|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |   Институт Информационных технологий | |
|  | |
| Кафедра Математического обеспечения и стандартизации информационных технологий | |
|  | |
|  | |

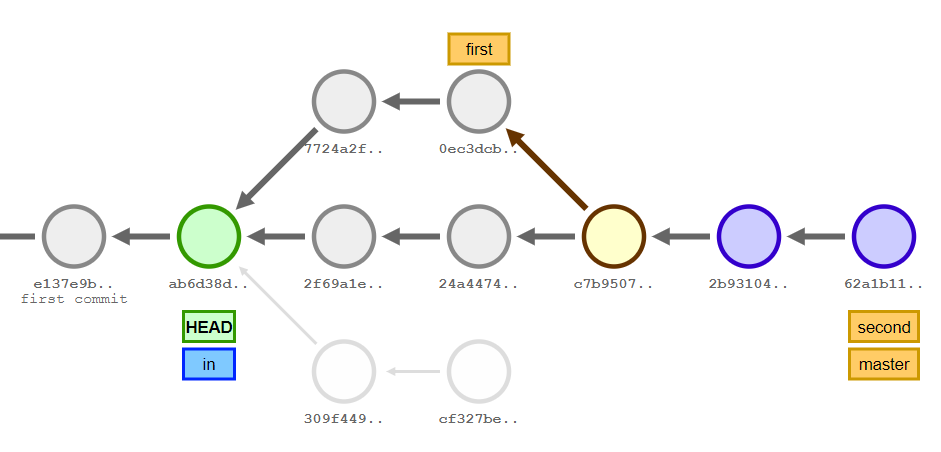
|  |  |
| --- | --- |
| **Практическая работа № 5** | |
| **по дисциплине** | |
| **«**Конфигурационное управление**»** | |
|  | |
| Выполнил студент группы | . |
| Принял преподаватель | . |

|  |  |  |
| --- | --- | --- |
| Лабораторная работа выполнена | «\_\_»\_\_\_\_\_\_\_202\_\_ г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_202\_\_ г. | *(подпись руководителя)* |

Москва 2023

# **Задача 1**

На сайте [https://onlywei.github.io/explain-git-with-d3](https://onlywei.github.io/explain-git-with-d3%20) или [http://git-school.github.io/visualizing-git/](http://git-school.github.io/visualizing-git/%20) (цвета могут отличаться, есть команды undo/redo) с помощью команд эмулятора git получить следующее состояние проекта (сливаем master с first, перебазируем second на master): см. картинку ниже. Прислать свою картинку.

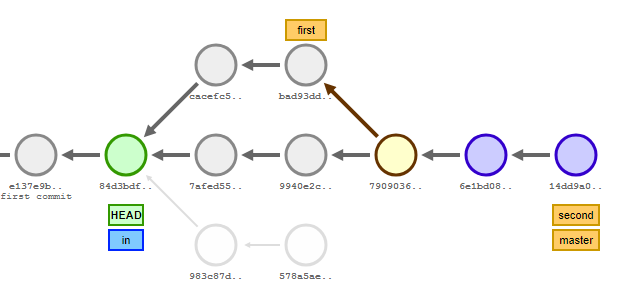


***Решение:***

Последовательность git-команд:

|  |
| --- |
| git commit  git branch first  git branch second  git commit  git commit  git checkout first  git commit  git commit  git checkout second  git commit  git commit  git checkout master  git merge first  git checkout second  git rebase master  git checkout master  git reset second  git checkout 84d3bdf  git tag in |

Результат построения:

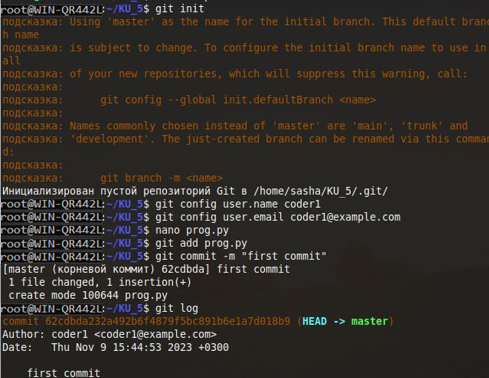


# **Задача 2**

Создать локальный git-репозиторий. Задать свои имя и почту (далее – coder1). Разместить файл prog.py с какими-нибудь данными. Прислать в текстовом виде диалог с git.

***Решение:***

Лог терминала:



# **Задача 3**

Создать рядом с локальным репозиторием bare-репозиторий с именем server. Загрузить туда содержимое локального репозитория. Команда git remote -v должна выдать информацию о server! Синхронизировать coder1 с server.

Клонировать репозиторий server в отдельной папке. Задать для работы с ним произвольные данные пользователя и почты (далее – coder2). Добавить файл readme.md с описанием программы. Обновить сервер.

Coder1 получает актуальные данные с сервера. Добавляет в readme в раздел об авторах свою информацию и обновляет сервер.

Coder2 добавляет в readme в раздел об авторах свою информацию и решает вопрос с конфликтами.

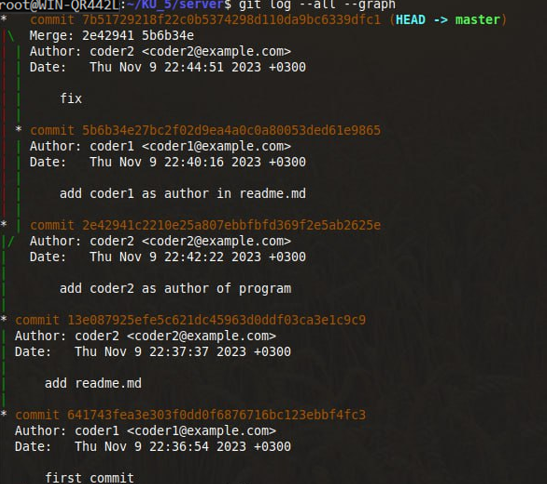
Прислать список набранных команд и содержимое git log.

***Решение:***

Последовательность git-команд:

|  |
| --- |
| cd ../  git init --bare server  cd rep1  git remote add origin ../server  git push origin master  cd ../  git clone server rep2  cd rep2  git config user.name coder2  git config user.email coder2@example.com  nano readme.md  git add readme.md  git commit -m "add readme.md"  git push origin master  cd ../rep1  git pull origin master  nano readme.md  git add readme.md  git commit -m "add coder1 as author of program"  git push origin master  cd ../rep2  nano readme.md  git add readme.md  git commit -m "add coder2 as author of program"  git push origin master  git pull --no-ff  nano readme.md  git add readme.md  git commit -m "fix"  git push origin master |

Содержимое git log:



# **Задача 4**

Написать программу на Питоне (или другом ЯП), которая выводит список содержимого всех объектов репозитория. Воспользоваться командой "git cat-file -p". Идеальное решение – не использовать иных сторонних команд и библиотек для работы с git.

***Решение:***

Код программы на Python:

|  |
| --- |
| **import** **os**  **import** **subprocess**  **try**:  files = [dirs + files **for** dirs **in** os.listdir(".git/objects") **for** files **in** os.listdir(".git/objects" + "/" + dirs)]  **except** FileNotFoundError:  **try**:  files = [dirs + files **for** dirs **in** os.listdir("objects") **for** files **in** os.listdir("objects" + "/" + dirs)]  **except** FileNotFoundError:  print("git-репозиторий не обнаружен")  exit(**1**)    **for** obj **in** files:  subprocess.call("git cat-file -p " + obj, shell=**True**) |

Тестирование:

