

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»

Институт информационных технологий и технологического образования
Кафедра информационных технологий и электронного обучения

Основная профессиональная образовательная программа
Направление подготовки 09.03.01 Информатика и вычислительная техника
Направленность (профиль) «Технологии разработки программного обеспечения»
форма обучения – очная

Вариативное задание 2.1
Практика

Обзор функций GitHub Desktop

Подготовила студентка 2 курса
Нечаева Наталья Андреевна

Санкт-Петербург
2023

Оглавление

Начало работы.....	3
Создание репозитория.....	4
Статус и история репозитория.....	5
Создание файлов.....	6
Публикация в GitHub.....	8
Удаление изменений.....	10
Добавление папки с файлом.....	11
Ветки.....	11
Создание новой ветки.....	11
Удаление файлов из ветки и их слияние.....	12
Удаление ветки.....	13

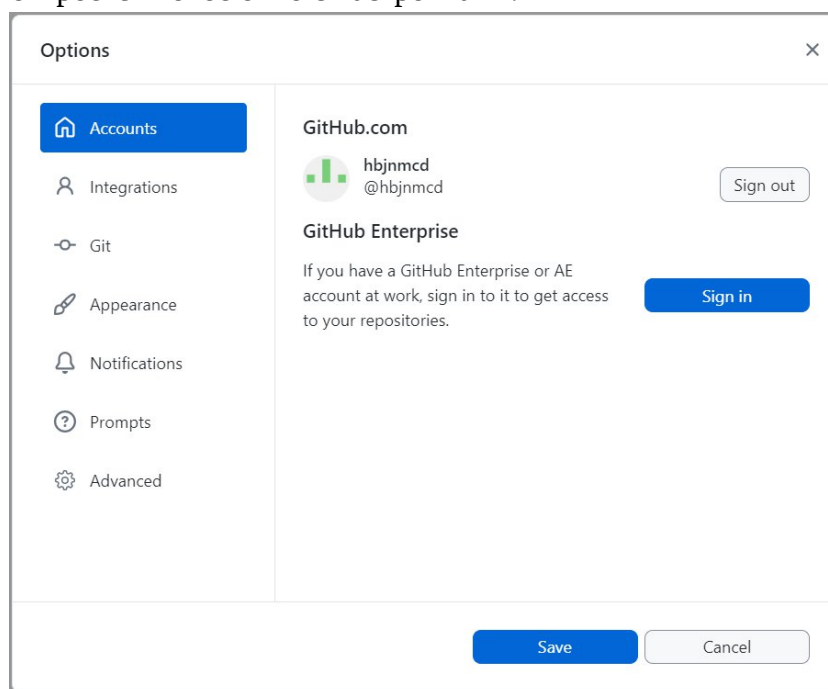
Начало работы

Для скачивания GitHub Desktop необходимо перейти на [сайт](#), после чего выбрать нужную версию и скачать приложение. После открыть его.



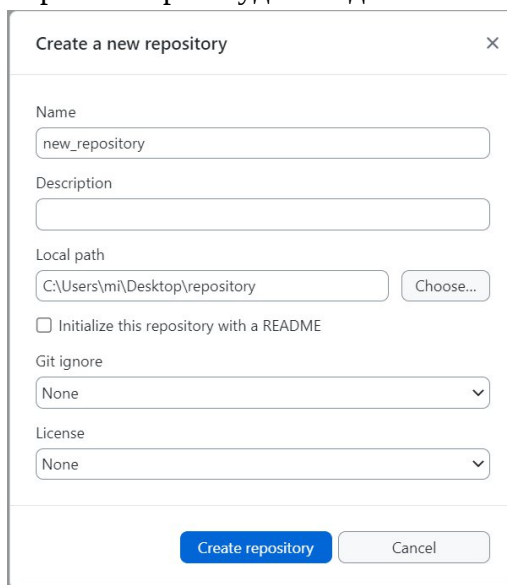
После открытия приложения, необходимо авторизоваться.

Приложение можно настроить под себя, выбрать в верхнем меню File → Options, после чего откроется новое окно с настройками.



