МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ

образовательное учреждение

высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра защиты информации



**ЛАБОРАТОРНАЯ РАБОТА №1**

**«Работа с Git и GitHub»**

**по дисциплине: «*Программирование*»**

Выполнил:Проверил:

Студент гр. «АБ-420», «АВТФ» *Ассистент кафедры ЗИ*

*Твердохлеб Андрей Сергеевич Исаев Г. А.*

«\_\_» \_\_\_\_\_\_ 2025г«\_\_\_» \_\_\_\_\_\_ 2025 г.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (подпись)

Новосибирск 2025

**Цели и задачи работы**: изучение платформы GitHub и практическое использование Git для хранения, обновления и распространения исходного кода проекта.

**Задание №1 к работе**: Разработать алгоритм решения задачи по индивидуальному заданию.

Геометрические фигуры. Реализовать функции вычисления параметров для следующих геометрических фигур:

1. Трапеция – периметр, площадь, длина средней линии;

**Методика выполнения работы**:

1. Разработать алгоритм решения задачи по индивидуальному заданию.
2. Написать и отладить программу решения задачи.
3. Протестировать работу программы на различных исходных данных.
4. Продемонстрировать работу с Git и GitHub на примере написанной программы.
5. По запросу преподавателя быть готовым модифицировать/добавить функционал программы.
6. Ответить на теоретические вопросы к лабораторной работе на выбор преподавателя.

**Листинг программы**:

Ссылка на репозиторий: https://github.com/DEVILxAndrey/LabGit

Trapecia.cpp

#include <iostream>

#include <cmath>

using namespace std;

// Функция для вычисления периметра

double perimeter(double a, double b, double c, double d) {

return a + b + c + d;

}

// Функция для вычисления длины средней линии

double midLine(double a, double b) {

return (a + b) / 2;

}

// Функция для вычисления площади через высоту

double heightarea(double a, double b, double c, double d) {

double midLine = (a + b) / 2;

return midLine \* sqrt((pow(c, 2) - (pow((pow(a - b, 2) + pow(c, 2) - pow(d, 2)) / (2 \* (a - b)), 2))));

}

int main() {

setlocale(LC\_ALL, "rus");

double a, b, c, d;

cout << "Введите параметры сторон трапеции (сначала основания, затем стороны): ";

cin >> a >> b >> c >> d;

// Вывод данных

cout << "Периметр: " << perimeter(a, b, c, d) << endl;

cout << "Средняя линия: " << midLine(a, b) << endl;

cout << "Площадь: " << heightarea(a, b, c, d) << endl;

}

**Результат работы программы:**

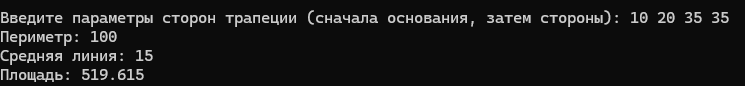


Рис. 1 Результат работы программы

**Полное описание всех этапов процесса работы:**

1. Создание репозитория в GitHub
2. Клонирование репозитория в рабочую папку, открытие проекта в IDE (VisualStudioCode).
3. Создание отдельной ветки “Tverdokhleb” и файла Trapecia.cpp, в котором будет написан код.
4. Написание кода, доработка, сохранение, тестирование, отладка.
5. Выполнение слияния дополнительной ветки “Tverdokhleb” с файлом Trapecia.cpp в основную “main”.
6. Сохранение и загрузка в репозиторий.
7. Оформление файла [README.md](https://vk.com/away.php?to=http%3A%2F%2FREADME.md&cc_key=).

**Описание использованных функций :**

1. git clone <SSH ссылка на репозиторий> - Клонирование репозитория

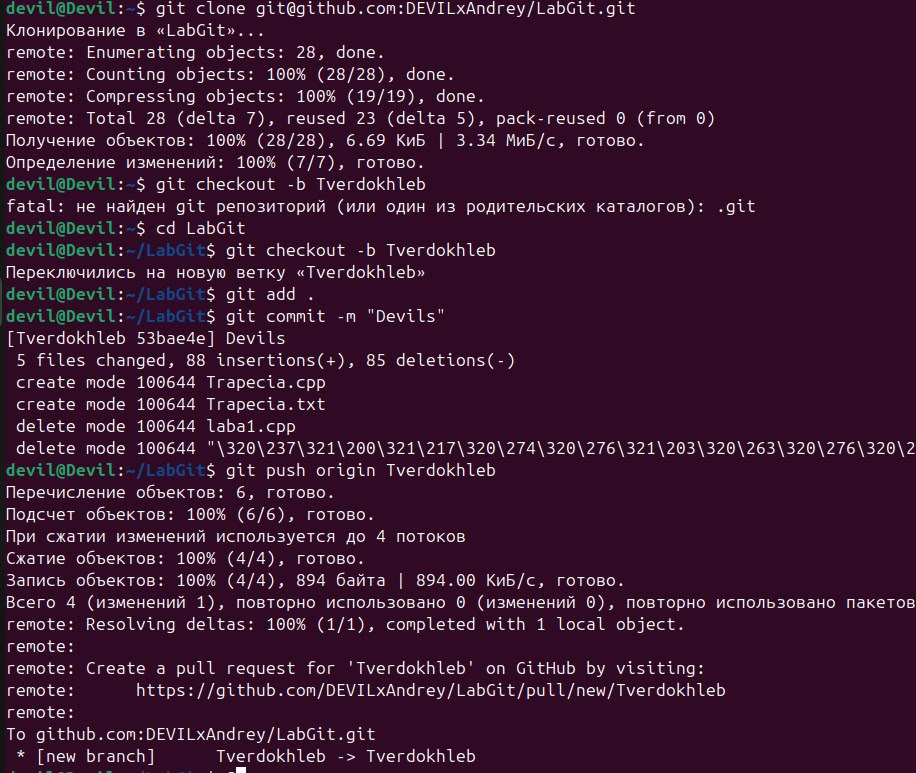


Рис. 2 git clone

2. git checkout –b <ветка> - создаётся новая локальная ветвь и начинает отслеживать удалённую

