Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Пермский национальный исследовательский

политехнический университет»

(ПНИПУ)

Кафедра вычислительной математики и механики

Лабораторная работа

по дисциплине: «Информатика в приложении к отрасли»

по теме: «Git, Python»

Выполнил:

Студент группы ЦТУ-20-3б

Ушенкина София Сергеевна

Проверил:

ассистент кафедры ВММБ

Нетбай Георгий Владимирович

Пермь

2022 г

**Задание к лабораторной работе:**

**Задание №1.** Создание виртуального окружения, установка модулей в виртуальное окружение при помощи менеджера пакетов pip.

1. Создайте проект в PyCharm. При создании проекта создайте виртуальное окружение.
2. Установите в виртуальное окружение модуль numpy
3. Установите в виртуальное окружение модуль matplotlib
4. Напишите программу, которая вычисляет N первых чисел Фибоначчи, записывает их в массив и строит график: величина числа Фибоначчи от его номера.

**Задания на самостоятельную работу с Git**

Перед выполнением работы:

* установите программу GitKraken (https://www.gitkraken.com/download) – студенты могут поставить программу самостоятельно
* зарегистрируйтесь на сервисе GitKraken
* зарегистрируйтесь на сервисе Github (https://github.com/)
* авторизуйте GitKraken на сервисе Github

**Одиночная работа**

**Задание №1.** Создание репозитория на сервисе Github через браузер.

1. Создайте репозиторий lab01 на сервисе Github через браузер.
2. Клонируйте репозиторий на компьютер при помощи GitKraken
3. Создайте проект в PyCharm в данном репозитории
4. Добавьте папку с виртуальным окружением venv в исключения системы контроля версий
5. Создайте файл main.py в директории проекта с программой, которая реализует алгоритм Евклида – алгоритм нахождения наибольшего общего делителя (НОД) пары целых чисел.
6. Создайте коммит, отправьте его на сервер Github
7. Откройте в браузере репозиторий, убедитесь, что созданная программа находится в репозитории.

**Задание №2.** Создание репозитория на сервисе Github через программу GitKraken.

1. Создайте репозиторий lab02 на сервисе Github через программу GitKraken.
2. Создайте проект в PyCharm в данном репозитории
3. Добавьте папку с виртуальным окружением venv в исключения системы контроля версий
4. Создайте файл main.py в директории проекта с программой, которая реализует алгоритм решето Эратосфена – алгоритм нахождения простых чисел до заданного натурального числа путем постепенного отсеивания составных чисел.
5. Создайте коммит, отправьте его на сервер Github
6. Откройте в браузере репозиторий, убедитесь, что созданная программа находится в репозитории. Убедитесь, что в облако не попала папка venv.

**Задание №3.** Основы работы с ветками

1. Создайте репозиторий lab03 на сервисе Github через программу GitKraken.
2. Создайте проект в PyCharm в данном репозитории
3. Добавьте папку с виртуальным окружением venv в исключения системы контроля версий
4. Создайте файл main.py в директории проекта с программой, которая запрашивает с клавиатуры n чисел и сортирует их при помощи алгоритма сортировки пузырьком по возрастанию и выводит их на экран.
5. Создайте коммит, отправьте его на сервер Github
6. Откройте в браузере репозиторий, убедитесь, что созданная программа находится в репозитории.
7. Создайте ветку от текущего коммита. Назовите ветку new-feature, переключитесь на неё.
8. Измените программу так, чтобы она после чтения чисел запрашивала направление сортировки (по возрастанию/по убыванию) и производила сортировку соответствующим образом.
9. Создайте коммит, отправьте его на сервис Github.
10. Откройте в браузере репозиторий, убедитесь, что созданная программа находится в репозитории.
11. При помощи программы GitKraken, слейте изменения из ветки new-feature в основную ветку вашего репозитория.
12. Отправьте изменения на сервер GitHub.
13. Откройте в браузере репозиторий, убедитесь, что внесенные изменения находится в репозитории.

