

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра информационных систем и технологий

Отчет по лабораторной работе №16.

Дисциплина: «Основы программной инженерии»

Выполнил:

Студент группы ПИЖ-б-о-22-1,

направление подготовки: 09.03.04

«Программная инженерия»

ФИО: Рядская Мария Александровна

Проверил:

Воронкин Р. А.

Ставрополь 2024

Тема: Лабораторная работа 2.13 Модули и пакеты.

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


Выполнение работы:

1. Изучила теоретический материал работы.
2. Создала репозиторий на git.hub.

Создайте новый репозиторий



Репозиторий содержит все файлы проекта, включая историю изменений. У вас уже есть репозиторий проекта в другом месте? [Импортируйте репозиторий.](#)

Обязательные поля отмечены звездочкой (*).

Владелец *	Название репозитория *
 мрядская ▾	/ lab16
	✓ lab1.

Отличные названия репозитория короткие и запоминающиеся. Нужно вдохновение? Как насчет [fictional-waddle?](#)

Описание (необязательно)

- ☒  **Общедоступный**
Этот репозиторий может видеть любой пользователь Интернета. Вы сами выбираете, кто может совершать коммиты.
- ☐  **Приватный**
Вы сами выбираете, кто может просматривать этот репозиторий и фиксировать его в нем.

Инициализируйте этот репозиторий с помощью:

- ☒ **Добавьте файл README**
Здесь вы можете написать подробное описание вашего проекта. [Узнайте больше о README.](#)

Добавить файл .gitignore

.шаблон gitignore: Отсутствует ▾

Рисунок 1 – создание репозитория

3. Клонировала репозиторий.

```
C:\>cd git1

C:\git1>git clone https://github.com/mryadskaya/lab16.git
Cloning into 'lab16'...
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.

C:\git1>cd lab16

C:\git1\lab16>
```

Рисунок 2 – клонирование репозитория

Дополнить файл gitignore необходимыми правилами.

```
76     target/
77
78     # Jupyter Notebook
79     .ipynb_checkpoints
80
81     # IPython
82     profile_default/
83     ipython_config.py
84
85     # pyenv
86     #   For a library or package, you might want to ignore these files since the code is
87     #   intended to run in multiple environments; otherwise, check them in:
88     # .python-version
89
90     # pipenv
91     #   According to pya/pipenv#598, it is recommended to include Pipfile.lock in version control.
92     #   However, in case of collaboration, if having platform-specific dependencies or dependencies
93     #   having no cross-platform support, pipenv may install dependencies that don't work, or not
94     #   install all needed dependencies.
95     #Pipfile.lock
96
```

Рисунок 3 – .gitignore для IDE PyCharm

4.Организовать свой репозиторий в соответствии с моделью ветвления git-flow.

```
C:\git1>cd lab16

C:\git1\lab16>git checkout -b develop
Switched to a new branch 'develop'

C:\git1\lab16>
```

Рисунок 4 – создание ветки develop

5. 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.

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

import zad

if __name__ == "__main__":
    n = input("Введите вашу фамилию: ")
    l = input("Введите ваше имя: ")

    # Создаем замыкание с шаблоном
    greeting_template = fun.create_greeting_template("Уважаемый %F%, %N%! Вы делаете работу по замыканиям функций.")

    # Вызываем внутреннюю функцию замыкания и отображаем результат
    result = greeting_template(n, l)
    print(result)
```

задание 1,1

⋮

\\Users\\ADMIN\\PycharmProjects\\pythonProject\\venv\\Scripts\\python.exe "C:\\git1\\lab11\\PyCharm\\задание1\\задание 1,1.py"

Введите вашу фамилию: |

Рисунок 5 – главная часть программы



```
Project Alt+1 zad.py × задание 1,1.py
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 def create_greeting_template(template):
5
6     def inner_function(last_name, first_name):
7         formatted_template = template.replace('%F%', last_name).replace('%N%', first_name)
8         return formatted_template
9
10    return inner_function
```

Рисунок 6 – модуль

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

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

import sys
from MY import *

if __name__ == '__main__':
    help1.help1()
    # Список работников.
    workers = []
    # Организовать бесконечный цикл запроса команд.
    while True:
        # Запросить команду из терминала.
        command = input(">>> ").lower()
        # Выполнить действие в соответствие с командой.

        match command:
            case 'exit':
                break

            case 'add':
                # Добавить словарь в список.
                i = add1.add1()
                workers.append(i)

                # Отсортировать список в случае необходимости.
                if len(workers) > 1:
                    workers.sort(key=lambda item: item.get('name', ''))
```

Рисунок 7 – файл задание2








Имя	Д
 __pycache__	0
 __init__	0
 add1	0
 error1	0
 help1	0
 list	0
 select	0

Рисунок 8 – Структура

```
d.py < задание2.py < __init__.py < add.py < error.p
__all__ = ["add1", "error1", "help1", 'list', 'select']
```

Рисунок 9 – Файл __init__.py

```
Список команд:

add - добавить информацию;
list - вывести список ;
select <тип> - вывод на экран фамилия, имя; знак Зодиака; дата рождения
help - отобразить справку;
exit - завершить работу с программой.

>>> add
Фамилия и инициалы? ryadskaya m
знак зодиака? rak
Год рождения? 2005
>>> add
Фамилия и инициалы? lkjhgf
знак зодиака? kjh
Год рождения? 852
>>> add
Фамилия и инициалы? dfg
знак зодиака? er
Год рождения? 799
>>> list
+-----+-----+-----+-----+
| 1 | dfg | er | 799 |
| 2 | lkjhgf | kjh | 852 |
| 3 | ryadskaya m | rak | 2005 |
+-----+-----+-----+-----+
```

