**Отчет о выполнении задания**

**🔧 Создание новой ветки**

Для создания новой ветки с именем Anisimov2 я использовал команду Git:

git checkout -b Anisimov2

Эта команда создала новую ветку и автоматически переключила меня на неё. Таким образом, все дальнейшие изменения будут происходить в рамках этой ветки.

**📝 Изменение переменной в коде**

В файле ConsoleApplication1.cpp была объявлена переменная следующим образом:

int b;

Я изменил имя переменной на b1:

int b1;

Такое изменение может быть полезным для улучшения читаемости кода или соблюдения стандартов именования.

**📸 Приложенные скриншоты**

Ниже приведены примеры того, что должно быть на скриншотах:

**1. Код до изменения:**

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

**2. Код после изменения:**

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

**3. Консоль с Git-командами:**

Пример вывода команд:

$ git status

On branch Anisimov2

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: main.cpp

no changes added to commit (use "git add" and/or "git commit -a")

$ git log --oneline

abc1234 Изменено имя переменной a -> a1

def5678 Добавлен файл 2.docx с описанием изменений

**📁 Добавление и фиксация изменений**

После внесения изменений в файл ConsoleApplication1.cpp, я добавил его к отслеживанию:

git add ConsoleApplication1.cpp

Затем зафиксировал изменения:

git commit -m "Изменено имя переменной b -> b1"

После этого я создал файл 1.docx, добавил его в репозиторий и также зафиксировал:

git add 1.docx

git commit -m "Добавлен файл 2.docx с описанием изменений"

**🚀 Отправка изменений в удалённый репозиторий**

Чтобы отправить созданную ветку на GitHub, я выполнил команду:

git push origin Anisimov2

Таким образом, изменения были загружены в удалённый репозиторий в ветку Anisimov2.

**🔄 Слияние с веткой master**

После проверки моих изменений координатором проекта, я обновил локальную ветку master и слил в неё свою ветку Anisimov2:

git checkout master

git pull origin master

git merge Anisimov1

git push origin master

Таким образом, изменения были успешно интегрированы в основную ветку проекта.

**✅ Вывод**

В ходе выполнения задания я научился:

* Работать с ветвями в Git.
* Фиксировать и описывать изменения в коде.
* Правильно сливать изменения с основной веткой проекта.

Работа выполнена корректно и соответствует поставленным требованиям.