**Работа в парах**

**Задание №4а.** Основы работы с pull-requests.

1. Создайте репозиторий lab04 на сервисе Github через программу GitKraken.
2. Создайте проект в PyCharm в данном репозитории
3. Добавьте папку с виртуальным окружением venv в исключения системы контроля версий
4. Создайте файл lib.py в директории проекта с функцией, которая принимает на вход N списков и возвращает количество одинаковых элементов в них.
5. Создайте коммит, отправьте его на сервер Github
6. Откройте в браузере репозиторий, убедитесь, что созданная программа находится в репозитории.

**Основная часть:**

**Задание №1.** Создание репозитория на сервисе Github через браузер.

1. Создайте проект в PyCharm. При создании проекта создайте виртуальное окружение.

Изображение выглядит как текст

Автоматически созданное описание

1. Установите в виртуальное окружение модуль numpy

Изображение выглядит как текст

Автоматически созданное описание

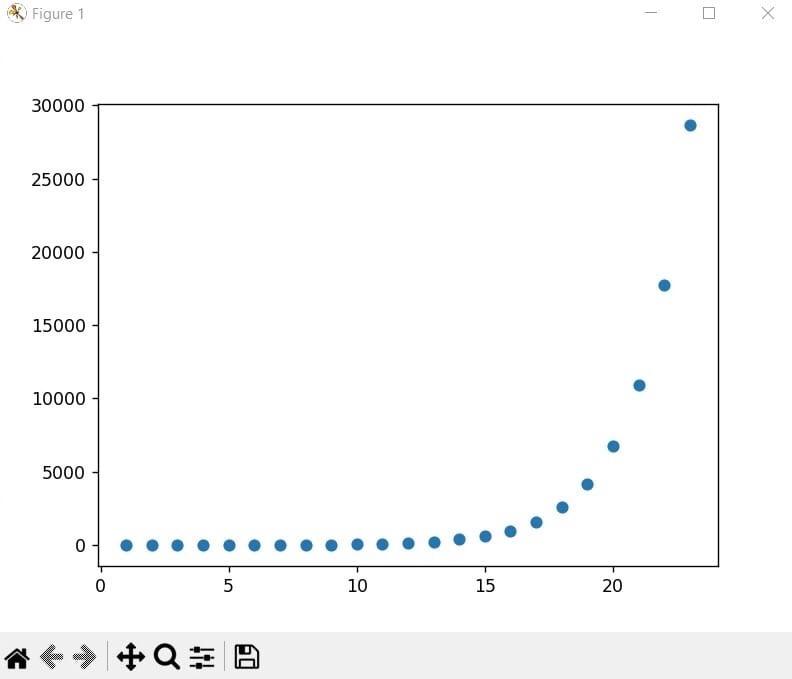
1. Установите в виртуальное окружение модуль matplotlib

Изображение выглядит как текст

Автоматически созданное описание

1. Напишите программу, которая вычисляет N первых чисел Фибоначчи, записывает их в массив и строит график: величина числа Фибоначчи от его номера.

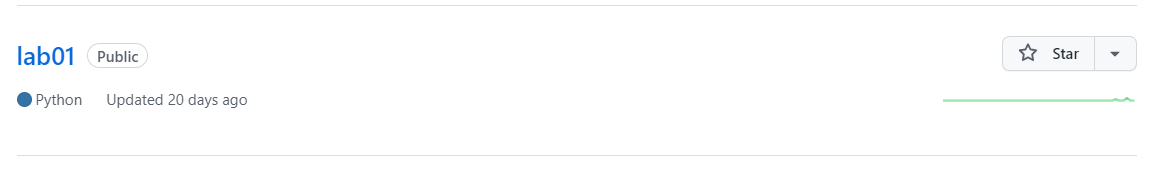
import numpy as np  
import matplotlib.pyplot as plt  
  
  
def fibonacci(x):  
 result = [1, 1]  
 if 0 >= x:  
 print("ошибка")  
 else:  
 for i in range(x):  
 result.append(result[i] + result[i + 1])  
 return result  
  
  
fig, ax = plt.subplots()  
n = int(input())  
x1 = np.array([i + 1 for i in range(n + 2)])  
y1 = np.array(fibonacci(n))  
ax.scatter(x1, y1)  
plt.show()



**Одиночная работа**

**Задание №1.** Создание репозитория на сервисе Github через браузер.

1. Создайте репозиторий lab01 на сервисе Github через браузер.



