ФЕДЕРАЛЬНОЕ АГЕНСТВО СВЯЗИ

Ордена Трудового Красного Знамени федеральное государственное

Бюджетное учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра информатики

Лабораторная работа №1

По дисциплине: «Технология разработки программного обеспечения»

Тема: «Основы работы с системой контроля версий»

Выполнил студент:

Группы: БФИ1902

Шацкий Е. И.

Проверила: Мосева М.С

Москва, 2020 г.

**Содержание**

[**1. Цель работы 2**](#_Toc51019401)

[**2. Задание 2**](#_Toc51019402)

[**3. Ход работы 3**](#_Toc51019403)

[**Вывод 6**](#_Toc51019404)

[**Список использованных источников 7**](#_Toc51019405)

**1. Цель работы**

Цель данной лабораторной работы — изучить подходы к командной разработке в системе контроля версий Git.

**2. Задание**

Задание на лабораторную работу — провести ряд следующих операций в предыдущей работе либо вновь созданном репозитории:

1. Настроить игнорирование git-ом файлов определенного расширения с использованием glob-шаблонов

2. Вывести лог коммитов в измененном пользовательском формате

3. Переименовать ветку

4. Провести в ветке работу и слить с веткой master, удалить слитую ветку

5. Настроить удаленный репозиторий: получить изменения из репозитория и отправить изменения

6. Организовать совместную работу над проектом в соответствии с выбранной идеологией

**3. Ход работы**

Для того, чтобы настроить игнорирование Git-ом файлов определенного расширения необходимо создать файл «.gitignore». Процесс создания файла и добавление его в репозиторий представлен на рисунке 1.

