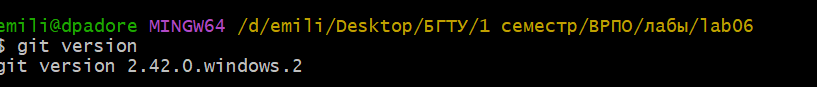
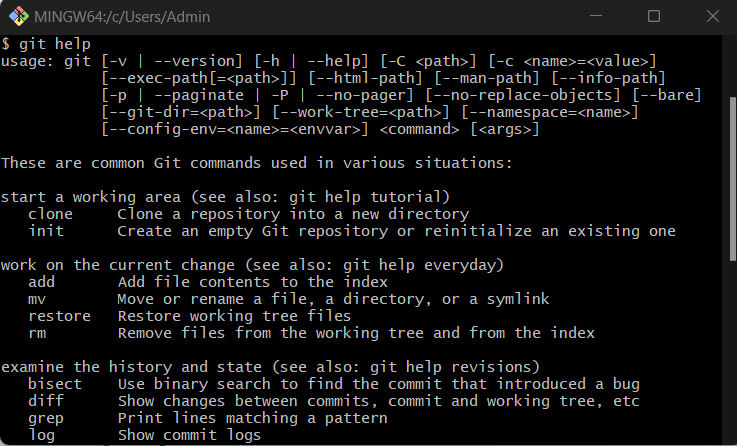
**Лабораторная работа №6**

**4.**

****

**5.**Получите справку по **Git**. Ознакомьтесь со списком командам. $git help



**6.**Создайте локальный репозиторий.

Рабочая папка→ Показать доп.параметры(контексное меню)→ Open GitBash Here.

**7.**Выполните конфигурацию гит.

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

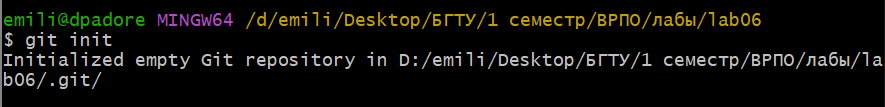
**8.**Проверить настройки конфигурационного файла командой git config --list

Изображение выглядит как текст, снимок экрана, Шрифт

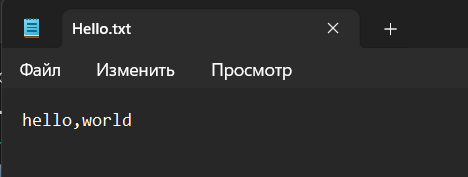
Автоматически созданное описание

**9.**Инициализируйте локальный репозиторий в рабочей папке, где будет он будет размещен.

**git init – создание репозитория**

****

**10.**С помощью блокнота создайте в папке текстовый файл Hello.txt.



**11.**Научитесь получать информацию о статусе (status) репозитория.

**git status – показывает какие файлы изменились между текущей стадией и HEAD. Файлы разделяются на 3 категории: новые файлы, измененные файлы, добавленные новые файлы**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**12.** Проиндексируйте файл Hello.txt

**git add <имена файлов> – добавляет файлы в индекс**

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание**

**13.**Зафиксируйте изменения индексированных файлов в репозитории.

**git commit – выполняет коммит проиндексированных файлов в репозиторий**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**14.**Просмотрите журнал с помощью команды git log

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**15.**Научитесь получать информацию о проиндексированных и неиндексированных файлах. Для этого измените существующий файл Hello.txt, добавьте в репозиторий новый файл и исследуйте состояние репозитория. Зафиксируйте изменения в репозитории.

git status – показывает какие файлы изменились между текущей стадией и HEAD.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**17.**Ознакомиться с шаблонами .gitignore и настройте этот файл для вашего репозитория.

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**18.**Создайте (branch) новую ветку, например, test и переключитесь (checkout) в нее.

Изображение выглядит как текст, снимок экрана, Шрифт

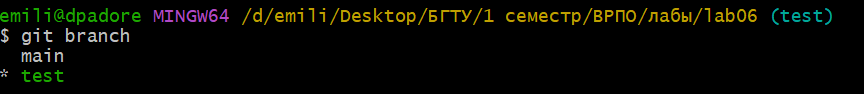
Автоматически созданное описание

**19.**Создайте новый текстовый файл test.html и зафиксируйте его (commit)

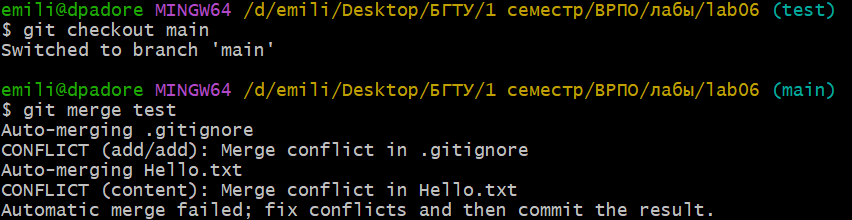
**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**20.**Просмотрите список всех веток и определите текущую ветку.



