

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 №2.11

Дисциплина: «Программирование на Python»

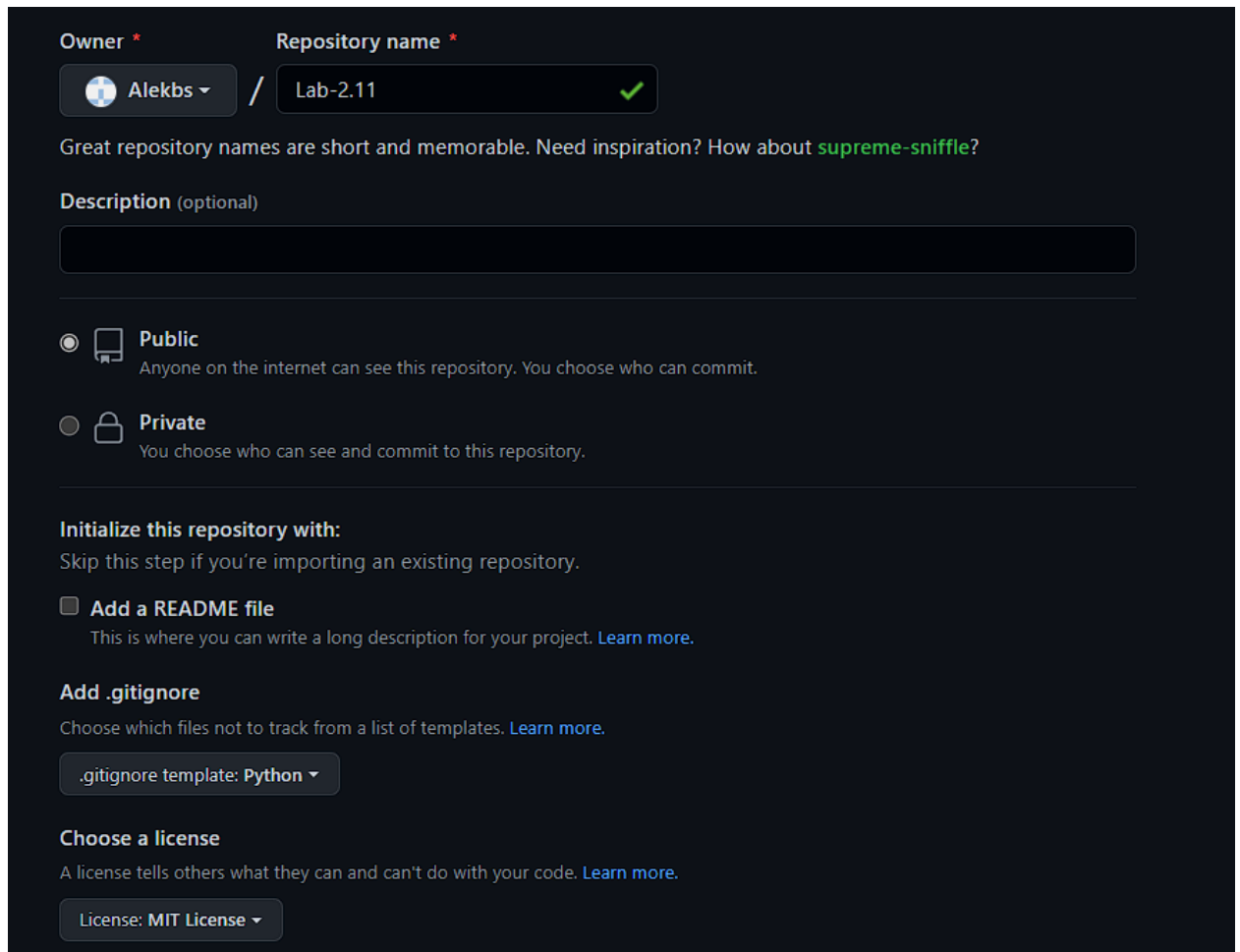
Тема: «Замыкания в языке Python»

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

Ставрополь 2022

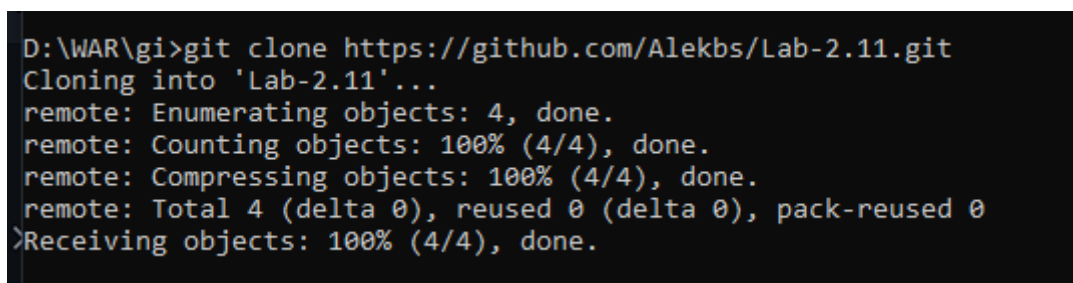
Цель работы: приобретение навыков по работе с замкнутыми при написании программ с помощью языка программирования Python версии 3.x.