1. Клонируйте репозиторий на компьютер при помощи GitKraken



1. Создайте проект в PyCharm в данном репозитории

Изображение выглядит как текст

Автоматически созданное описание

1. Добавьте папку с виртуальным окружением venv в исключения системы контроля версий

Изображение выглядит как текст

Автоматически созданное описание

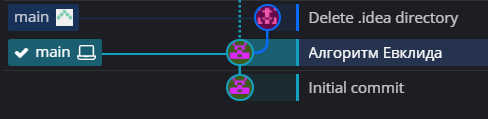
1. Создайте файл main.py в директории проекта с программой, которая реализует алгоритм Евклида – алгоритм нахождения наибольшего общего делителя (НОД) пары целых чисел.

Изображение выглядит как текст

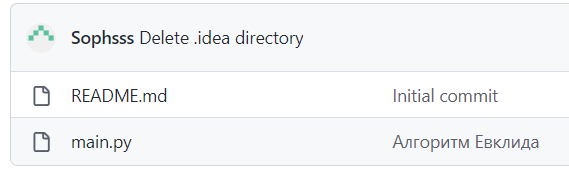
Автоматически созданное описание

a = 50  
b = 130  
  
while a != 0 and b != 0:  
 if a > b:  
 a = a % b  
 else:  
 b = b % a  
  
print(a + b)

1. Создайте коммит, отправьте его на сервер Github

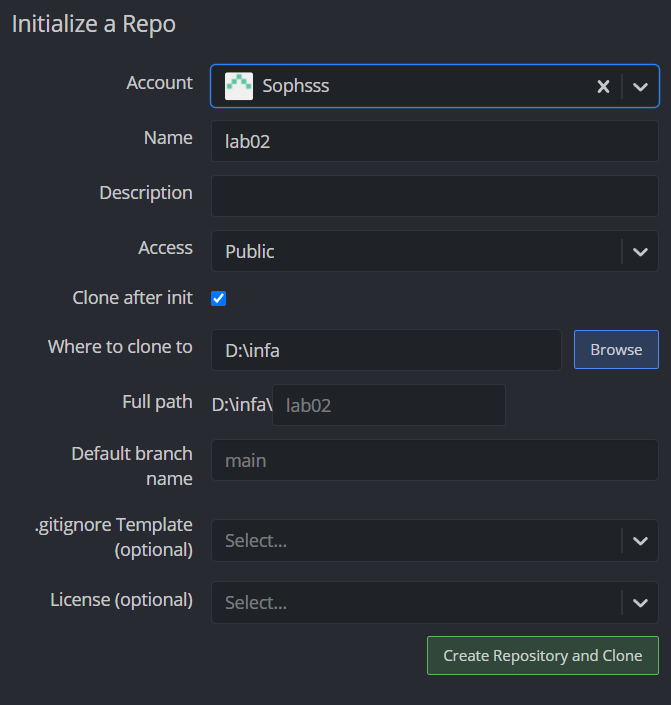


1. Откройте в браузере репозиторий, убедитесь, что созданная программа находится в репозитории.



**Задание №2.** Создание репозитория на сервисе Github через программу GitKraken.

1. Создайте репозиторий lab02 на сервисе Github через программу GitKraken.



1. Создайте проект в PyCharm в данном репозитории

Изображение выглядит как текст

Автоматически созданное описание

1. Добавьте папку с виртуальным окружением venv в исключения системы контроля версий

Изображение выглядит как текст

Автоматически созданное описание

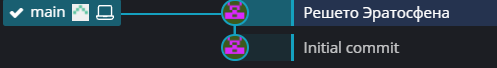
1. Создайте файл main.py в директории проекта с программой, которая реализует алгоритм решето Эратосфена – алгоритм нахождения простых чисел до заданного натурального числа путем постепенного отсеивания составных чисел.

