

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования «СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций
Институт цифрового развития

ОТЧЁТ
по лабораторной работе №2.13

Дисциплина: «Программирование на Python»

Тема: «Модули и пакеты»

Вариант 2

Выполнил: студент 2 курса,
группы ИВТ-б-о-21-1
Богдан Александр Анатольевич

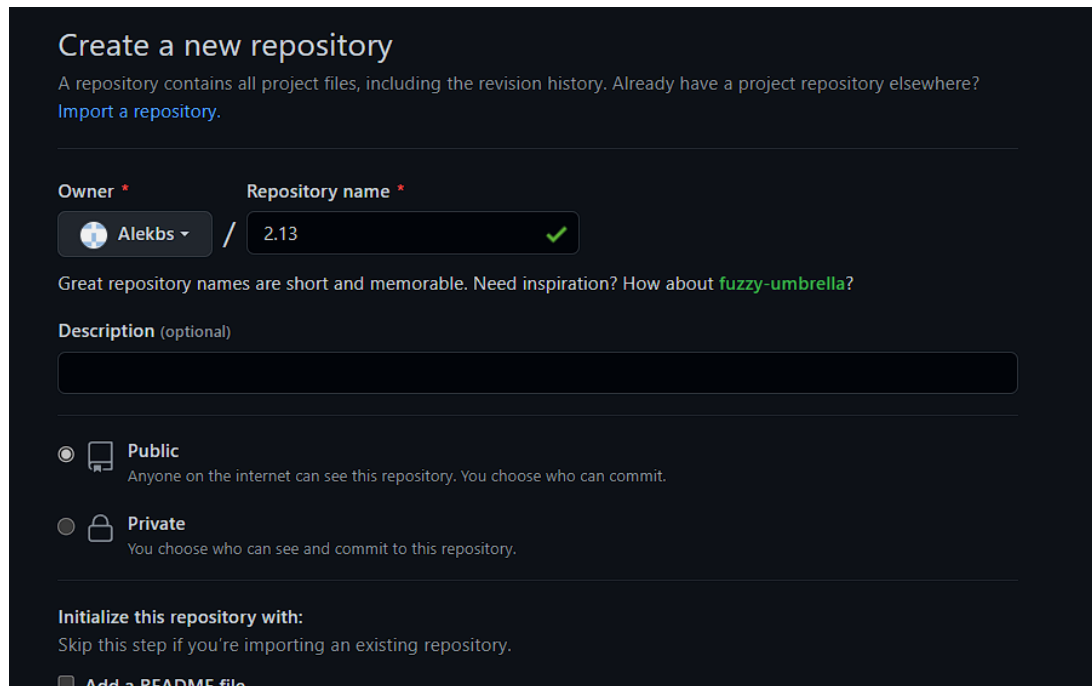
Ставрополь 2022

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

Практическая часть:

1. Создал репозиторий, клонировал его, изменил файл .gitignore.



Организовал репозиторий в соответствии с моделью ветвления git-flow



Create a new repository


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


Owner * Repository name *

 Aleks / 2.13 

Great repository names are short and memorable. Need inspiration? How about **fuzzy-umbrella**?

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:
Skip this step if you're importing an existing repository.

☐ Add a README file

Рисунок 1. Создание репозитория

```
# Created by https://www.toptal.com/developers/gitignore/api/pycharm,python
# Edit at https://www.toptal.com/developers/gitignore?templates=pycharm,python

### PyCharm ###
# Covers JetBrains IDEs: IntelliJ, RubyMine, PhpStorm, AppCode, PyCharm, CLion, Android Studio, WebStorm and Rider
# Reference: https://intellij-support.jetbrains.com/hc/en-us/articles/206544839

# User-specific stuff
.idea/**/workspace.xml
.idea/**/tasks.xml
.idea/**/usage.statistics.xml
.idea/**/dictionaries
.idea/**/shelf
.idea/
.idea

# AWS User-specific
.idea/**/aws.xml

# Generated files
.idea/**/contentModel.xml

# Sensitive or high-churn files
.idea/**/dataSources/
.idea/**/dataSources.ids
.idea/**/dataSources.local.xml
.idea/**/sqlDataSources.xml
.idea/**/dynamic.xml
.idea/**/uiDesigner.xml
.idea/**/dbnavigator.xml

# Gradle
.idea/**/gradle.xml
.idea/**/libraries

# Gradle and Maven with auto-import
```

