Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №1**

По дисциплине «Конструирование программного обеспечения»

Тема: «Система контроля версий Git»

**Выполнил:**

Студент 1 курса

Группы ПО-9

Солышко Д.А.

**Проверил:**

Скарубо А.О.

Брест 2022

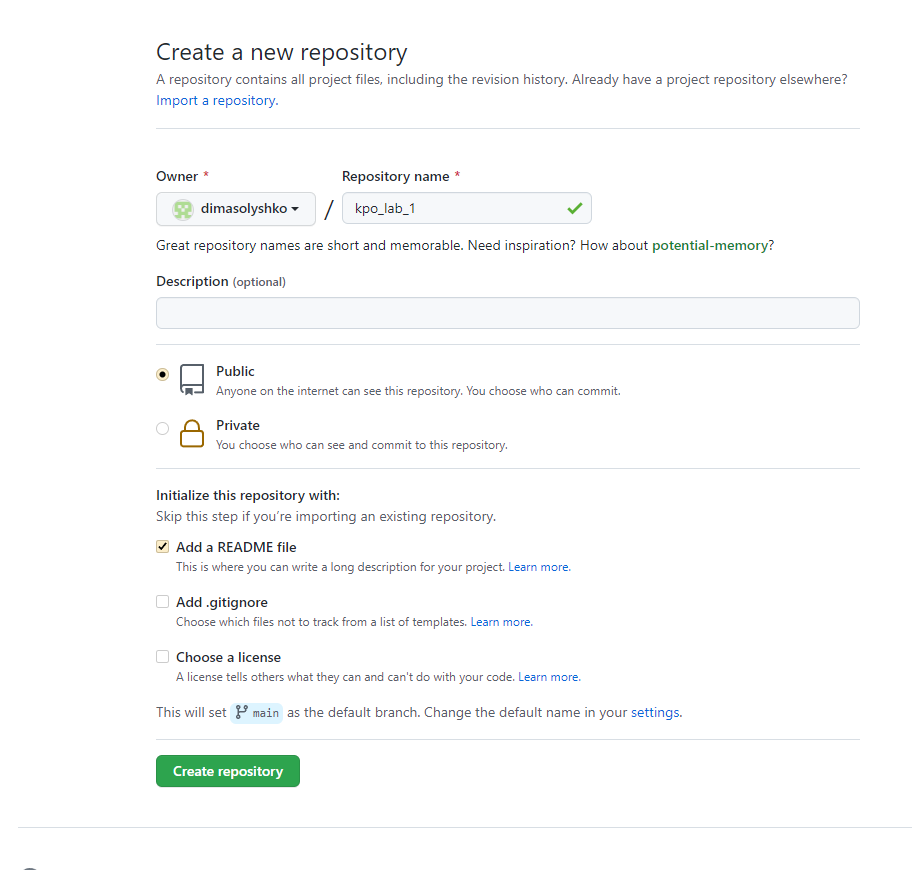
**Цель работы:** научиться использовать десктопный клиент GitHubдля управления процессом Git, изучить методы и основные функции при работе с репозиторием

**Задание:** выполнить простые операции по созданию и работе с копией удалённого репозитория

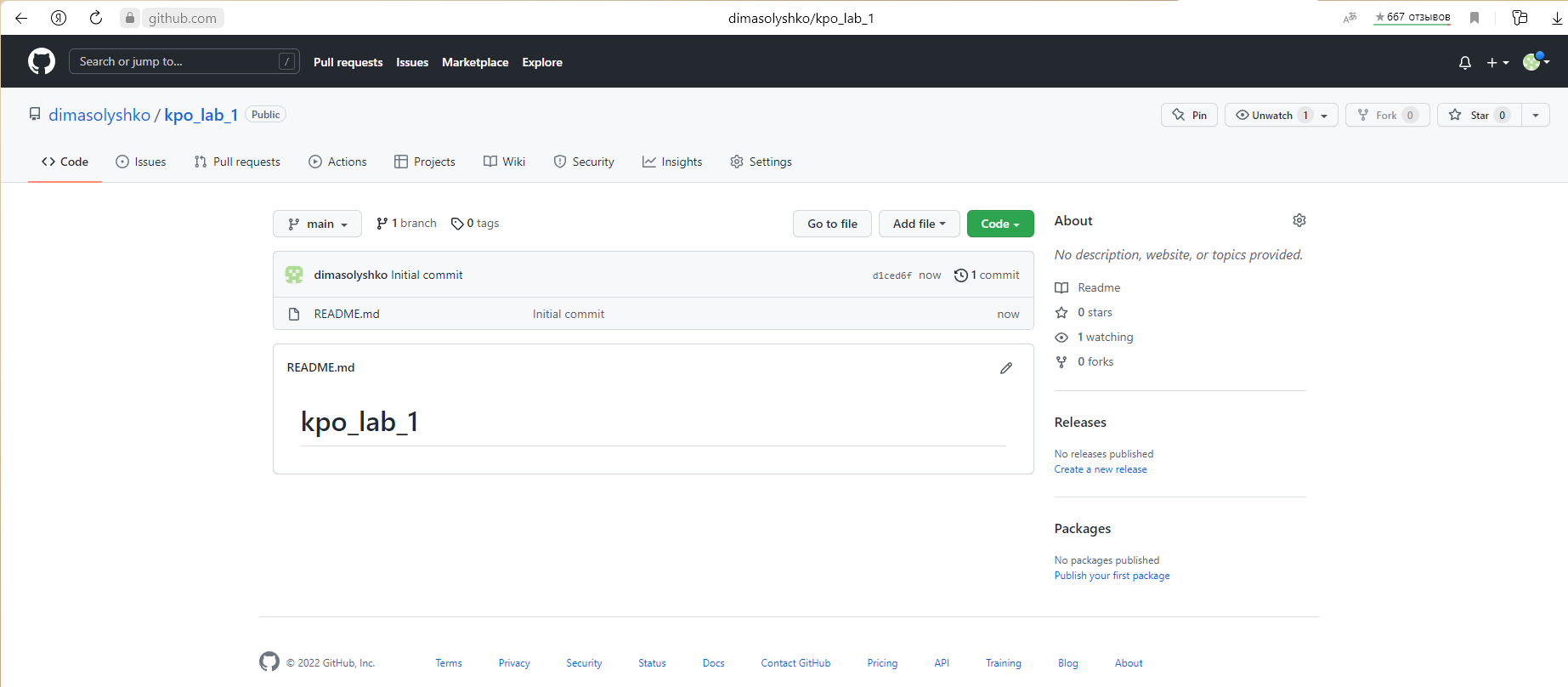
**Задание 1**

Создать удалённую копию репозитория

Настраиваем репозиторий и нажимаем “create repository”.



Мы успешно создали репозиторий!