Создание репозитория

Для создания нового репозитория в верхнем меню нужно выбрать File → New repository, после чего задать название, описание, и прочие желаемые характеристики. После нажатия *Create repository* создастся репозиторий будет создан.



The image shows a dialog box titled "Create a new repository" with a close button (X) in the top right corner. The dialog contains several input fields and options:

- Name:** A text input field containing "new_repository".
- Description:** An empty text input field.
- Local path:** A text input field containing "C:\Users\mi\Desktop\repository" and a "Choose..." button to the right.
- Initialize this repository with a README:** An unchecked checkbox.
- Git ignore:** A dropdown menu showing "None".
- License:** A dropdown menu showing "None".

At the bottom of the dialog, there are two buttons: "Create repository" (highlighted in blue) and "Cancel".

Статус и история репозитория

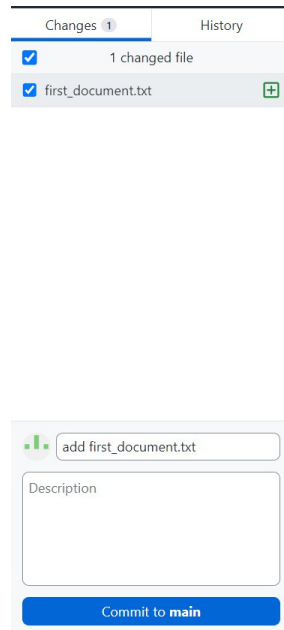
Для просмотра изменений файлов есть *Changes* и *History*, находящиеся под панелью, показывающей, в каком репозитории и какой из его веток сейчас находится пользователь.



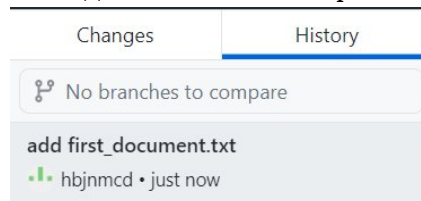
Создание файлов

Для добавления файла в репозиторий необходимо предварительно создать этот файл и поместить его в созданную нами папку репозитория. Этот файл появился в *Changes* и его изменения пока не отслеживаются.

Для того, чтобы это изменить, необходимо *Добавить* этот файл в репозиторий. Для этого ниже, в поле ввода, нужно вписать *add *name**.

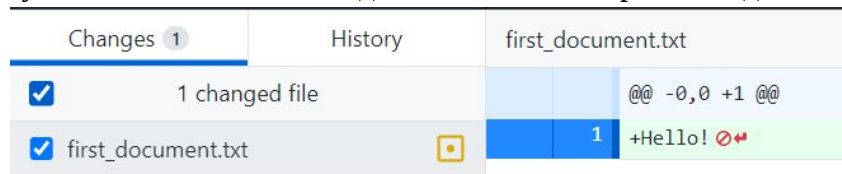


После нажатия *Commit to main* добавление этого файла отразится в истории.

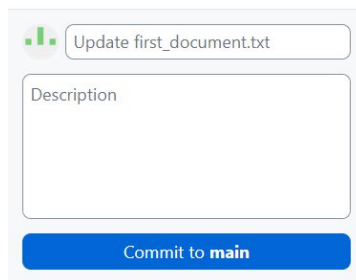


Пока что этот файл пустой, о чем также будет сообщено правее.

При изменении этого файла, его изменение отобразится в *Changes*; его статус будет помечен как *Modified*, а правее будет показано, какие именно изменения в нем произошли. В нашем случае показано, что была добавлена новая строка с надписью.



Для сохранения изменений опять же ниже нужно ввести, что нужно сделать, в нашем случае это *Update *name**.



Update first_document.txt

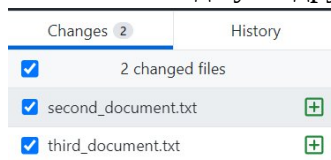
Description

Commit to main

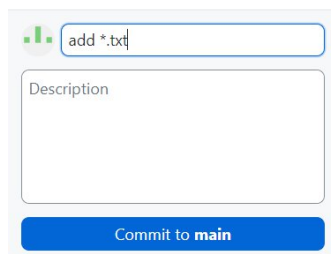
И, аналогично, это появится в Истории, с указанием, кто конкретно произвел изменения и какие, и время, когда произошло изменение.