Изображение выглядит как текст

Автоматически созданное описание

n = int(input())  
  
a = []  
for i in range(n + 1):  
 a.append(i)  
  
a[1] = 0  
  
i = 2  
while i <= n:  
 if a[i] != 0:  
 j = i + i  
 while j <= n:  
 a[j] = 0  
 j = j + i  
 i += 1  
a = set(a)  
a.remove(0)  
print(a)

1. Создайте коммит, отправьте его на сервер Github



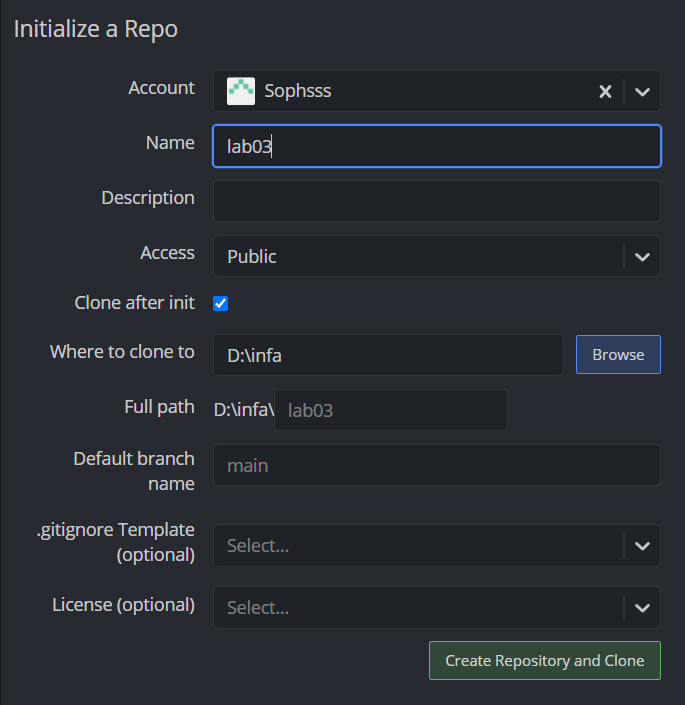
1. Откройте в браузере репозиторий, убедитесь, что созданная программа находится в репозитории. Убедитесь, что в облако не попала папка venv.

Изображение выглядит как текст

Автоматически созданное описание

**Задание №3.** Основы работы с ветками

1. Создайте репозиторий lab03 на сервисе Github через программу GitKraken.

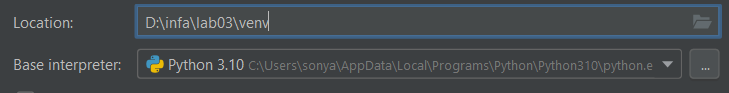


1. Создайте проект в PyCharm в данном репозитории

Изображение выглядит как текст

Автоматически созданное описание

1. Добавьте папку с виртуальным окружением venv в исключения системы контроля версий

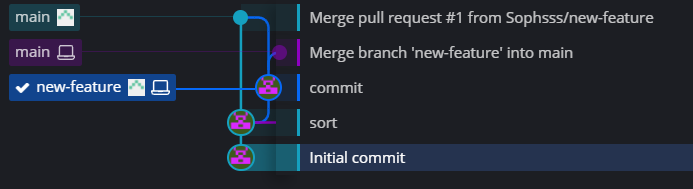


1. Создайте файл main.py в директории проекта с программой, которая запрашивает с клавиатуры n чисел и сортирует их при помощи алгоритма сортировки пузырьком по возрастанию и выводит их на экран.