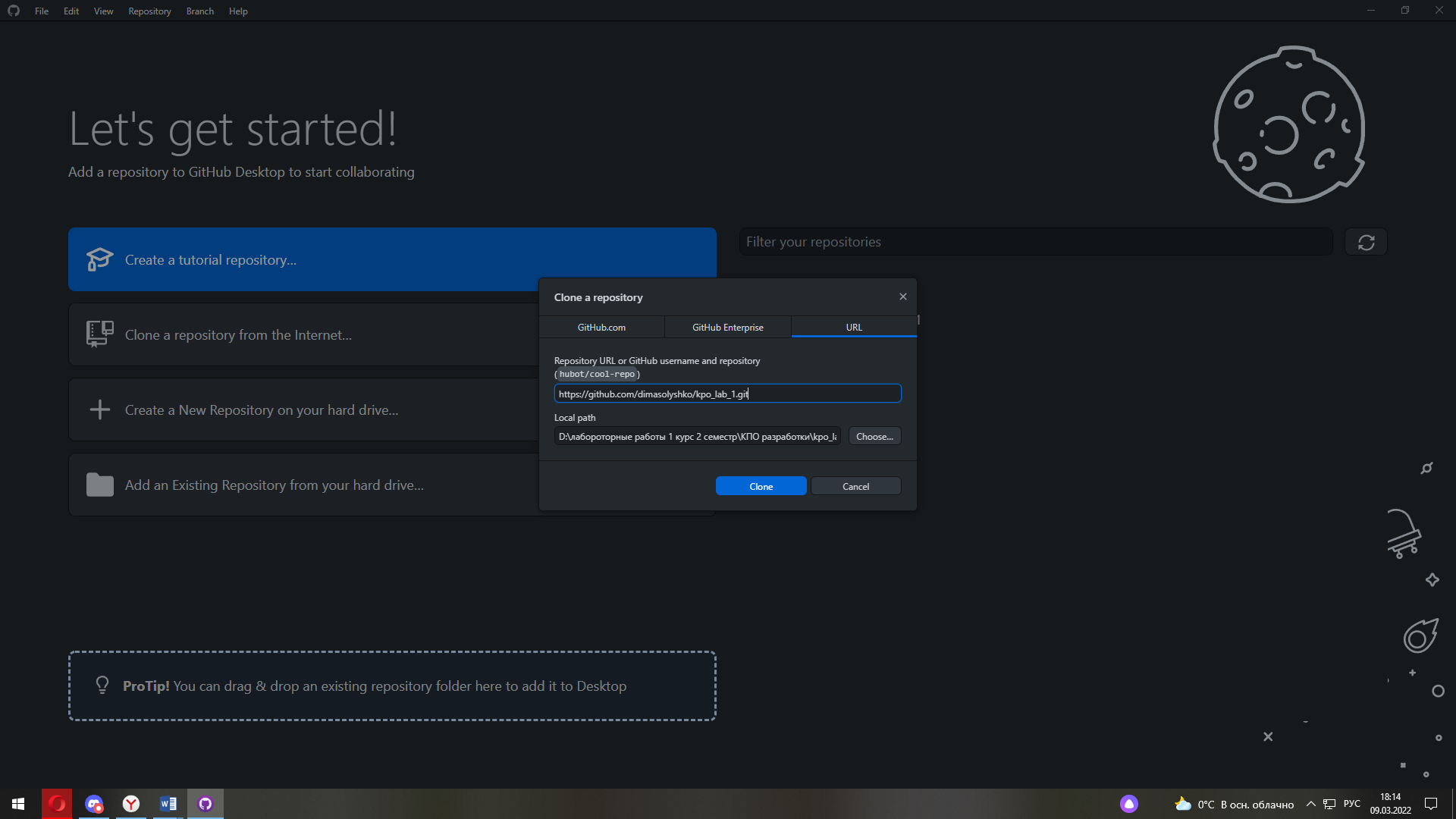


**Задание 2**

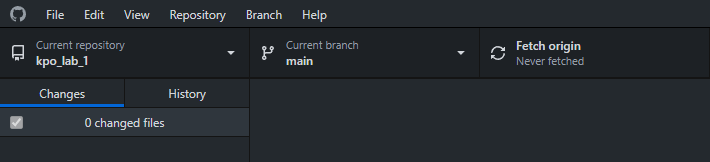
Клонировать репозиторий на компьютер

Для этого необходимо скачать приложение “GitHub Desktop”, зарегестрироваться, после чего с помощью URL-ссылки мы легко может клонировать репозиторий с сайта на компьютер.

Как мы видим репозиторий успешно клонирован.



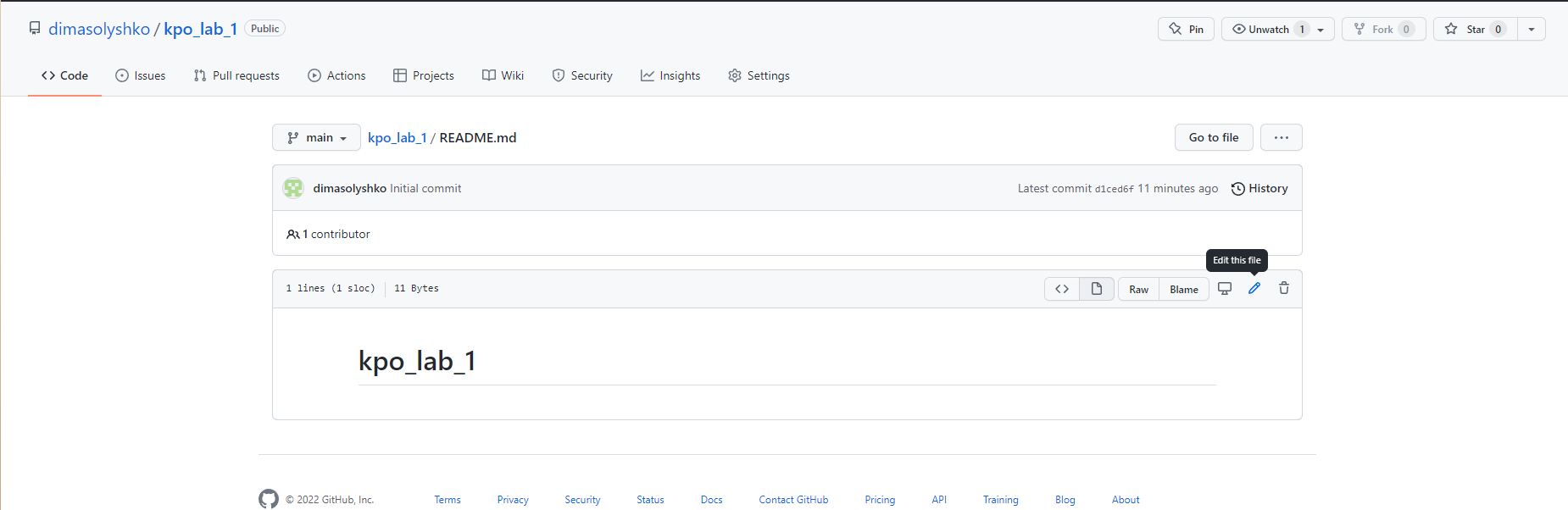
Репозиторий успешно клонирован!