Рисунок 3. Изменение файла .gitignore

```

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

Which branch should be used for integration of the "next release"?
- develop
Branch name for "next release" development: [develop] rel
Fatal: Local branch 'rel' does not exist.

D:\WAR\gi\2.13>git flow init

Which branch should be used for integration of the "next release"?
- develop
Branch name for "next release" development: [develop] develop

How to name your supporting branch prefixes?
Feature branches? [feature/] fea
Bugfix branches? [bugfix/] bug
Release branches? [release/] rel
Hotfix branches? [hotfix/] hot
Support branches? [support/] sup
Version tag prefix? [] pre
Hooks and filters directory? [D:/WAR/gi/2.13/.git/hooks] hook

```

Рисунок 4. Организация репозитория в соответствии с git-flow

2. Проработал примеры:

```

PS C:\Users\super> & C:/Users/super/AppData/Local/Programs/Python/Python39-0.9999987317275395
PS C:\Users\super>

```

Рисунок 5. Результат работы примера 1

```

PS C:\Users\super> & C:/Users/super/AppData/Local/Programs/Python/Python39-0.9999987317275395
PS C:\Users\super>

```

Рисунок 6. Результат работы примера 2

3. Выполнил индивидуальное задание:

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.

```

launcher 54705 -- C:\Users\super\Desktop\сф.
Результат работы функции:
<title>текст</title>
PS C:\Users\super\Desktop\Cfi\2.13\tasks>

```

Рисунок 7. Результат работы индивидуального задания 1

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.

```

>>> select
+-----+-----+-----+
|          Ф.И.О.          |      Группа      |   Оценки   |
+-----+-----+-----+
>>> add
Фамилия и инициалы? Петров П.П.
Группа? А
Введите оценки по 5 предметам через пробел: 4 5 5 5 4
>>> select
+-----+-----+-----+
|          Ф.И.О.          |      Группа      |   Оценки   |
+-----+-----+-----+
| Петров П.П.              | А                | 4,5,5,5,4 |
+-----+-----+-----+
>>> list
+-----+-----+-----+
|          Ф.И.О.          |      Группа      |   Оценки   |
+-----+-----+-----+
| Петров П.П.              | А                | 4,5,5,5,4 |
| Иванов И.И.              | А                | 3,5,4,3,5 |
+-----+-----+-----+

```

Рисунок 8. Результат работы индивидуального задания 2

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. Стоит заметить, что модули могут быть написаны не только на языке Python, но и на других языках (например С).

2. Какие существуют способы подключения модулей в языке Python?

За один раз можно импортировать сразу несколько модулей, для этого их нужно перечислить через запятую после слова `import`. Если вы хотите задать псевдоним для модуля в вашей программе, можно воспользоваться вот таким синтаксисом: `import имя_модуля import имя_модуля1, имя_модуля2`. 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 ... import`. Для импортирования нескольких функций из модуля, можно перечислить их имена через запятую. Импортируемому объекту можно задать псевдоним. `import имя_модуля as новое_имя`.

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`. Пакеты используются для формирования пространства имен, что позволяет работать с модулями через указание уровня вложенности (через точку).

4. Каково назначение файла `__init__.py` ?

Файл `__init__.py` нужен для объявления структуры пакета

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

В переменную `__all__` вносятся все модули пакета

Вывод: в результате выполнения работы были приобретены навыки по работе с модулями и пакетами языка программирования Python версии 3х.