Рисунок 10 – результат работы программы

7.Зафиксировала все изменения в github в ветке develop.

```

C:\git1\lab16>git add .
C:\git1\lab16>git commit -m"сохраненные"
[develop 759794d] сохраненные
9 files changed, 154 insertions(+)
create mode 100644 "PyCharm\320\267\320\260\320\264\320\260\320\275\320\270\320\2651\zad.py"
create mode 100644 "PyCharm\320\267\320\260\320\264\320\260\320\275\320\270\320\2651\320\267\320\260\320\264\320\260\320\275\320\270\320\265 1,1.py"
create mode 100644 "PyCharm\320\267\320\260\320\264\320\260\320\275\320\270\320\2652\MY\__init__.py"
create mode 100644 "PyCharm\320\267\320\260\320\264\320\260\320\275\320\270\320\2652\MY\add1.py"
create mode 100644 "PyCharm\320\267\320\260\320\264\320\260\320\275\320\270\320\2652\MY\error1.py"
create mode 100644 "PyCharm\320\267\320\260\320\264\320\260\320\275\320\270\320\2652\MY\help1.py"
create mode 100644 "PyCharm\320\267\320\260\320\264\320\260\320\275\320\270\320\2652\MY\list.py"
create mode 100644 "PyCharm\320\267\320\260\320\264\320\260\320\275\320\270\320\2652\MY\select.py"
create mode 100644 "PyCharm\320\267\320\260\320\264\320\260\320\275\320\270\320\2652\320\267\320\260\320\264\320\260\320\275\320\270\320\2652.py"
C:\git1\lab16>git push origin develop
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 2 threads
Compressing objects: 100% (15/15), done.
Writing objects: 100% (15/15), 3.60 KiB | 368.00 KiB/s, done.
Total 15 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:      https://github.com/mryadskaya/lab16/pull/new/develop
remote:
To https://github.com/mryadskaya/lab16.git
 * [new branch]      develop -> develop
C:\git1\lab16>

```

Рисунок 8 – фиксация изменений в ветку develop

8. Слила ветки.

```

C:\git1\lab16>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

C:\git1\lab16>git merge develop
merge: develop - not something we can merge

C:\git1\lab16>git merge develop
Updating f304aeb..759794d
Fast-forward
.../zad.py" | 10 ++++++
...320\264\320\260\320\275\320\270\320\265 1,1.py" | 15 ++++++++
.../MY/__init__.py" | 1 +
.../MY/add1.py" | 15 ++++++++
.../MY/error1.py" | 8 +++++
.../MY/help1.py" | 14 ++++++
.../MY/list.py" | 27 ++++++++
.../MY/select.py" | 23 ++++++++
...60\320\264\320\260\320\275\320\270\320\2652.py" | 41 ++++++++
9 files changed, 154 insertions(+)
create mode 100644 "PyCharm\320\267\320\260\320\264\320\260\320\275\320\270\320\2651\zad.py"
create mode 100644 "PyCharm\320\267\320\260\320\264\320\260\320\275\320\270\320\2651\320\267\320\260\320\264\320\260\320\275\320\270\320\265 1,1.py"
create mode 100644 "PyCharm\320\267\320\260\320\264\320\260\320\275\320\270\320\2652\MY\__init__.py"
create mode 100644 "PyCharm\320\267\320\260\320\264\320\260\320\275\320\270\320\2652\MY\add1.py"
create mode 100644 "PyCharm\320\267\320\260\320\264\320\260\320\275\320\270\320\2652\MY\error1.py"
create mode 100644 "PyCharm\320\267\320\260\320\264\320\260\320\275\320\270\320\2652\MY\help1.py"
create mode 100644 "PyCharm\320\267\320\260\320\264\320\260\320\275\320\270\320\2652\MY\list.py"
create mode 100644 "PyCharm\320\267\320\260\320\264\320\260\320\275\320\270\320\2652\MY\select.py"
create mode 100644 "PyCharm\320\267\320\260\320\264\320\260\320\275\320\270\320\2652\320\267\320\260\320\264\320\260\320\275\320\270\320\2652.py"
C:\git1\lab16>

```

Рисунок 9 – сливание ветки develop в ветку main

Контрольные вопросы:

1 Что является модулем языка Python?

В Python модуль - это файл с расширением .py, содержащий код на языке Python. Модуль может содержать определения функций, классов,

переменных и другие инструкции, которые могут быть использованы в других модулях или скриптах.

2 Какие существуют способы подключения модулей в языке Python?

Импорт по имени модуля - `import module`

Импорт с переименованием - `import module as m`

Импорт конкретных элементов модуля – `from module_name import item1, item2`

Импорт всех элементов модуля - `from module_name import *`

3 Что является пакетом языка Python?

В Python пакет - это директория, которая содержит один или несколько модулей, а также файл `__init__.py`, который указывает Python, что эта директория является пакетом. Пакеты используются для организации и структурирования больших проектов на Python. Они позволяют логически группировать связанные модули вместе, что облегчает управление и поддержку кода.

4 Каково назначение файла `__init__.py` ?

Файл `__init__.py` может быть пустым или может содержать переменную `__all__`, хранящую список модулей, который импортируется при загрузке через конструкцию.

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

Переменная `__all__` в файле `__init__.py` в Python используется для определения списка символов, которые будут импортированы, когда мы используем конструкцию `from package_name import *`. Когда мы импортируем все

символы из пакета с помощью `from package_name import *`, Python импортирует

только те символы, которые перечислены в списке `__all__`.