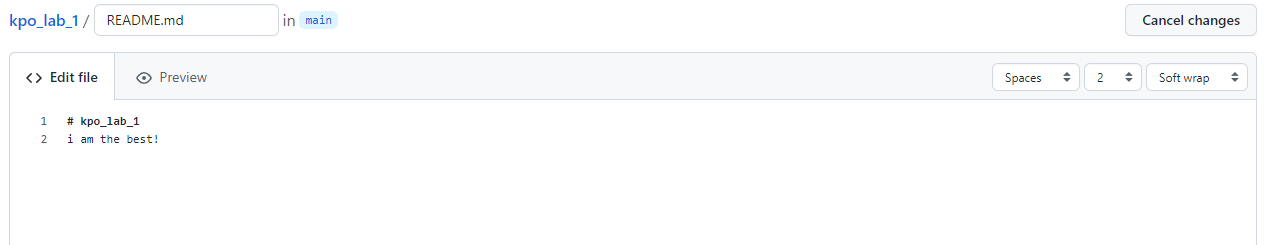
**Задание 3**

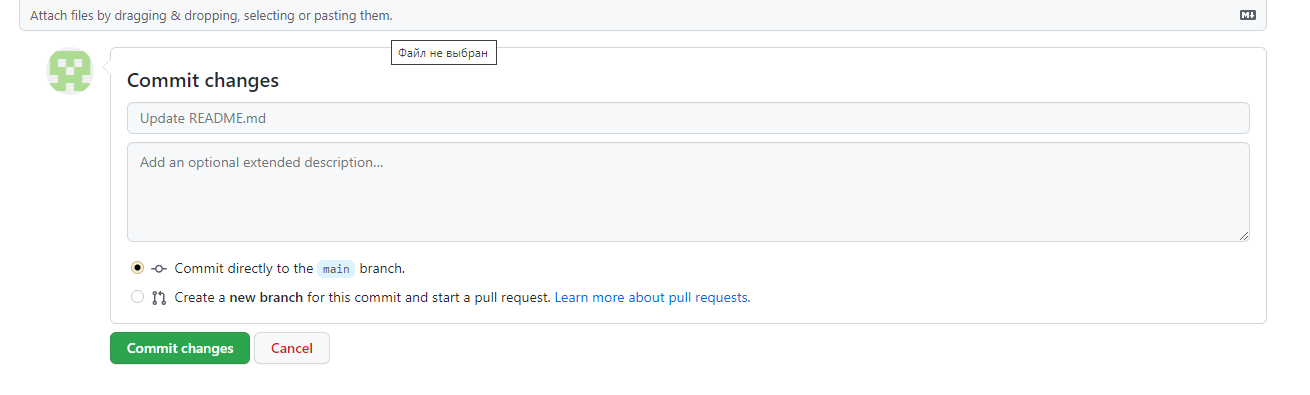
Через сайт изменить описание проекта (в файле README)

Для этого достаточно в самом репозитории возле файла README нажать на “edit this file”, перейдя в вкладку редактирование мы добавляем все что нам нужно.

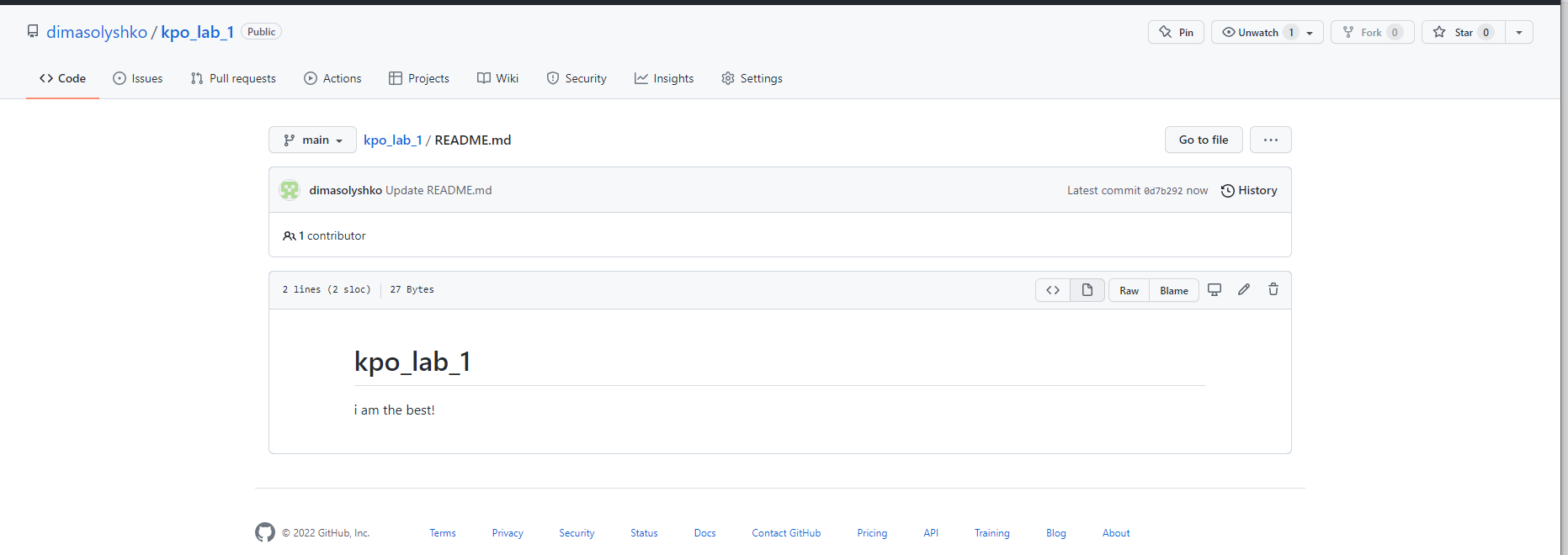


Меняем описание(добавляя I am the best!)



После переходим вниз и нажимаем “Commit changes”.

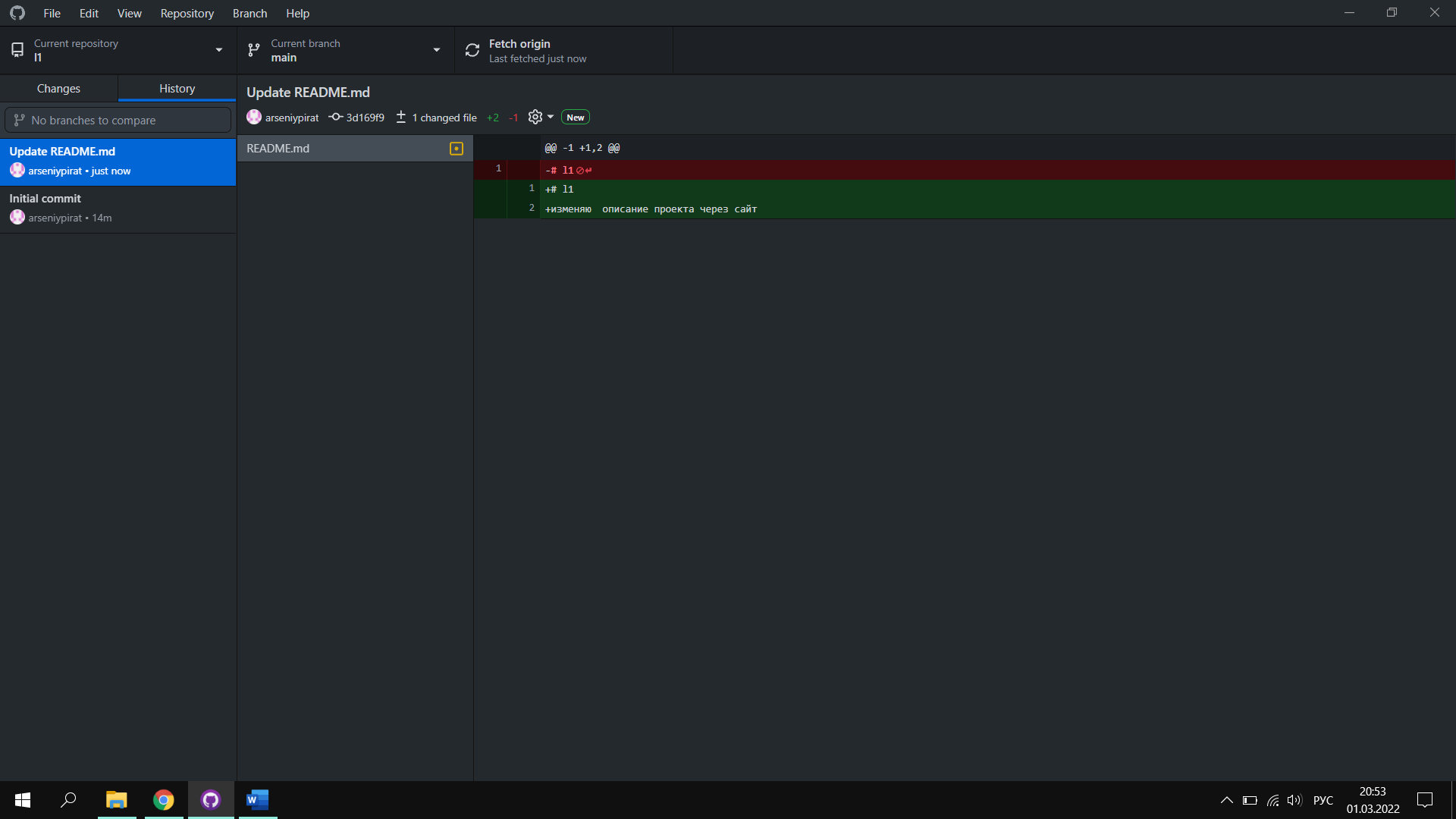
Мы изменили описание проекта через сайт!