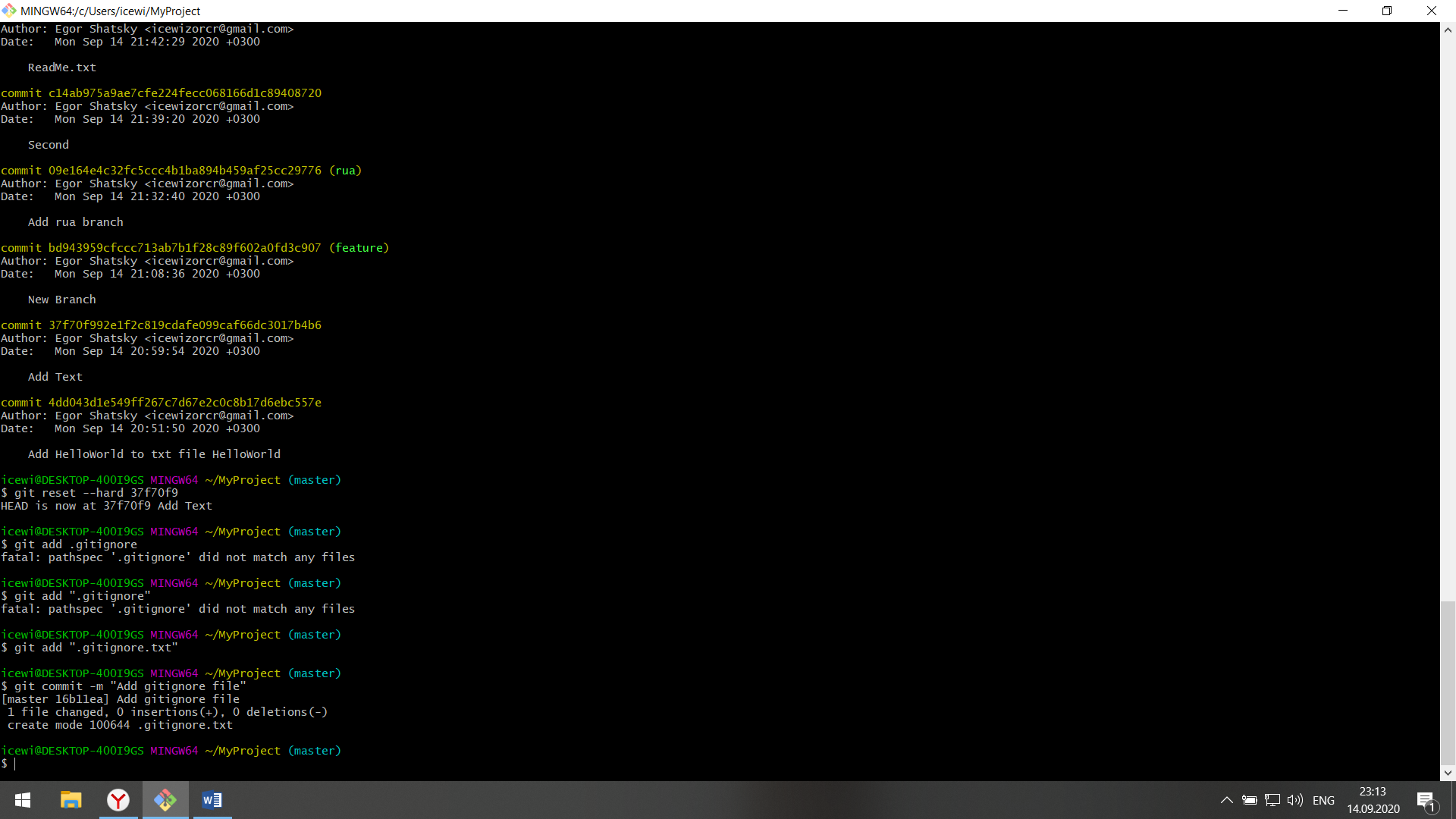
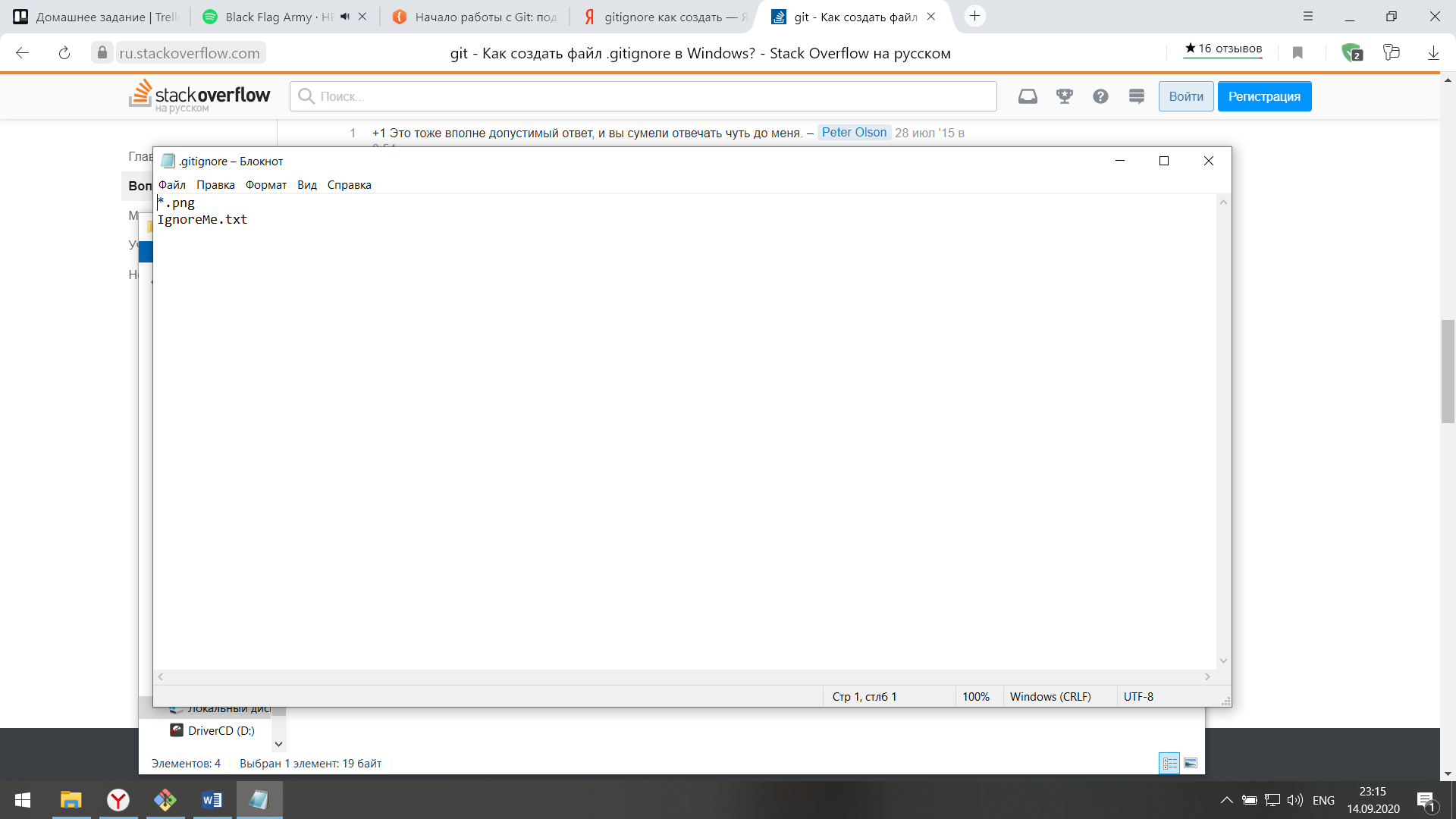