Бывают случаи, когда есть несколько файлов с одним и тем же расширением, и их всех нужно добавить в репозиторий. Чтобы не добавлять каждый файл вручную, можно воспользоваться маской «*.расширение файла*». Рассмотрим на примере текстовых файлов.

Для начала их нужно создать, в той же папке, что и ранее. После проделать те же действия, что и ранее, но ввести в поле ввода уже другой запрос:



Changes 2	History
2 changed files	
<input checked="" type="checkbox"/> second_document.txt	<input data-bbox="925 891 949 925" type="button" value="+"/>
<input checked="" type="checkbox"/> third_document.txt	<input data-bbox="925 929 949 963" type="button" value="+"/>



add *.txt

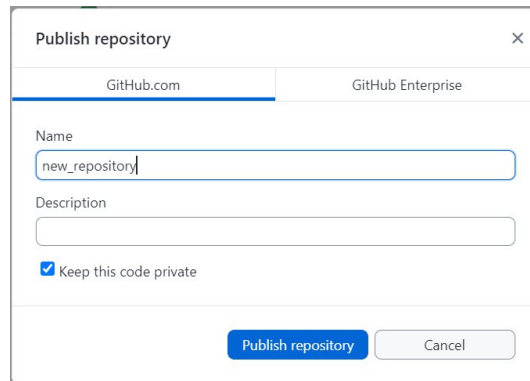
Description

Commit to main

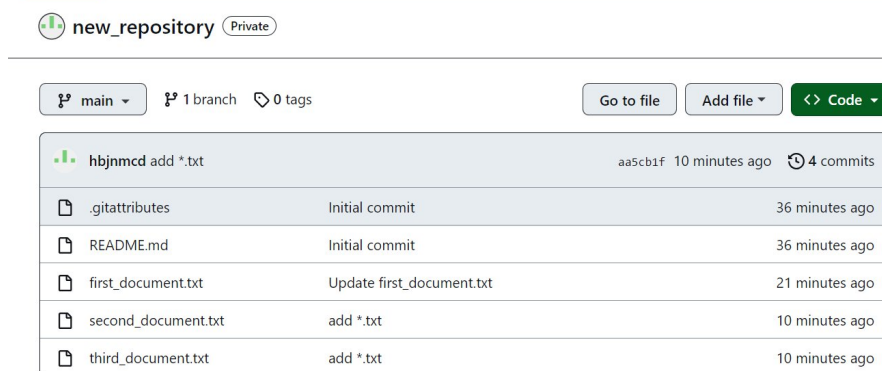
Их добавление также отразится в Истории.

Публикация в GitHub

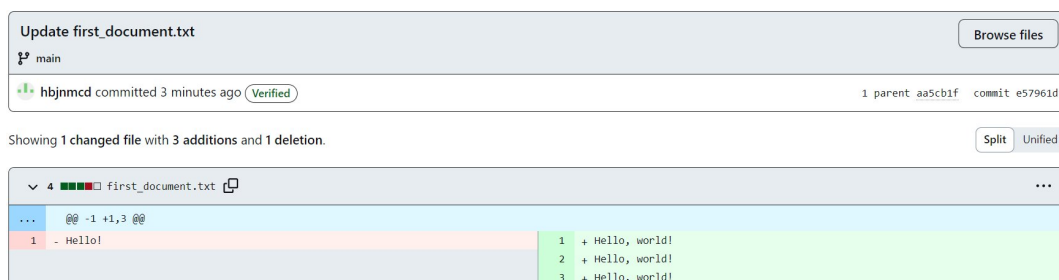
Каждый созданный в GitHub Desktop репозиторий можно опубликовать в своем профиле в GitHub, в том же, с которого была произведена авторизация в скачиваемом приложении. Для публикации репозитория нужно на панели сверху выбрать Publish repository, после в открывшемся окне подтвердить название и описание репозитория и нажать на синюю кнопку.



После этого репозиторий будет опубликован на сайте, причем время изменений и добавлений новых файлов будет соответствовать действительности.