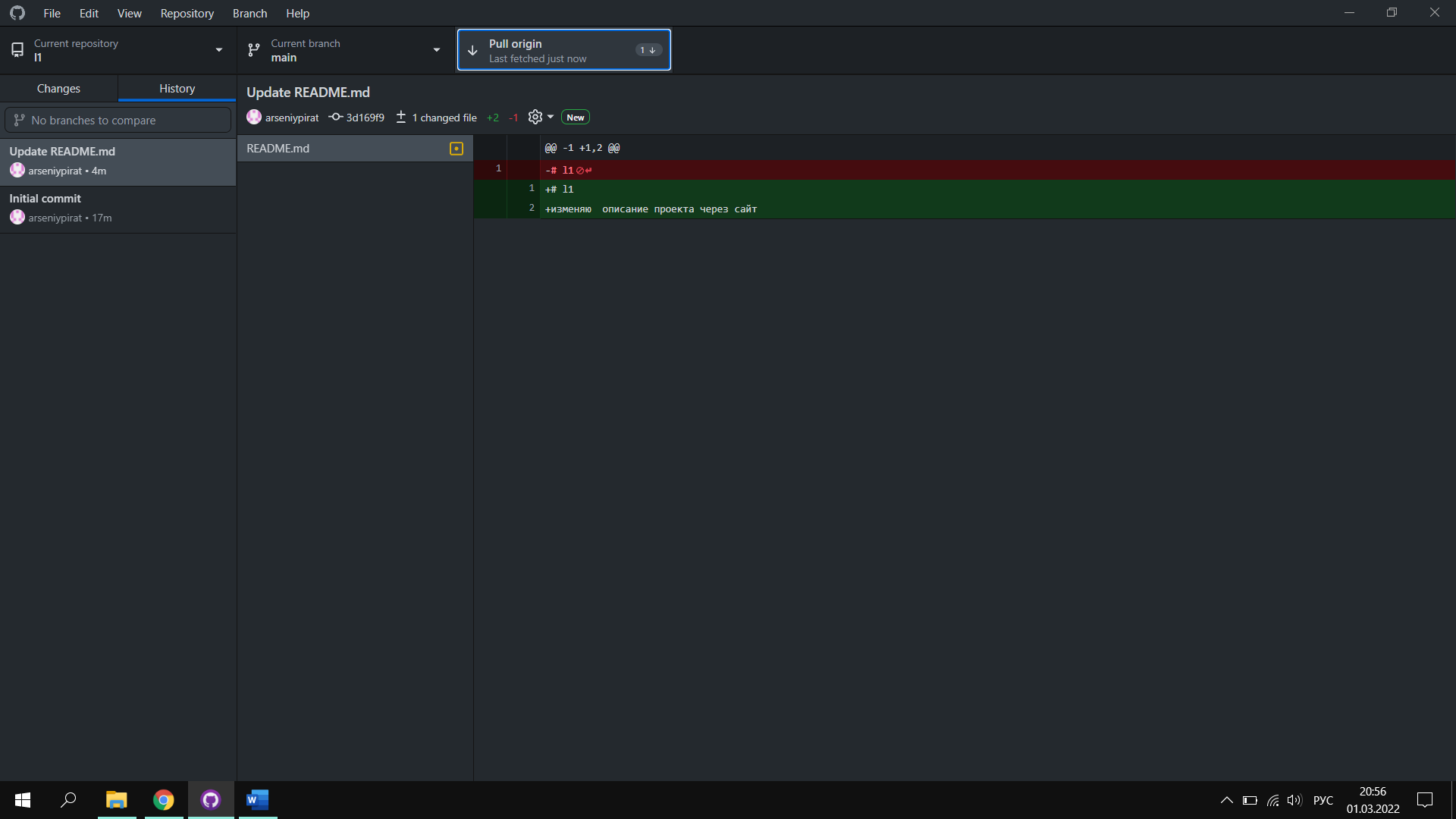
**Задание 4**

Вытягивание последних изменений в репозитории

Для вытягивание изменений нам понадобится функция “Fetch origin”, которая находится справа от вкладки выбора репозитория.



После чего для выгрузки изменений на сайт нажмем “Pull origin”.



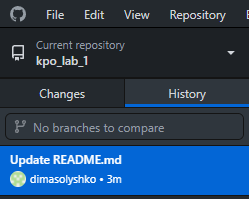
Как мы видим репозиторий был успешно выгружен, а изменения зафиксированы.