**21.**Выполните слияние (merge) веток



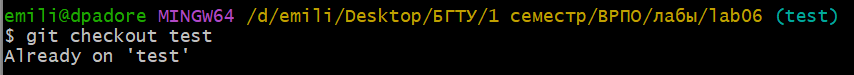
**22.**По очереди отредактируйте файл test.html в ветках master и test. Выполните commit. Выполните слияние в одной и второй ветках. Разрешите ситуацию конфликта слияний.

1.Изменить файл в ветке master

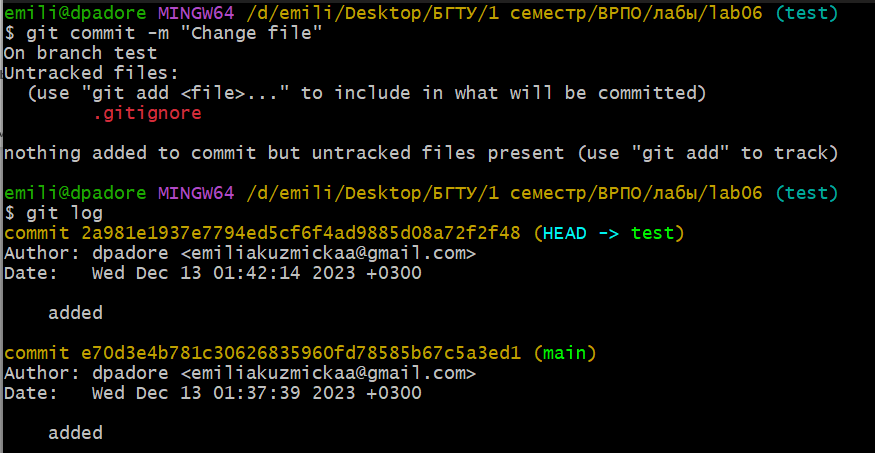
3. Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

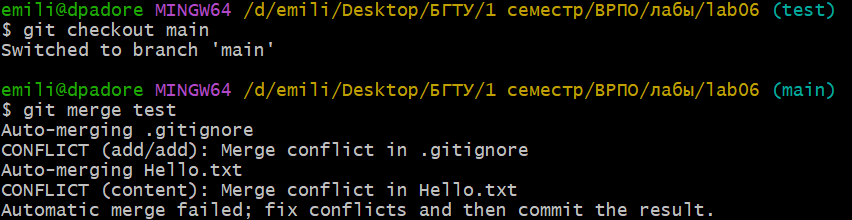
4.Изменения в ветке test



5.При переходе в ветку master мы получаем сообщение о том, что у нас есть измененный файл и мы не можем просто так изменить состояние с одной ветки на другую. Для этого мы будем вынуждены индексировать этот файл и после чего выполнить commit

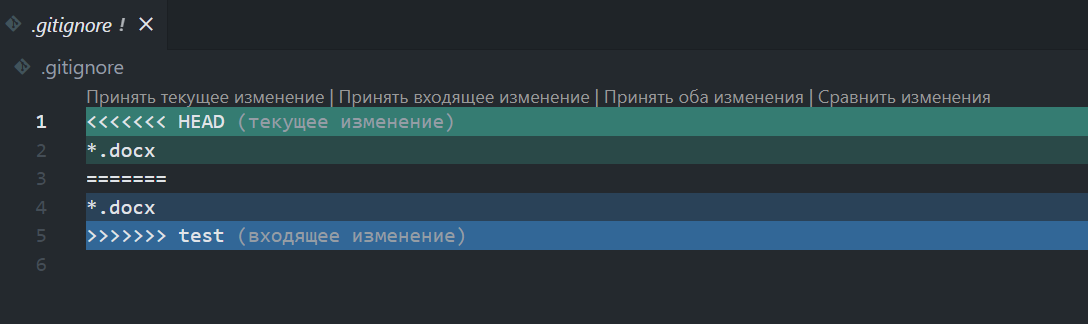


6.Потом переходим в master, выполняем слияние и получаем конфликт



7.Переходим в test.html

Здесь мы должны отредактировать файл так, как мы хотим видеть его в финальном результате. До ===== показывается, что было в главной ветке, а после ==== что поменялось в ветке test. Затем редактируем файл как нам нужно и сохраняем



8.Выполнив $ git status мы увидим подсказку (use «git merge –abort» для того, чтобы отменить слияние или добавить уже отредактированный файл на индексацию):



9.Для выхода используем wq (записать и выйти).