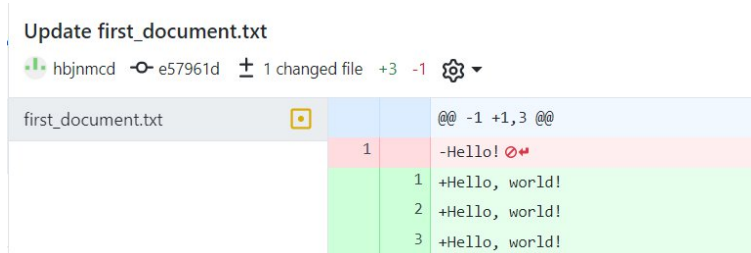
Посмотрим, что будет, если изменить репозиторий на сайте.



Для того, чтобы изменения файлов репозитория с сайта перенеслись на приложение, нужно:

1. Либо через верхнее меню выбрать Repository → Pull;
2. Либо на панели сверху, выбрать Fetch origin (ранее на этом месте была кнопка Publish repository), после чего приложение выведет, что есть изменения в файлах, которые отсутствуют на компьютере, и выбрать Pull origin.

В обоих случаях изменения загрузятся на компьютер и файл изменится, что отразится в Истории:



И наоборот, чтобы изменить репозиторий, находящийся на сайте, после проведения изменений на компьютере, нужно воспользоваться теми же способами.

Удаление изменений

Изменим снова файл, но уже на компьютере. Изменения появятся и в приложении:

Changes 1		History	first_document.txt
1 changed file			@@ -1,3 +1,3 @@
first_document.txt			
	1	-Hello, world!	
	2	-Hello, world!	
	3	-Hello, world!	
	1	+Hi, world!	
	2	+Hi, world!	
	3	+Hi, world!	

Примем эти изменения.

Теперь, чтобы восстановить прошлую версию файла, перейдем в Историю, где после правого щелчка мыши по последнему изменению выберем *Revert changes in commit*.

Changes	History	Update first_document.txt
No branches to compare		hbijnmcd d00b59d 1 changed file +3 -3
Update first_document.txt		first_document.txt
hbijnmcd - just now		
Update first_document.txt		
hbijnmcd - 38 minutes ago		
add *.txt		
		Amend commit...
		Undo commit...
		Checkout commit
		Revert changes in commit

@@ -1,3 +1,3 @@	
1 - Hello, world!	1 + Hi, world!
2 - Hello, world!	2 + Hi, world!
3 - Hello, world!	3 + Hi, world!

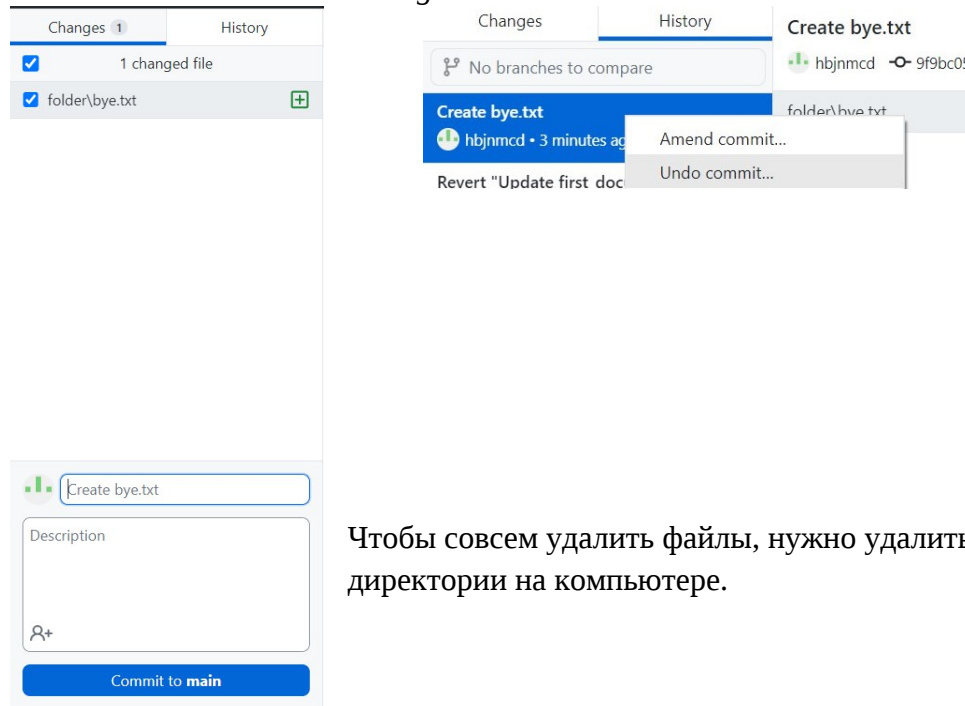
После этого файл изменится на исходный, а в Истории появится соответствующая запись.