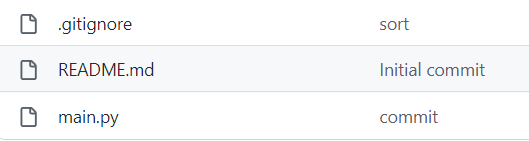
Изображение выглядит как текст

Автоматически созданное описание

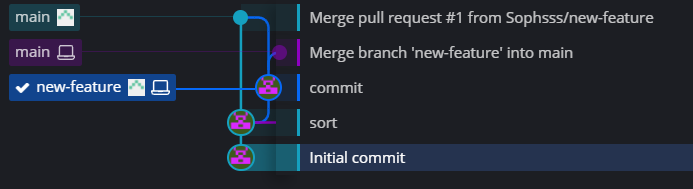
1. Создайте коммит, отправьте его на сервер Github



1. Откройте в браузере репозиторий, убедитесь, что созданная программа находится в репозитории.



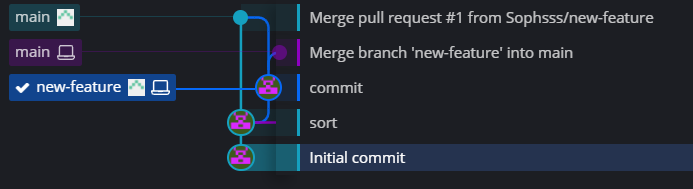
1. Создайте ветку от текущего коммита. Назовите ветку new-feature, переключитесь на неё.



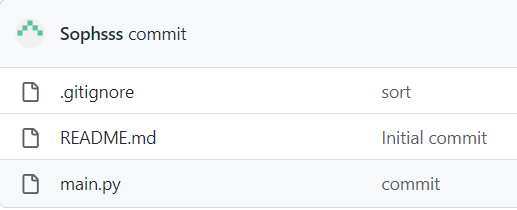
1. Измените программу так, чтобы она после чтения чисел запрашивала направление сортировки (по возрастанию/по убыванию) и производила сортировку соответствующим образом.

import random as rnd  
  
n = int(input())  
list = []  
num = int(input("По возрастанию - 1, по убыванию - 2"))  
  
for i in range(n):  
 list.append(rnd.randint(1, 99))  
print(list)  
  
if num == 1:  
 for i in range(n - 1):  
 for j in range(n - i - 1):  
 if list[j] > list[j + 1]:  
 list[j], list[j + 1] = list[j + 1], list[j]  
 print(list)  
else:  
 for i in range(n - 1):  
 for j in range(n - i - 1):  
 if list[j] > list[j + 1]:  
 list[j], list[j + 1] = list[j + 1], list[j]  
 list.reverse()  
 print(list)

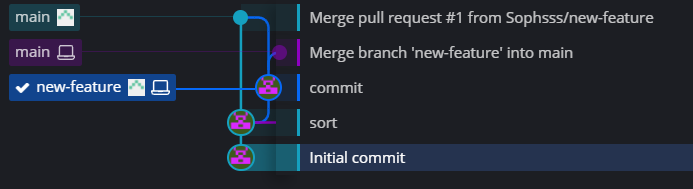
1. Создайте коммит, отправьте его на сервис Github.



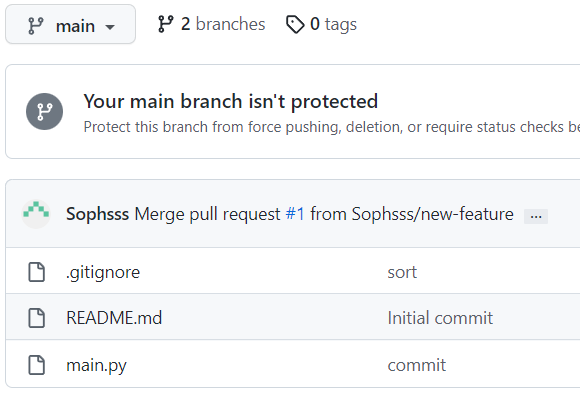
1. Откройте в браузере репозиторий, убедитесь, что созданная программа находится в репозитории.



