

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 №1
дисциплины «Основы программной инженерии»

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

(подпись)

Руководитель практики:
Воронкин Р.А., кандидат технических
наук, доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

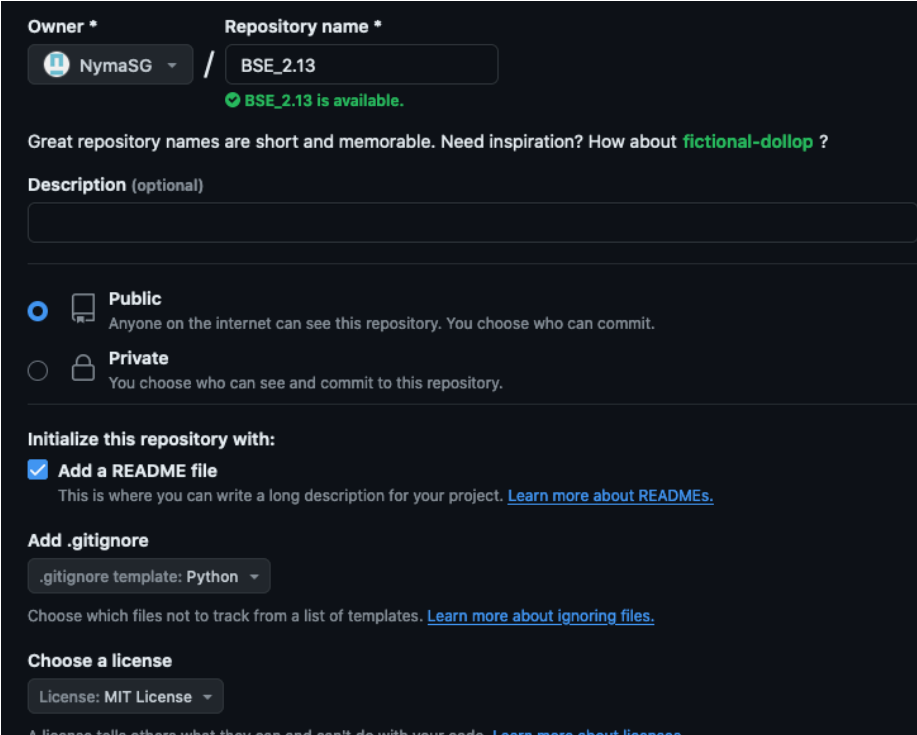
Ставрополь, 2023 г.

Тема: Модули и пакеты.

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

Методика выполнения

1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.



Owner * NymaSG / Repository name * BSE_2.13

Great repository names are short and memorable. Need inspiration? How about **fictional-dollop** ?

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.](#)

3. Выполните клонирование созданного репозитория.

```
[sultangoyaliev@MacBook-Pro-Sultan github % git clone https://github.com/NymaSG/BSE_2.13.git
Cloning into 'BSE_2.13'...
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
Unpacking objects: 100% (5/5), done.
sultangoyaliev@MacBook-Pro-Sultan github %
```

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

```
[sultangoyaliev@MacBook-Pro-Sultan bse_2.13 % git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
```

5. Выполните индивидуальные задания. Приведите в отчете скриншоты работы программ решения индивидуального задания.

5.1 Индивидуальное 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.

Основной файл.

```
individual_task_1.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  import module_1 as md
6
7  if __name__ == "__main__":
8      type_param = input("Введите тип (list/tuple): ")
9      numbers_string = input("Введите ваши переменные: ")
10
11     converter = md.create_converter(type_param)
12
13     result = converter(numbers_string)
14
15     print(result)
```

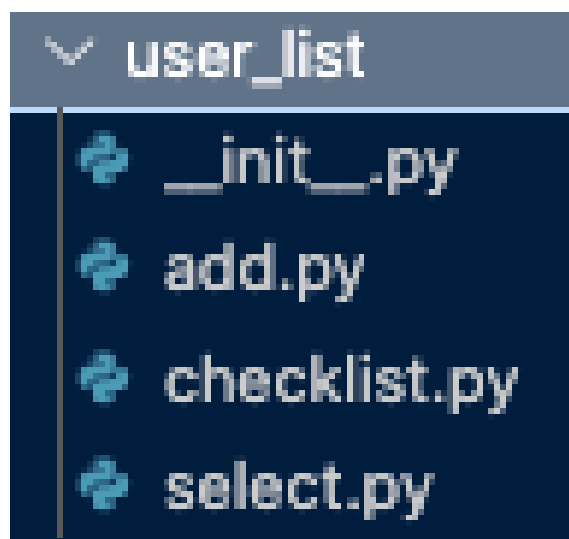
Модуль.

```
module_1.py > create_converter
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  def create_converter(type_param):
6      def converter(input_string):
7          numbers = list(map(int, input_string.split()))
8          if type_param == 'list':
9              return numbers
10         else:
11             return tuple(numbers)
12     return converter
```

5.2 Индивидуальное 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.

Реализовал пакет `user_list`.