Практическая часть:



The screenshot shows the GitHub 'Create new repository' page. At the top, there are two input fields: 'Owner' with a dropdown menu showing 'Aleks' and 'Repository name' with the text 'Lab-2.11' and a green checkmark. Below these is a hint: 'Great repository names are short and memorable. Need inspiration? How about [supreme-sniffle?](#)'. There is a 'Description (optional)' text area. Under the 'Visibility' section, 'Public' is selected with a radio button, and 'Private' is unselected. Below this is the 'Initialize this repository with:' section, which includes a checkbox for 'Add a README file' and a section for 'Add .gitignore' with a dropdown menu showing '.gitignore template: Python'. At the bottom, there is a 'Choose a license' section with a dropdown menu showing 'License: MIT License'.

Рисунок 1. Создание репозитория



```
D:\WAR\gi>git clone https://github.com/Alekbs/Lab-2.11.git
Cloning into 'Lab-2.11'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
>Receiving objects: 100% (4/4), done.
```

Рисунок 2. Клонирование репозитория

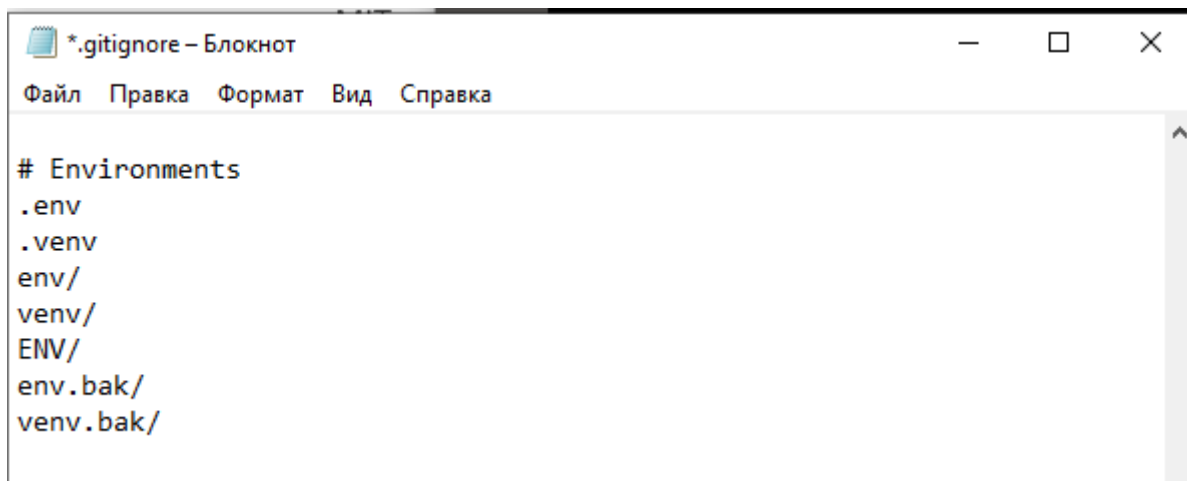


Рисунок 3. Изменение файла .gitignore

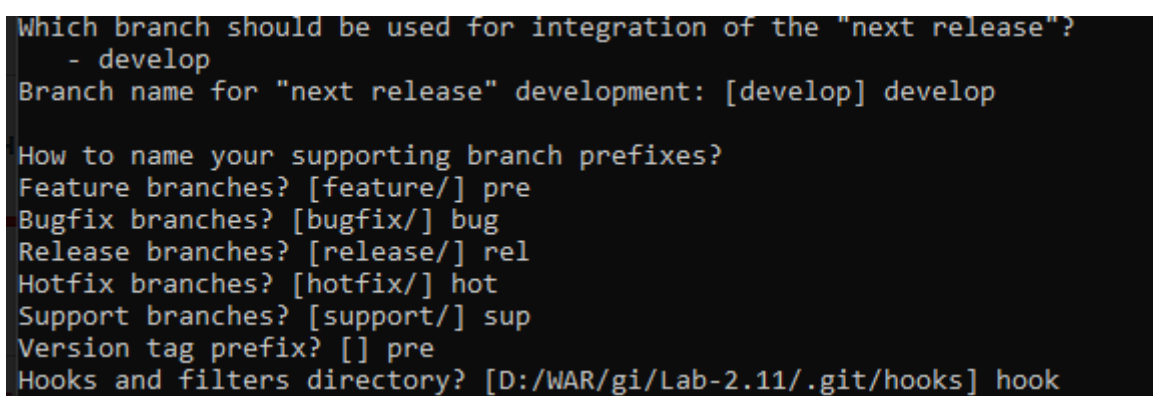


Рисунок 4. Организация репозитория в соответствии с git-flow

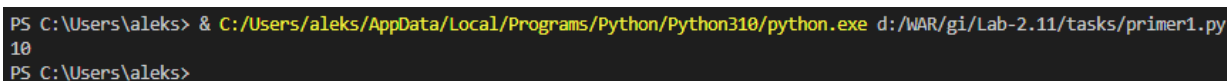


Рисунок 5. Результат выполнения примера 1

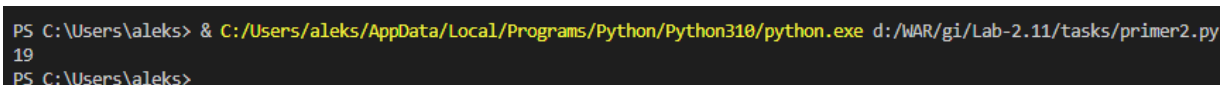
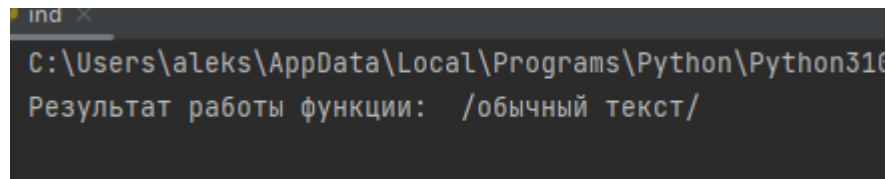


Рисунок 6. Результат выполнения примера 2

Индивидуальное задание (вариант 2):

Используя замыкания функций, объявите внутреннюю функцию, которая заключает строку *s* (*s* – строка, параметр внутренней функции) в произвольный тег, содержащийся в переменной *tag* – параметре внешней функции. Далее, на вход программы поступает две строки: первая с тегом, вторая с некоторым содержимым. Вторую строку нужно поместить в тег из первой строки с помощью реализованного замыкания. Результат выведите на экран.



```
ind x
C:\Users\aleks\AppData\Local\Programs\Python\Python310
Результат работы функции: /обычный текст/
```

Рисунок 7. Результат выполнения индивидуального задания

Контрольные вопросы:

1. Что такое замыкание?

Замыкание (closure) в программировании — это функция, в теле которой присутствуют ссылки на переменные, объявленные вне тела этой функции в окружающем коде и не являющиеся ее параметрами.

2. Как реализованы замыкания в языке программирования Python?

Замыкания в Python реализованы посредством манипулирования областью видимости функций.

3. Что подразумевает под собой область видимости Local?

Эту область видимости имеют переменные, которые создаются и используются внутри функций.

```
>>> def add_two(a):
x = 2
return a + x
>>> add_two(3)
5
>>> print(x)
Traceback (most recent call last):
File "", line 1, in print(x)
NameError: name 'x' is not defined
```