Changes	History	Revert "Update first_document.txt"
No branches to compare		hbijnmcd c601929 1 changed file +3 -3
Revert "Update first_document.txt"		This reverts commit d00b59d8dfde6dc3906d83a258ffcbf93c3d5a5d.
hbijnmcd - 1 minute ago		first_document.txt
		@@ -1,3 +1,3 @@
		1 - Hi, world!
		2 - Hi, world!
		3 - Hi, world!
		1 + Hello, world!
		2 + Hello, world!
		3 + Hello, world!

File	Edit	Format	View	Help
Hello, world!				
Hello, world!				
Hello, world!				

Добавление папки с файлом

Сначала нужно создать папку с файлом в нужной директории. После этого это отразится в *Changes*. Можно добавить отслеживание состояния этого файла. Чтобы снять его отслеживание, в *Истории* после щелчка правой кнопкой мыши выбрать *Undo commit...*, после чего он снова появится в *Changes*.

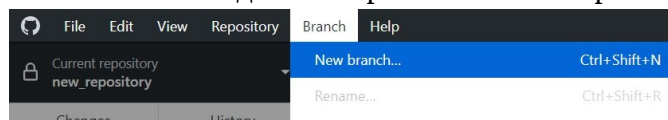


Чтобы совсем удалить файлы, нужно удалить их из директории на компьютере.

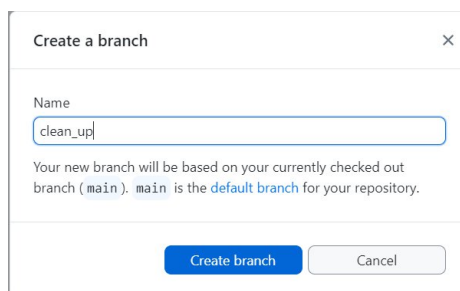
Ветки

Создание новой ветки

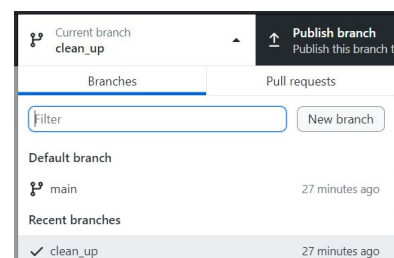
Для создания новой ветки необходимо в верхнем меню выбрать Branch → New branch.



В открывшемся окне вписываем название новой ветки.

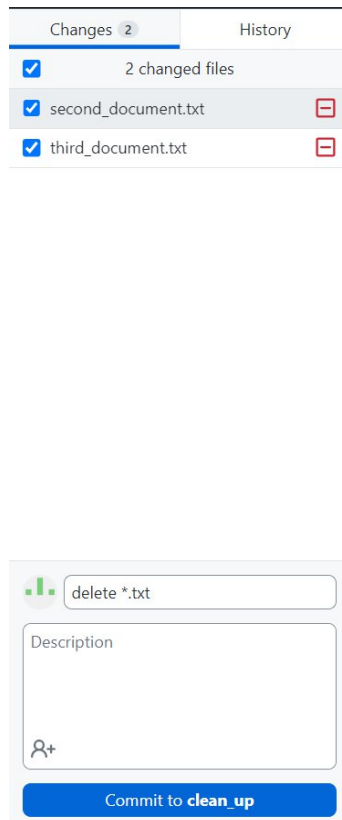


Для переключения между ветками используем кнопку на панели *Current branch*. Кстати, здесь тоже можно создать новую ветку, нажав на кнопку *New branch* слева от поисковой строки.