Файл _init_.py

```
user_list > _init_.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  __all__ = ["add", "checklist", "select"]
```

Файл add.py

```
user_list > add.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  def add_users(users):
6      name = input("Введите Фамилию и Имя: ")
7      phone_number = input("Введите номер телефона: ")
8      year = list(map(int, input("Введите дату рождения (Пример: 20 07 2004): ").split()))
9
10     user = {
11         'name': name,
12         'phone_number': phone_number,
13         'year': year
14     }
15
16     users.append(user)
17
18     if len(users) > 1:
19         users.sort(key=lambda item: item.get('name', ''))
```

Файл checklist.py

```
user_list > checklist.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  def checklist_users(users):
6      line = '+-{}-+-{}-+-{}-+-{}+'.format('-' * 4, '-' * 20, '-' * 18, '-' * 10)
7      print(line)
8      print('| {:4} | {:20} | {:18} | {:10} |'.format(
9          '№',
10         'Название',
11         'Номер телефона',
12         'Дата'
13     ))
14     print(line)
15
16     for idx, user in enumerate(users, 1):
17         print('| {:4} | {:20} | {:18} | {:10} |'.format(
18             idx,
19             user['name'],
20             user['phone_number'],
21             ''.join(map(str, user['year']))
22         ))
23     print(line)
```

Файл select.py

```

user_list > select.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  def select_user(users):
5      print('Введите числов месяца (1-12): ')
6      while True:
7          num = int(input())
8          if num < 1 or num > 12:
9              print('Значение введено неправильно! Попробуйте еще раз.')
10             else:
11                 break
12
13     line = '+-{}--{}--{}--{}--+'.format('-' * 4, '-' * 20, '-' * 18, '-' * 10)
14     print(line)
15     print('| {:4} | {:20} | {:18} | {:10} |'.format(
16         '№',
17         'Название',
18         'Номер телефона',
19         'Дата'
20     ))
21     print(line)
22
23     for idx, user in enumerate(users, 1):
24         if user['year'][1] == num:
25             print('| {:4} | {:20} | {:18} | {:10} |'.format(
26                 idx,
27                 user['name'],
28                 user['phone_number'],
29                 ''.join(map(str, user['year']))
30             )
31         )
32         print(line)

```

Файл individual_task_2.py

```

1  #!/usr/bin/env python3
2  #-*- coding: utf-8 -*-
3
4  #Выполнить индивидуальное задание лабораторной работы 2.8, оформив все классы программы в виде отдельного пакета.
5  #Разработанный пакет должен быть подключен в основную программу с помощью одного из вариантов команды import .
6  #Настроить соответствующим образом переменную __all__ в файле __init__.py пакета.
7
8
9  from user_list import add, select, checklist
10
11
12  if __name__ == "__main__":
13      users = []
14
15      while True:
16          command = input("$ ").lower()
17
18          if command == 'exit':
19              break
20
21          elif command == 'add':
22              add.add_users(users)
23
24          elif command == 'select':
25              checklist.checklist_users(users)
26
27          elif command == 'list':
28              select.select_user(users)

```

6. Зафиксируйте сделанные изменения в репозитории.

```
[sultangoyaliev@MacBook-Pro-Sultan bse_2.13 % git add .
[sultangoyaliev@MacBook-Pro-Sultan bse_2.13 % git commit -m "added individual tasks"
[develop 3ff2d20] added individual tasks
7 files changed, 134 insertions(+)
create mode 100644 individual_task_1.py
create mode 100644 individual_task_2.py
create mode 100644 module_1.py
create mode 100644 user_list/__init__.py
create mode 100644 user_list/add.py
create mode 100644 user_list/checklist.py
create mode 100644 user_list/select.py
```

7. Добавьте отчет по лабораторной работе в формате PDF в папку doc репозитория. Зафиксируйте изменения.

8. Выполните слияние ветки для разработки с веткой master/main.

```
[sultangoyaliev@MacBook-Pro-Sultan bse_2.13 % git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
[sultangoyaliev@MacBook-Pro-Sultan bse_2.13 % git merge develop
Updating dfa4423..3ff2d20
Fast-forward
 individual_task_1.py | 15 ++++++
 individual_task_2.py | 28 ++++++
 module_1.py          | 12 ++++++
 user_list/__init__.py |  4 ++++
 user_list/add.py      | 19 ++++++
 user_list/checklist.py | 24 ++++++
 user_list/select.py   | 32 ++++++
7 files changed, 134 insertions(+)
create mode 100644 individual_task_1.py
create mode 100644 individual_task_2.py
create mode 100644 module_1.py
create mode 100644 user_list/__init__.py
create mode 100644 user_list/add.py
create mode 100644 user_list/checklist.py
create mode 100644 user_list/select.py
```

9. Отправьте сделанные изменения на сервер GitHub.

10. Отправьте адрес репозитория GitHub на электронный адрес преподавателя.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что является модулем языка Python?

В языке программирования Python модуль — это файл с расширением .py, содержащий код, который можно использовать в других программах.

2. Какие существуют способы подключения модулей в языке Python?

Существует несколько способов подключения модулей в Python:

- Использование ключевого слова `import` для загрузки всего модуля.
- Использование ключевого слова `from` для импорта конкретных объектов из модуля.
- Использование `import` с ключевым словом `as` для создания псевдонима при импорте.
- Использование оператора `from module import *` для импорта всех объектов из модуля.

3. Что является пакетом языка Python?

Пакет в Python — это специальная директория, содержащая модули и/или другие пакеты. Пакеты позволяют организовать пространство имен и структурировать код.

4. Каково назначение файла `_init_.py`?

Файл `_init_.py` внутри пакета используется для указания, что данная директория должна рассматриваться как пакет, а не просто как обычная директория. Он может также содержать инициализационный код, который выполняется при импорте пакета.

5. Каково назначение переменной `_all_` файла `_init_.py` ?

Переменная `_all_` в файле `_all_.py` используется для определения списка имен, которые будут доступны для импорта при использовании оператора `from package import *`. Это предоставляет контроль над тем, какие имена экспортируются из пакета.

