname, people):
    result = []
    for i in people:
        if i.get("surname", "") == surname:
            result.append(i)
    return result
```

Рисунок 3.4 – Код модуля «select_people.py»

```
__init__.py U X
managment > __init__.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  __all__ = ["display_people", "get_instructions", "get_person", "select_people"]
5
```

Рисунок 3.5 – Настройка переменной «__all__»

```
2.py U X
2.py > main
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5  from managment import *
6  from datetime import datetime
7
8  def main():
9      people = []
10
11      while True:
12          command = input("Введите команду (add, info, list, exit, help): ").strip().lower()
13
14          match command:
15              case "exit":
16                  break
17
18              case "add":
19                  person = get_person.create()
20                  people.append(person)
21                  people.sort(
22                      key=lambda x: datetime.strptime(".".join(x["birthday"]), "%d.%m.%Y")
23                  )
24
25              case "info":
26                  surname = input("Введите фамилию: ")
27                  selected = select_people.sel(surname, people)
28                  display_people(selected)
29
30              case "list":
31                  display_people.show(people)
32
33              case "help":
34                  get_instructions.help()
35
36              case _:
37                  print(f"Неизвестная команда {command}", file=sys.stderr)
38
39  if __name__ == "__main__":
40      main()
```

Рисунок 3.6 – Импорт пакета, код главного файла

Ответы на контрольные вопросы

1. Модуль в языке Python представляет собой файл с расширением .py, содержащий Python код. Этот код может содержать определения функций, классов и переменных, а также выполнимые инструкции.
2. Существует несколько способов подключения модулей в Python:
3. Использование оператора import: `import module_name`
4. Использование оператора from: `from module_name import something`
5. Использование оператора import с ключевым словом as: `import module_name as alias`
6. Пакет в языке Python представляет собой каталог, содержащий модули и/или другие пакеты, а также специальный файл `__init__.py`. Пакет позволяет организовать и структурировать большие проекты на Python.
7. Файл `__init__.py` в пакете Python выполняет роль инициализации пакета. Этот файл может быть пустым или содержать инициализационный код, который будет выполнен при импорте пакета.
8. Переменная `__all__` в файле `__init__.py` используется для определения списка модулей, которые будут импортированы при импорте всего пакета с помощью оператора `from package_name import *`. Это позволяет контролировать, какие именно модули будут доступны при использовании данного синтаксиса импорта.