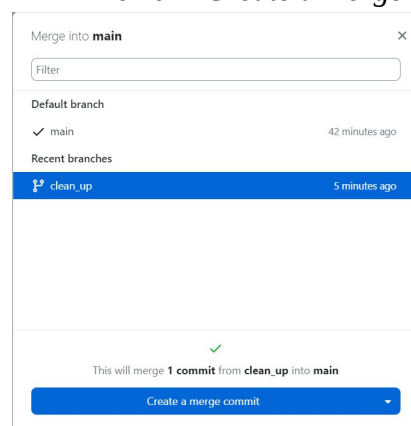
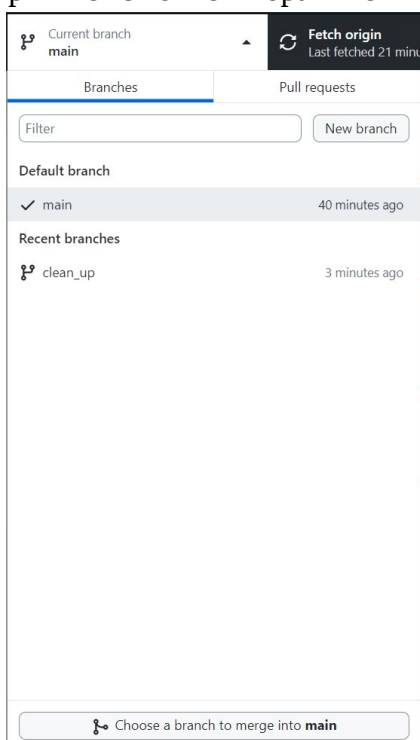
Удаление файлов из ветки и их слияние

Находясь в новой ветке, удалим из директории файлы `second_document` и `third_document`, после чего в приложении зафиксируем это изменение:



Из ветки `clean_up` данные файлы были удалены, но при переходе в основную ветку можно заметить, что файлы там присутствуют, и их Истории отличаются.

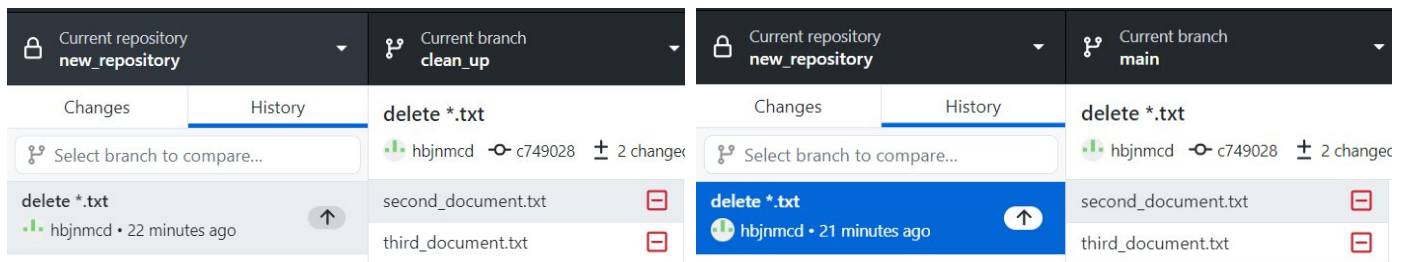
Для того, чтобы слить ветки в одну и тем самым объединить их изменения, нужно, открыв *Current branch*, в самом низу выбрать *Choose a branch to merge into main*, после чего в открывшемся окне выбрать ветки для слияния и нажать *Create a merge commit*.



После чего появится надпись под панелью, сигнализирующая об успехе операции. Теперь и при нахождении в главной ветке файлы удалены.

Successfully merged **clean_up** into **main**

Теперь Истории изменений веток не отличаются.



Удаление ветки

Для удаления ветки нужно выбрать *Current branch*, после чего щелкнуть правой кнопкой мыши по ветке, которую нужно удалить, и выбрать *Delete*.