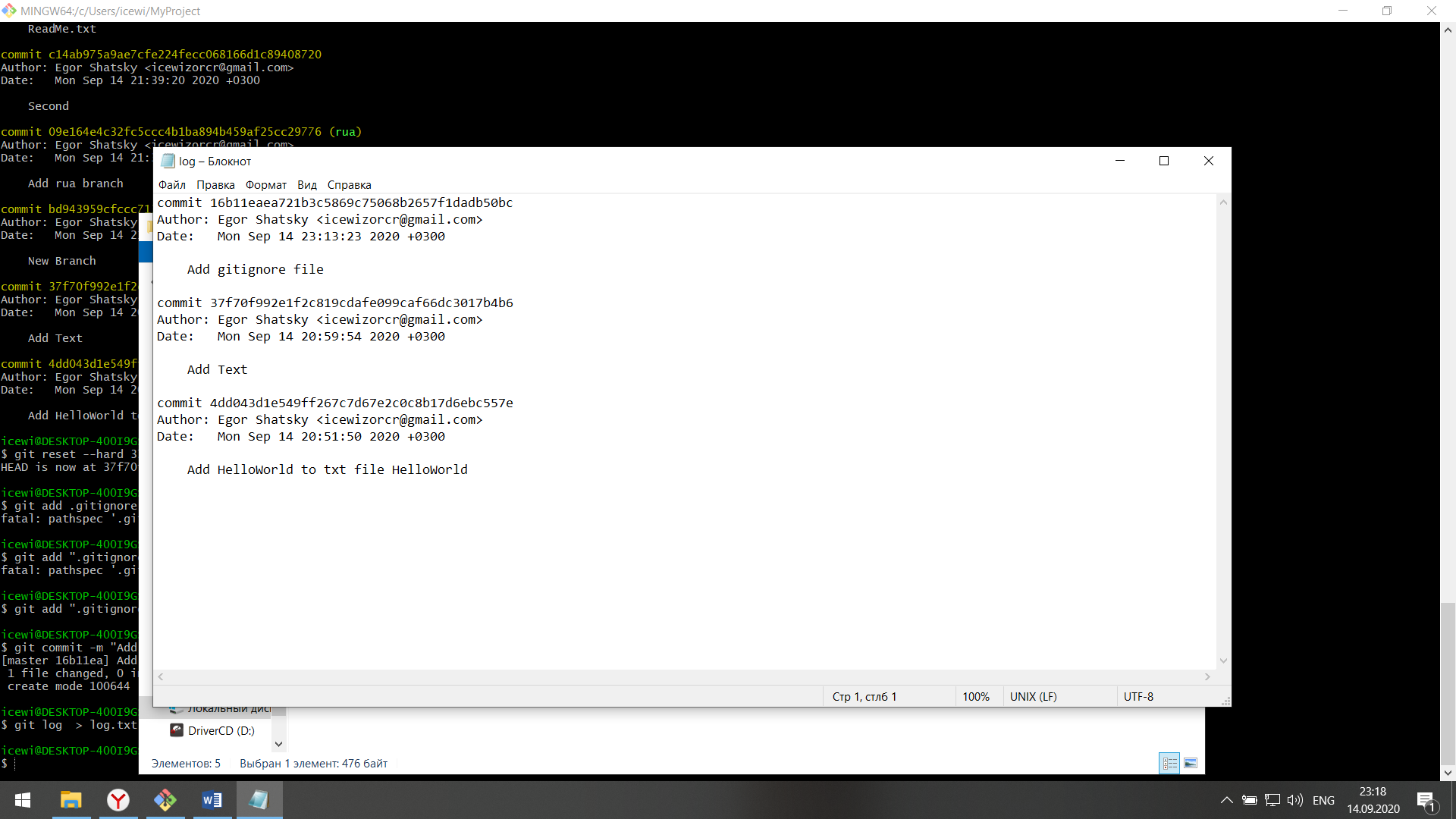