**Задание 5**

Просмотр изменений

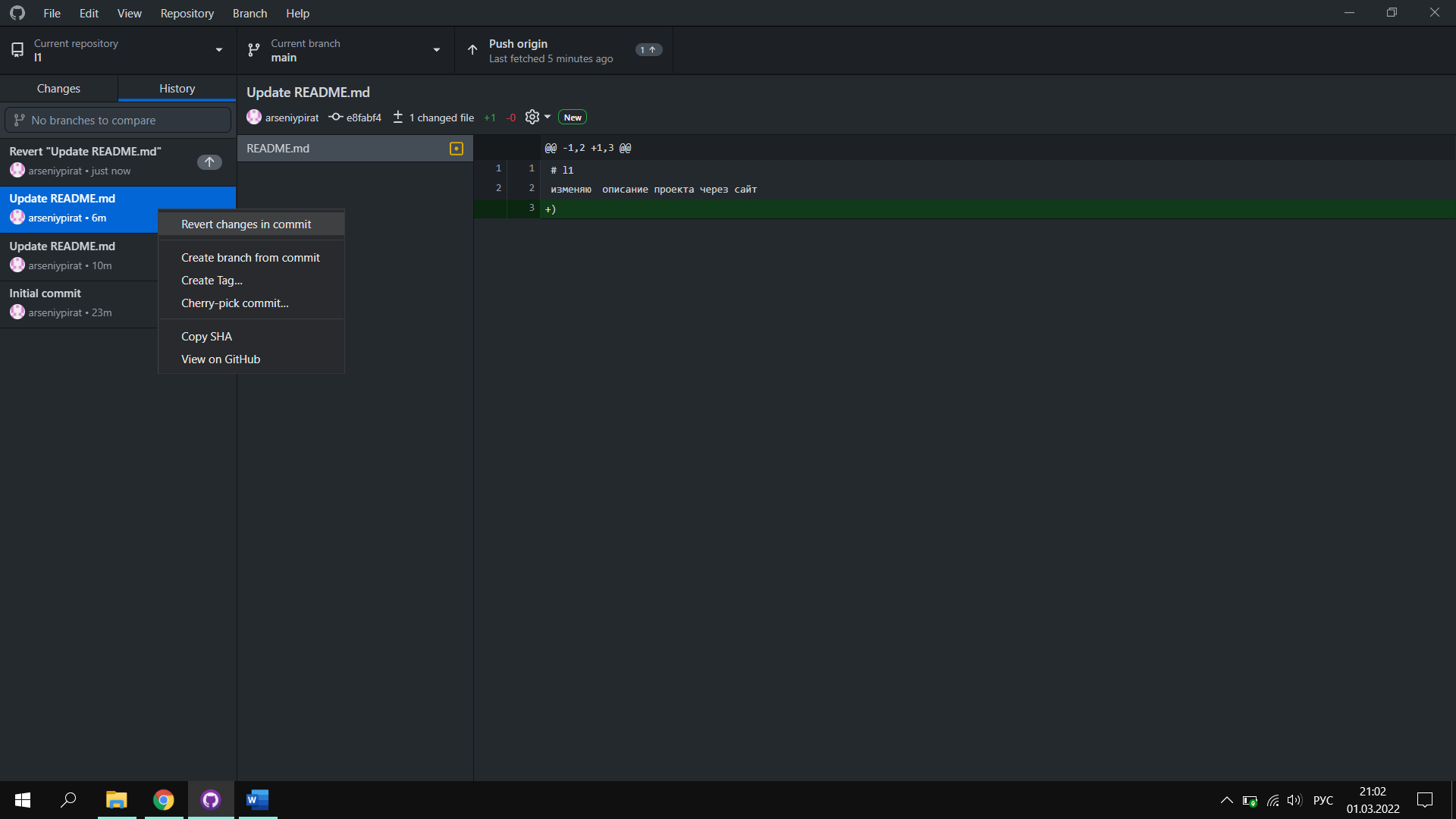
Просмотр изменений осуществляется в приложении во вкладке History справа от README.



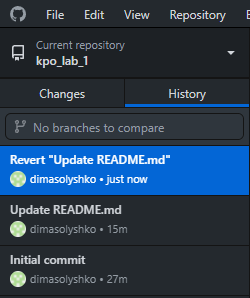
**Задание 6**

Отмена изменений в Задании 3

Для отмены изменений во вкладке History необходимо выбрать необходимый коммит, нажать ПКМ и выбрать Revert changes in commit.



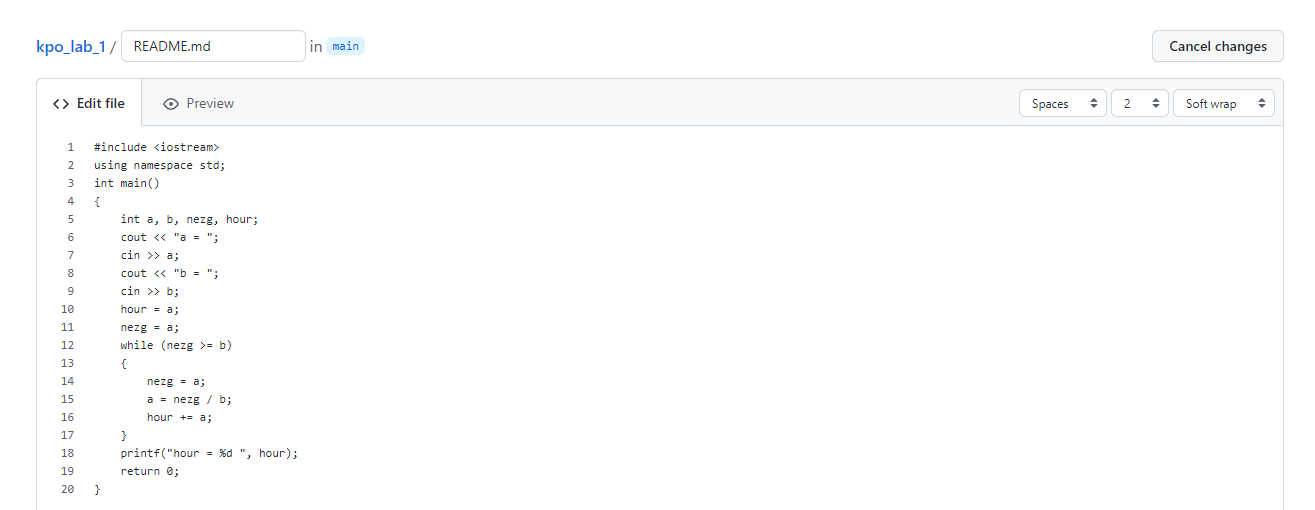
После чего появится вкладка в истории о удалении изменений, выгружаем на сайт.



**Задание 7**

Написать программу в основной ветке

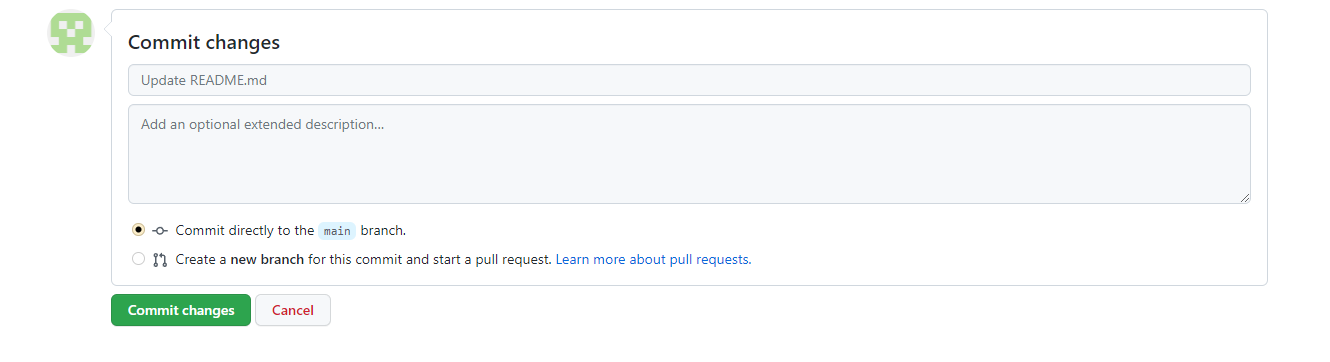
Я не стал создавать новый файл, а редактировав файл README, я написал код



**Задание 8**

Совершить коммит файла с кодом

Проверив правильность кода, я совершил коммит нажав “Сommit changes”.