1. При помощи программы GitKraken, слейте изменения из ветки new-feature в основную ветку вашего репозитория.



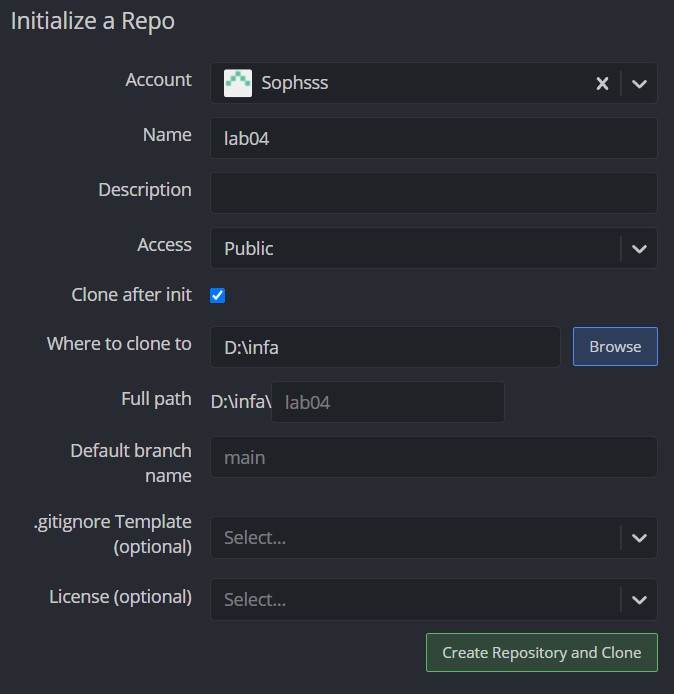
1. Отправьте изменения на сервер GitHub.
2. Откройте в браузере репозиторий, убедитесь, что внесенные изменения находится в репозитории.



**Работа в парах**

**Задание №4а.** Основы работы с pull-requests.

1. Создайте репозиторий lab04 на сервисе Github через программу GitKraken.

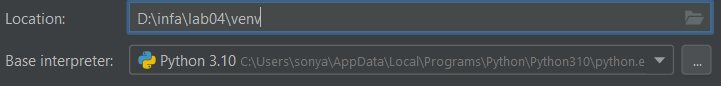


1. Создайте проект в PyCharm в данном репозитории

Изображение выглядит как текст

Автоматически созданное описание

1. Добавьте папку с виртуальным окружением venv в исключения системы контроля версий



1. Создайте файл lib.py в директории проекта с функцией, которая принимает на вход N списков и возвращает количество одинаковых элементов в них.

Изображение выглядит как текст

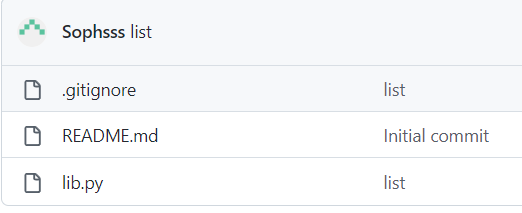
Автоматически созданное описание

def lib(a):  
 b = a[0]  
 c = 0  
 for i in range(1, len(a)):  
 for j in range(len(a[i])):  
 if(a[i][j] in b):  
 c += 1  
 b.append(a[i])  
 return c

1. Создайте коммит, отправьте его на сервер Github



1. Откройте в браузере репозиторий, убедитесь, что созданная программа находится в репозитории.



**Заключение:**

В лабораторной работе мы изучили принцип работы git, gitkraken. Реализовали алгоритм, вычисляющий N первых чисел Фибоначчи, алгоритм нахождения наибольшего общего делителя, алгоритм решето Эратосфена, алгоритм сортировки пузырьком.

В ходе работы узнали, что такое виртуальное окружение Python, какие существуют способы создать виртуальное окружение. Что такое pip, commit, как с помощью GitKraken просмотреть историю коммитов. Что такое ветка в git, как создать, объединить ветки. Что такое fork, pull request, gitignore.