Рисунок 1 — Создание файла «.gitignore»

Внутри данного файла мы можем указать, какой формат файлов будет игнорироваться Git-ом. Для примера возьмем файлы формата «.png» и файл с названием «IgnoreMe.txt». Теперь добавление или изменение файлов формата «.png» и изменение файла «IgnoreMe.txt» будут игнорироваться Git-ом. Содержимое файла «.gitignore» представлено на рисунке 2.

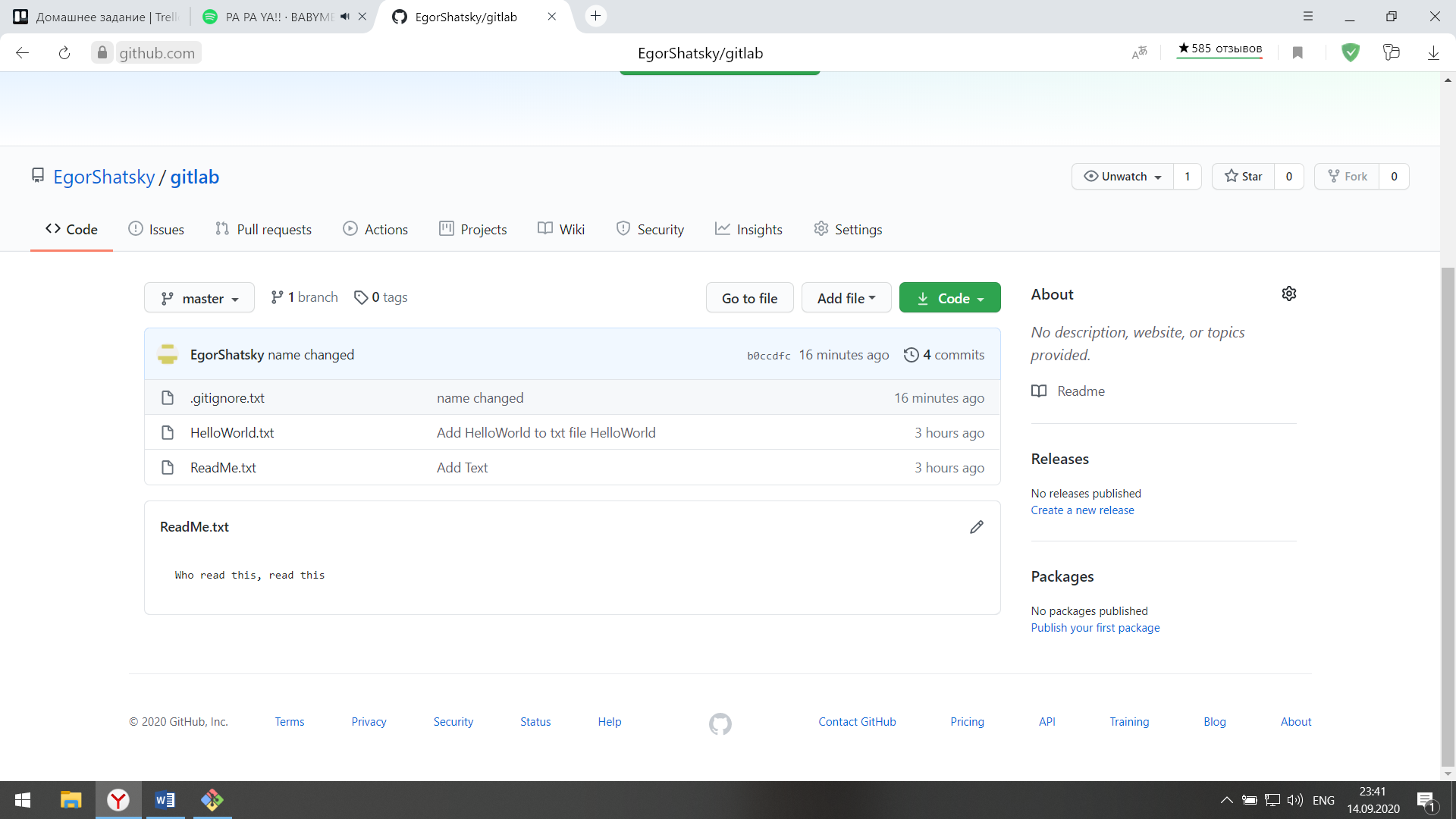
Рисунок 2 — Содержимое файла «.gitignore»

Вывести лог коммитов можно в текстовый файл при помощи команды «git log > log.txt». Содержимое файла «log.txt» представлено на рисунке 3.

Рисунок 3 — Содержимое файла «log.txt»

Теперь переименуем созданную нами в прошлой лабораторной работе ветку «rua» в «rua1». Сделаем это при помощи команды «git branch -m rua rua1». Теперь, перейдя в ветку «rua1», изменим содержимое файла «ReadMe.txt», произведем коммит и сольем ветку с веткой master. Удалим слитую ветку при помощи команды «git branch -d rua1».

Далее приступим к созданию удаленного репозиторий под названием «gitlab» на github. Для того, чтобы объединить локальный и удаленный репозитории мы воспользуемся командой «git remote add origin https://github.com/EgorShatsky/gitlab.git», после чего мы можем отправить содержимое локального репозитория на удаленный при помощи команды «git push -u origin master». Содержимое удаленного репозитория после произведенных операций представлено на рисунке 4.

Рисунок 4 — Содержимое удаленного репозитория

**Вывод**

Таким образом, мы изучили подходы к командной разработке в системе контроля версий Git.

**Список использованных источников**

1) ГОСТ 7.32-2017 Система стандартов по информации, библиотечному и издательскому делу. Отчёт о научно-исследовательской работе. Структура и правила оформления

2) ГОСТ 7.1-2003 Библиографическая запись. Библиографическое описание. Общие требования и правила составления