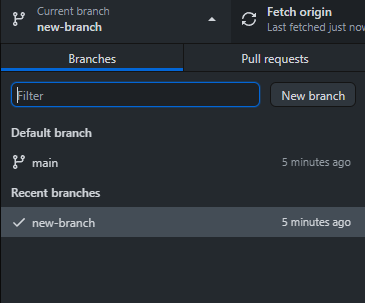




**Задание 9**

Создать новую ветку и переключиться в неё

Для создания новой ветки, необходимо в выбранном репозитории нажать на ветку main и задать новое имя для ветки в соответствующем поле, после чего нажать «new branch».

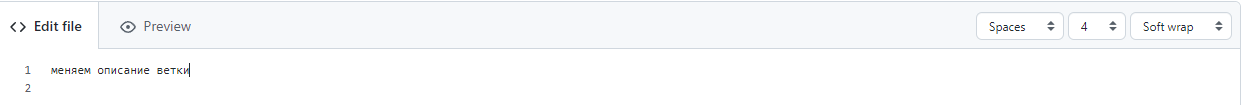


Для переключения между ними надо нажать на ветку main, после чего появятся все существующие ветки.

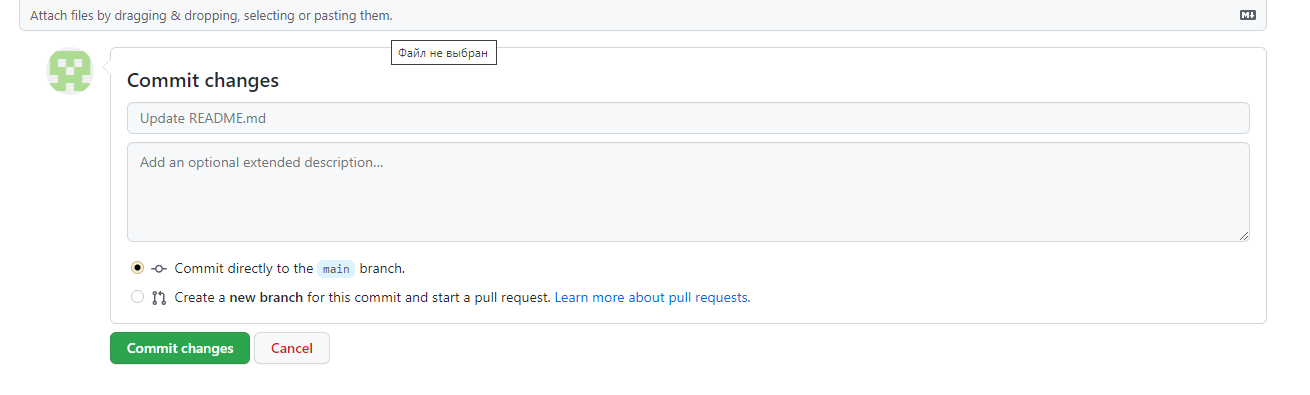
**Задание 10**

Изменить созданную ветку и закоммитить изменения

В выбранной ветке я поменял код на “меняем описание ветки”



И коммитим изменения!



У нас получилось добавить изменения в ветку!



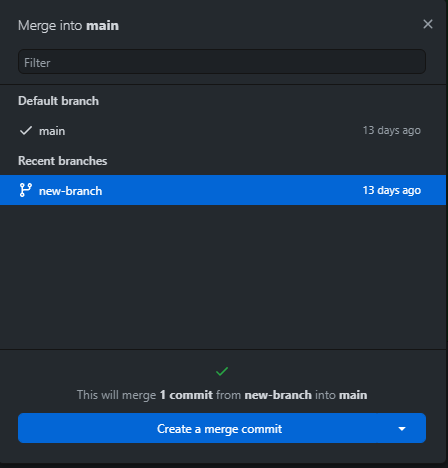
**Задание 11**

Провести слияние с основной веткой

Для слияния ветки «main» и «new branch», необходимо перейти в «current branch» и выбрав ветку main нажать на «choose a branch to merge into main»



Выбираем ветку «new branch» и нажимаем на «create a merge commit»

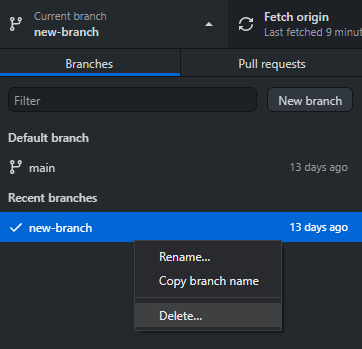


Мы провели слияние с основной веткой

**Задание 12**

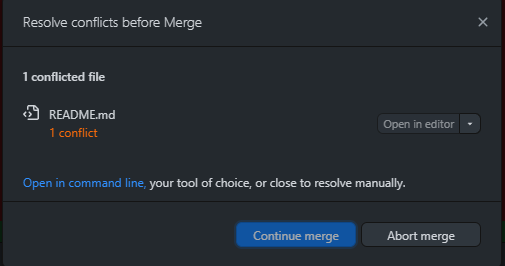
Удаление ненужной после слияния ветки

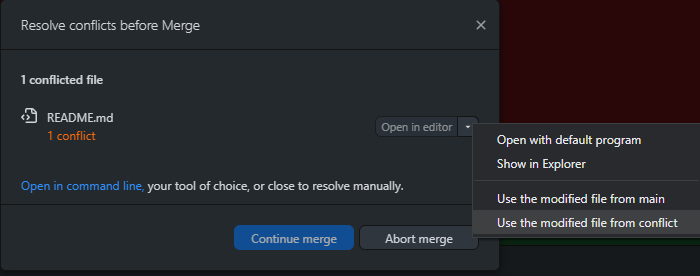
После слияния, мы сразу же можем удалить ненужную нам ветку.