Пример. В данной программе объявлена функция `add_two()`, которая прибавляет двойку к переданному ей числу и возвращает полученный результат. Внутри этой функции используется переменная `x`, доступ к которой снаружи невозможен. К тому же, эта переменная удаляется из

памяти каждый раз (во всяком случае, должна удаляться), когда завершается `add_two()`.

4. Что подразумевает под собой область видимости Enclosing?

Суть данной области видимости в том, что внутри функции могут быть вложенные функции и локальные переменные, так вот локальная переменная функции для ее вложенной функции находится в enclosing области видимости.

5. Что подразумевает под собой область видимости Global?

Переменные области видимости `global` – это глобальные переменные уровня модуля (модуль – это файл с расширением `.py`).

6. Что подразумевает под собой область видимости Build-in?

Уровень Python интерпретатора. В рамках этой области видимости находятся функции `open`, `len` и т. п., также туда входят исключения. Эти сущности доступны в любом модуле Python и не требуют предварительного импорта. Built-in – это максимально широкая область видимости.

7. Как использовать замыкания в языке программирования Python?

Для создания замыкания в Python, должны быть выполнены следующие пункты:

- У нас должна быть вложенная функция (функция внутри функции);
- вложенная функция должна ссылаться на значение, определенное в объемлющей функции;
- объемлющая функция должна возвращать вложенную функцию.

8. Как замыкания могут быть использованы для построения иерархических данных?

Начнем разбор данного термина с математической точки зрения, а точнее с алгебраической. Предметом алгебры является изучение алгебраических структур – множеств с определенными на них операциями. Под множеством обычно понимается совокупность определенных объектов. Наиболее простым примером числового множества, является множество натуральных чисел. Оно содержит следующие числа: 1, 2, 3, ... и т.д. до

бесконечности. Иногда, к этому множеству относят число ноль, но мы не будем этого делать. Над элементами этого множества можно производить различные операции, например, сложение. Какие бы натуральные числа мы не складывали, всегда будем получать натуральное число. С умножением точно также. Но с вычитанием и делением это условие не выполняется. Среди натуральных чисел нет числа -3 , для того, чтобы можно было использовать вычитание без ограничений, нам необходимо расширить множество натуральных чисел до множества целых чисел: Таким образом, можно сказать, что множество натуральных чисел замкнуто относительно операции сложения – какие бы натуральные числа мы не складывали, получим натуральное число, но это множество не замкнуто относительно операции вычитания.

Вывод: в результате выполнения работы были приобретены навыки по работе с замыканиями при написании программ с помощью языка программирования Python версии 3.x.