|  |
| --- |
| Федеральное государственное автономное |
| образовательное учреждение |
| Высшего образования |
| «СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ» |
| Институт космических и информационных технологий |
| институт |
| Программная инженерия |
| кафедра |

|  |
| --- |
| **ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ № 4** |
| Работа с удаленными репозиториями |
| тема |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Преподаватель | |  |  |  | П.В. Пересунько |
| подпись, дата | инициалы, фамилия |
| Студент | КИ23-17/1б, 32321401 |  |  |  | П.Ю. Городилов |
| номер группы, зачётной книжки | |  | подпись, дата |  | инициалы, фамилия |

Красноярск 2023

1. Цель

Практиковаться в работе с удаленным репозиторием.

1. Задачи

* Создать локальный репозиторий, создать публичный удаленный репозиторий и связать его с локальным репозиторием. Для удаленного репозитория может быть использовать GitLab.com, GitHub.com, gitea.com и другие по выбору студента.
* Написать код программы, согласно своему варианту.
* Запушить свои наработки в удаленный репозиторий.
* Зайти на удаленный репозиторий напарника и сделать форк проекта себе. Можно напарника эмулировать, если не смогли ни с кем договориться.
* Связать репозиторий с локальным, сделать изменения в репозитории, запушить изменения.
* Отправить запрос на слияние.
* Принять запрос на слияние от своего друга.
* Предоставить отчет, программу и скрипты на проверку, ответить на вопросы и выполнить дополнительные задания по усмотрению преподавателя.

1. Задание

**Необходимо написать две функции:**

* первая будет делать полученный цвет светлее;
* вторая будет делать полученный цвет темнее.

**Входные данные**

На вход каждой их этих функций подаётся строка «#HHHHHH», а также процентное значение, на которое нужно осветлить или затемнить полученный цвет.

**Сигнатуры функций и примеры**

Функции обязательно должны соответствовать определённым сигнатурам:

def lighten (color: str, percent: int) –> str: для функции осветления, и для затемнения: def darken (color: str, percent: int) –> str:

1. Основная часть
   1. Практическая часть

Сперва создал репозиторий на GitHub, затем fetch’ем извлек удаленные данные на локальный репозиторий.

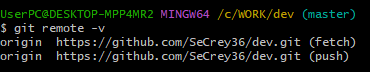


Рисунок 1 – Список соединенных удаленных репозиториев

Далее был написан код программы в соответствии с ТЗ. Консольная и GUI-версии.

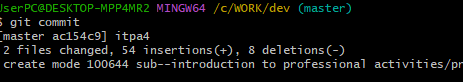


Рисунок 2 – Коммит изменений

Наработки были запушены в удаленный репозиторий.

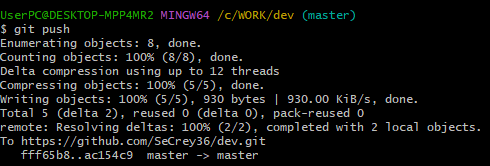
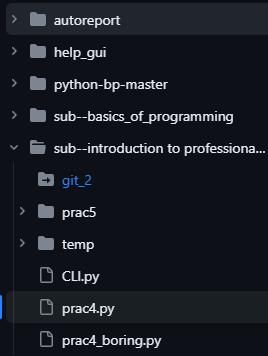


Рисунок 3 – Push-лог в Git Bash

Попросим иного пользователя добавить новый файл к репозиторию. После этого примем от него запрос на слияние (рисунок 8).

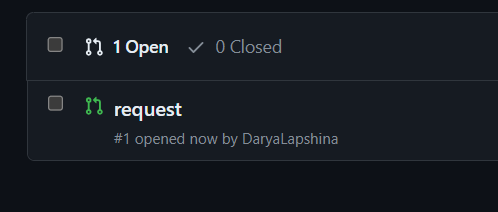


Рисунок 4 – Pull-request от пользователя

После слияния выполним пулл в локальном репозитории, чтобы актуализировать данные (рисунок 8).

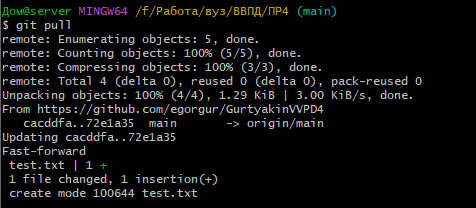


Рисунок 8 – Выполнил пулл

1. Вывод

В этой лабораторной были изучены способы работы с удалённым репозиторием git. На практике связали локальный и удалённый репозитории. Запушили изменения в удалённый репозиторий. Была изучена механика выполнения запроса на слияние.

.

**ПРИЛОЖЕНИЕ А**

**Код функций:**

def hex\_interpreter(h):  
 h = h.lstrip('#')  
 return list(int(h[i:i + 2], 16) for i in (0, 2, 4))  
  
  
def rgb\_interpreter(rgb):  
 return '#%02x%02x%02x' % rgb  
  
def lighten(h, percent):  
 color = hex\_interpreter(h)  
 goal\_1 = color[0] + (255 - color[0]) \* percent // 100  
 goal\_2 = color[1] + (255 - color[1]) \* percent // 100  
 goal\_3 = color[2] + (255 - color[2]) \* percent // 100  
 print(color)  
 print(goal\_1, goal\_2, goal\_3)  
 color = (goal\_1, goal\_2, goal\_3)  
 return rgb\_interpreter(color)  
  
def darken(h, percent):  
 color = hex\_interpreter(h)  
 goal\_1 = color[0] \* percent // 100  
 goal\_2 = color[1] \* percent // 100  
 goal\_3 = color[2] \* percent // 100  
 print(color)  
 print(goal\_1, goal\_2, goal\_3)  
 color = (goal\_1, goal\_2, goal\_3)  
 return rgb\_interpreter(color)