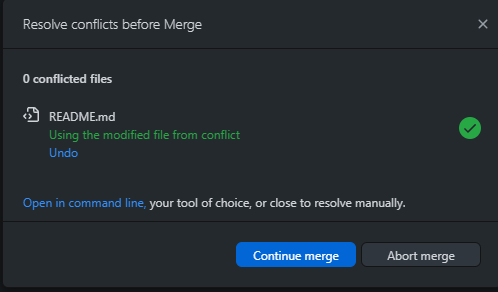


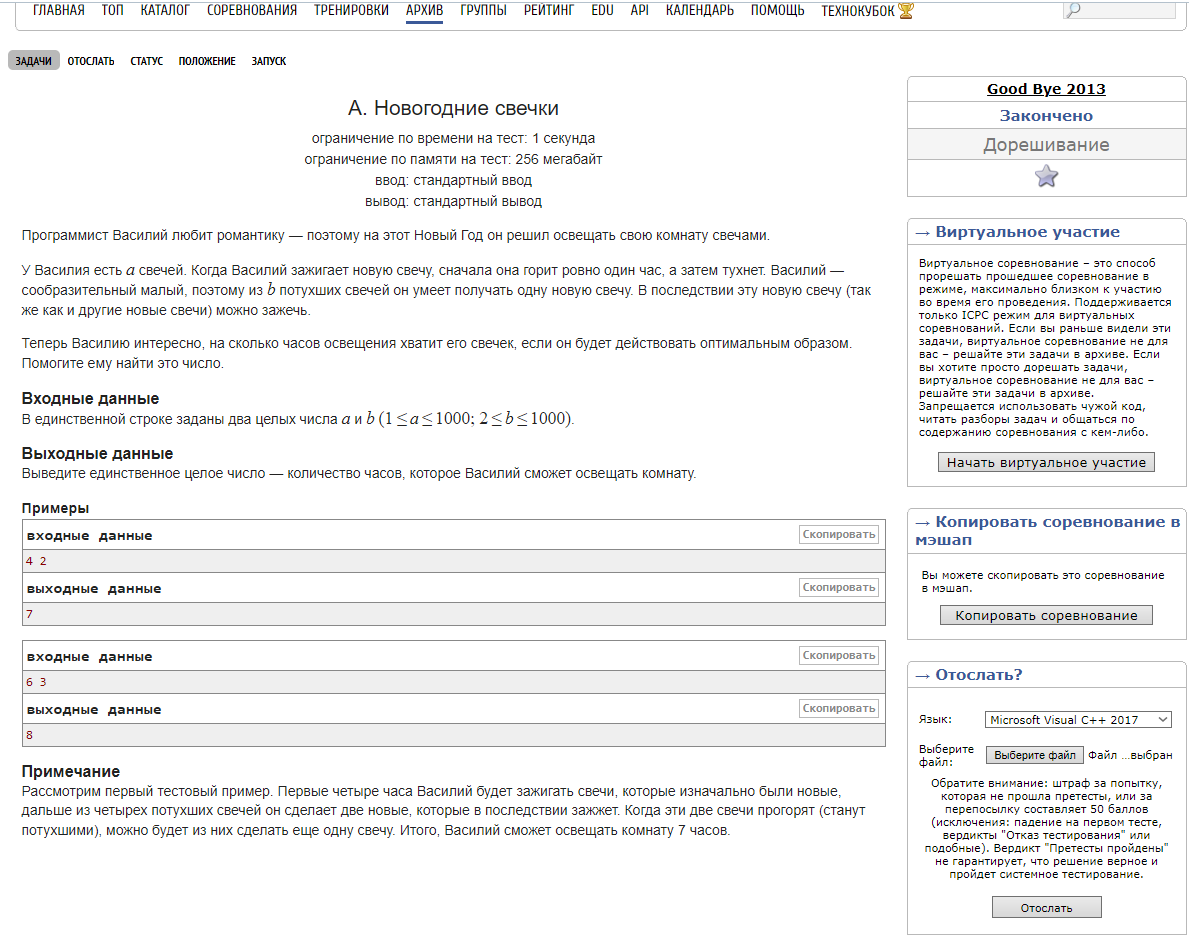
**Задание 13**

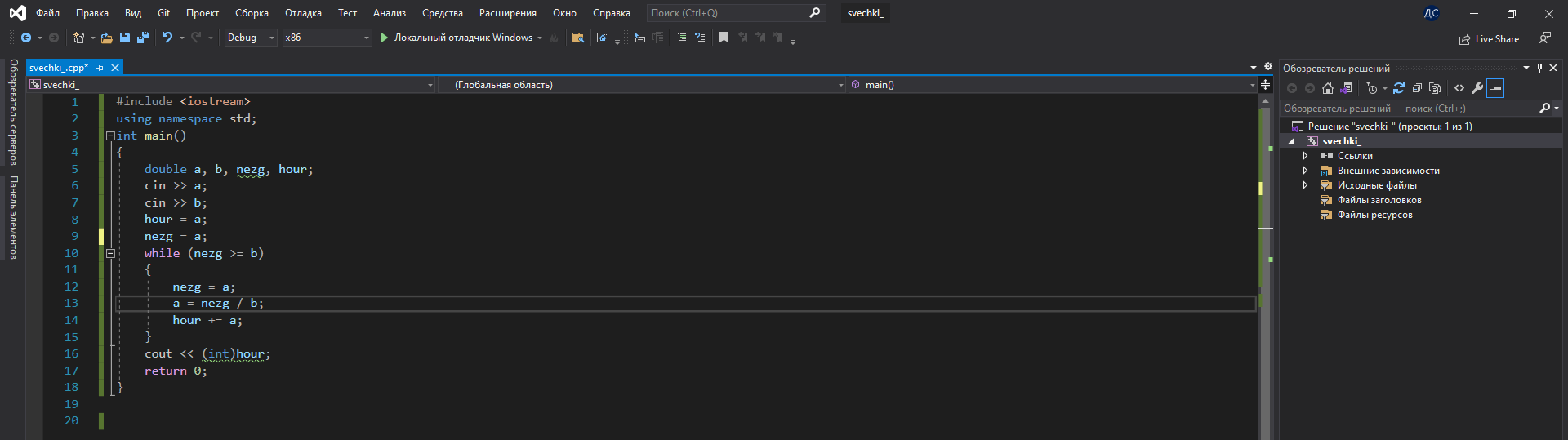
Повторяем пункты 9-11, но теперь так, чтобы при слиянии произошел конфликт. Решаем конфликт

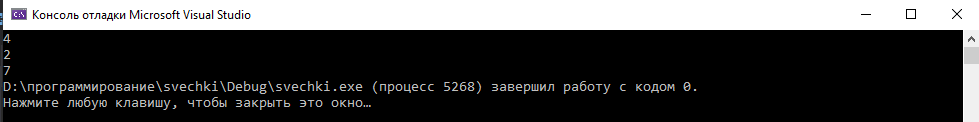


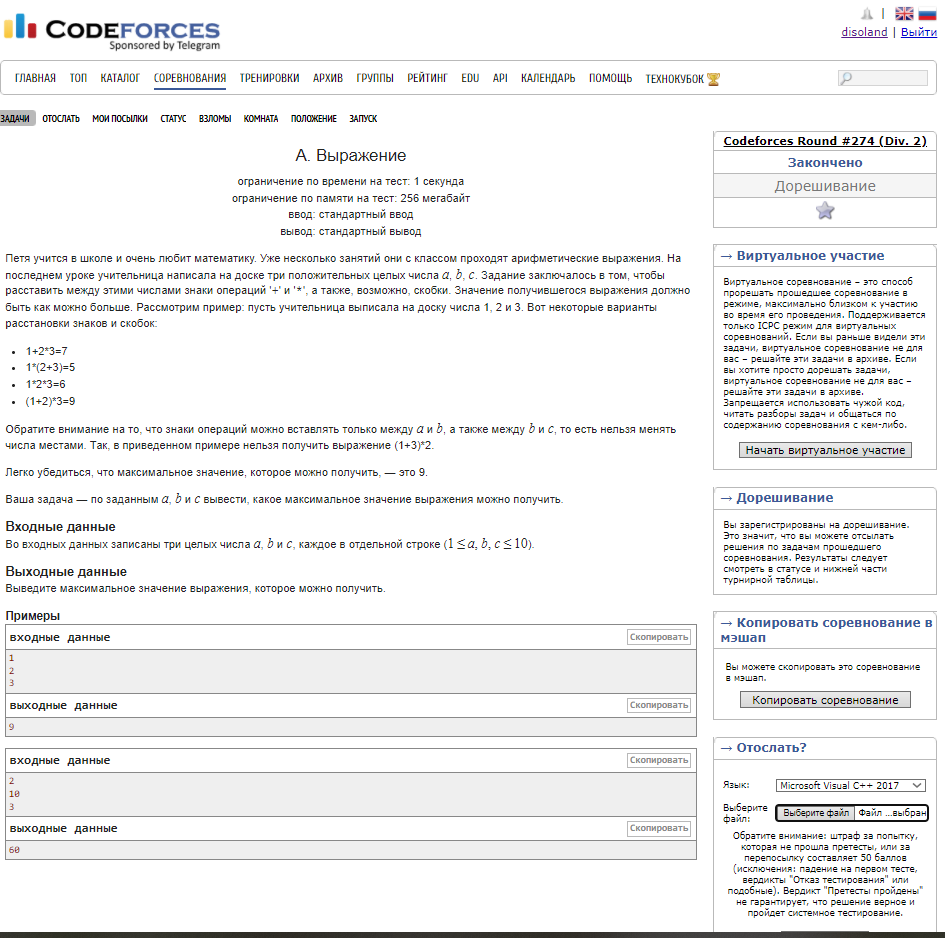


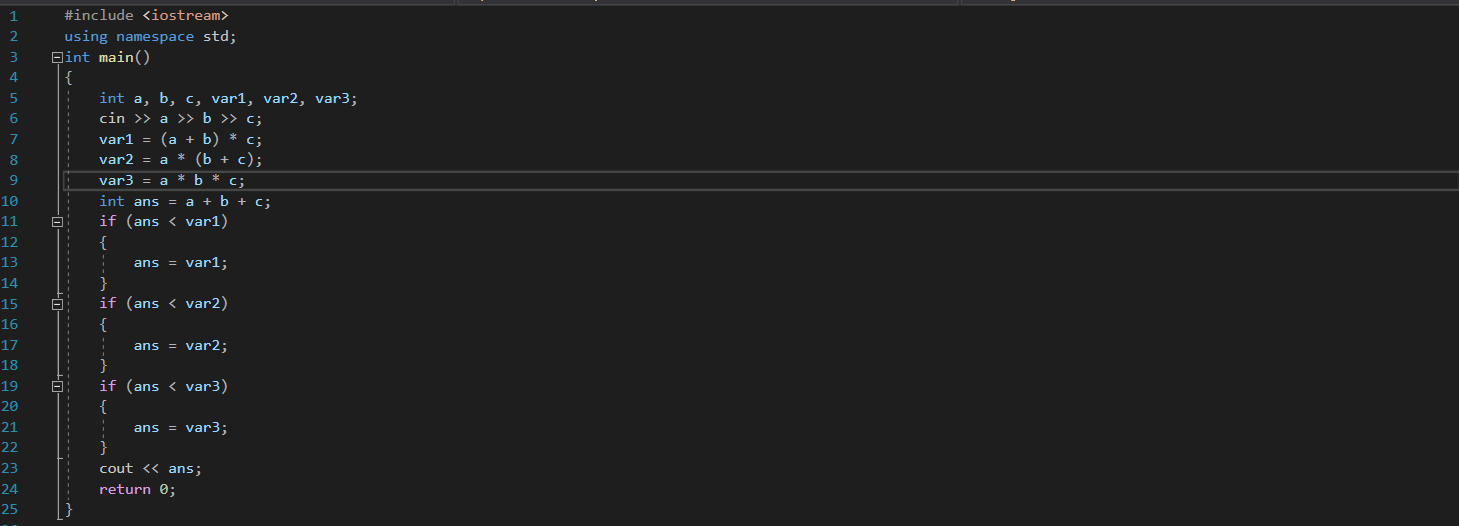


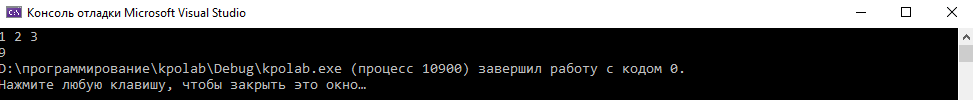


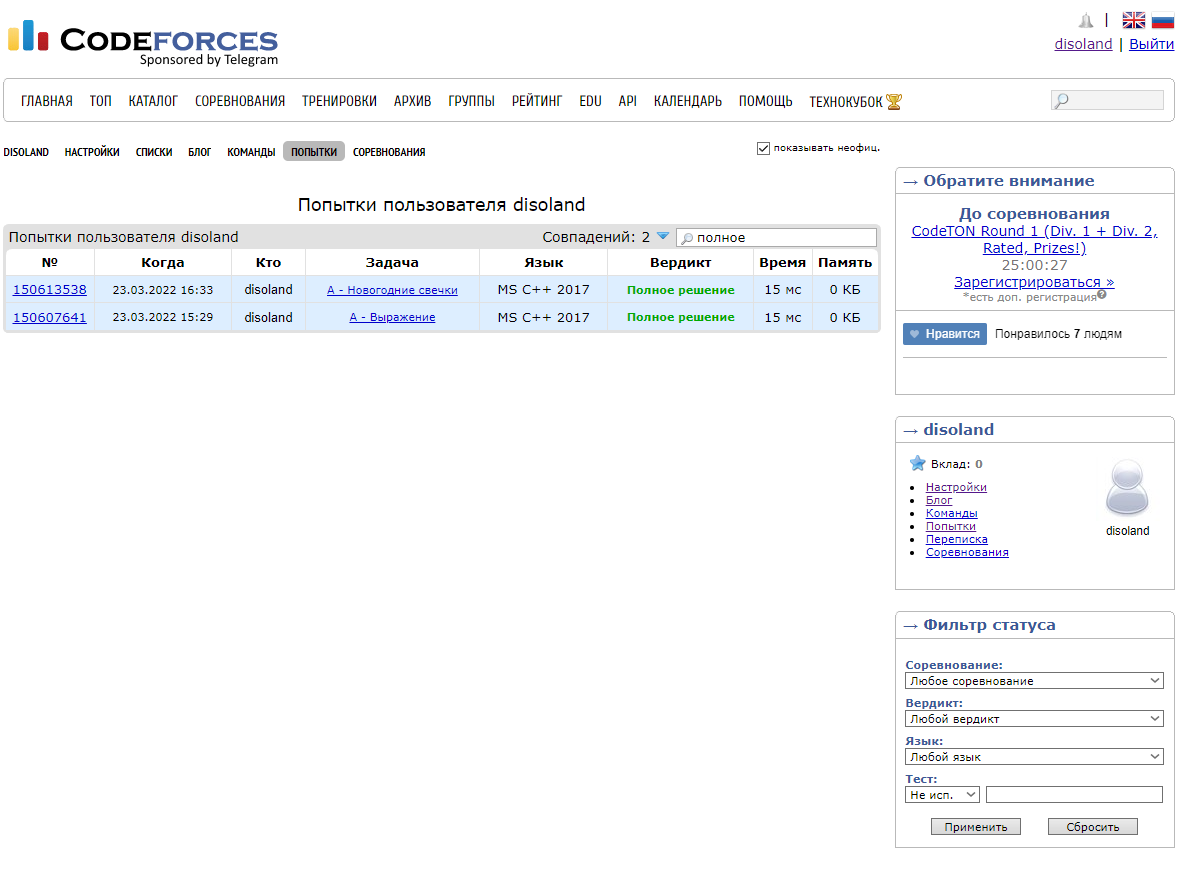










****