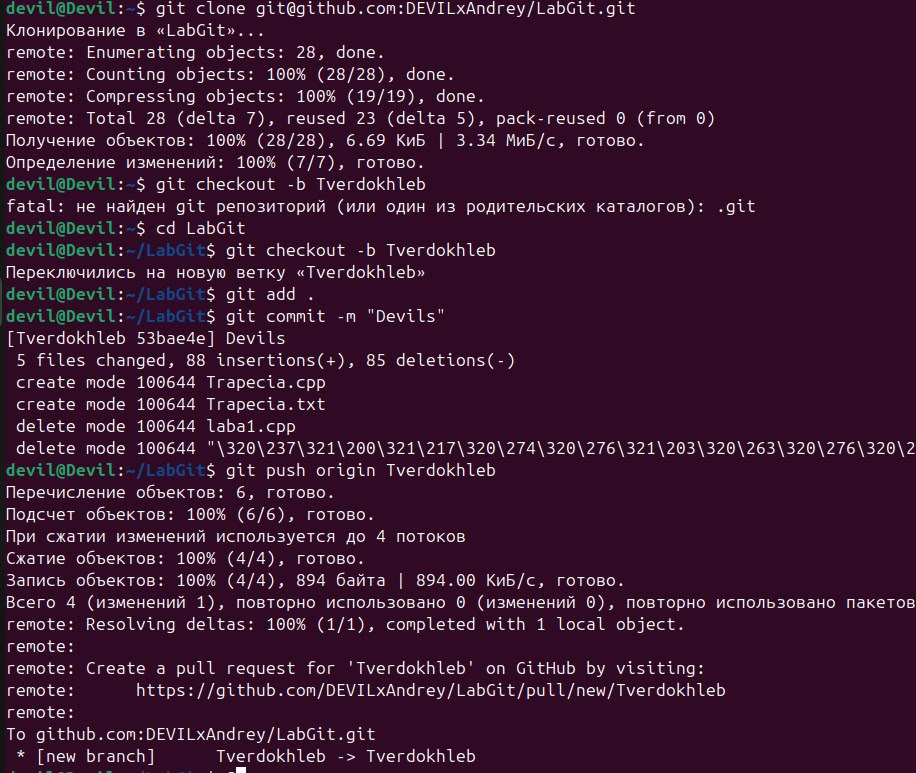


Рис. 3 git checkout

3. git add <название файлов> - сообщает Git, что вы хотите включить изменения в конкретном файле в следующий коммит.

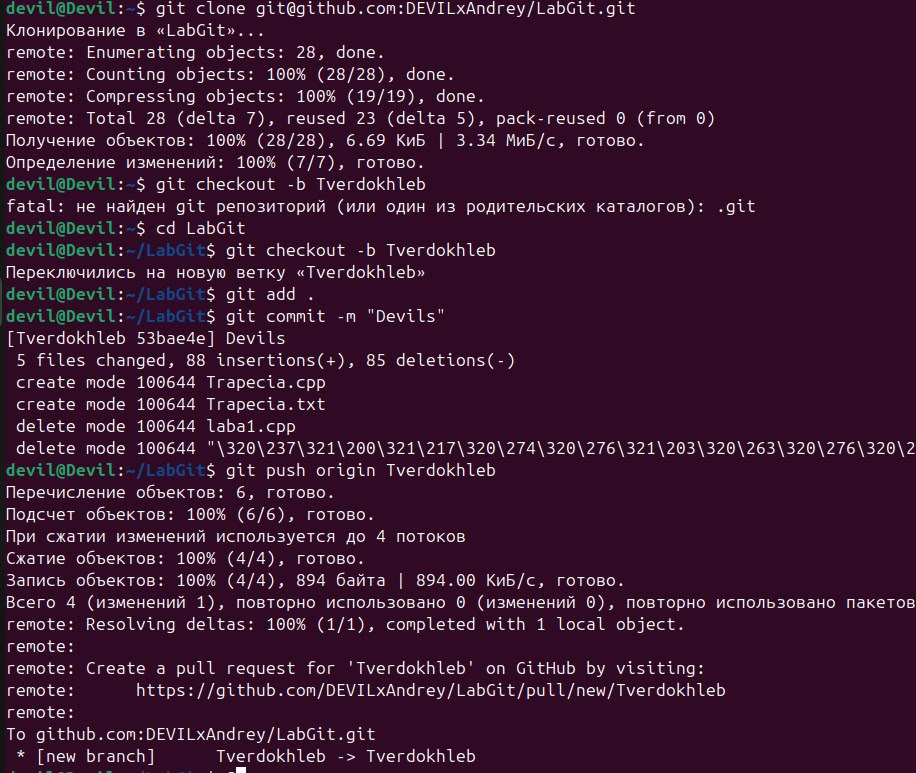


Рис. 4 git add

4. git commit -m <комментарий> - добавление коммита, фиксация изменений на ветке.

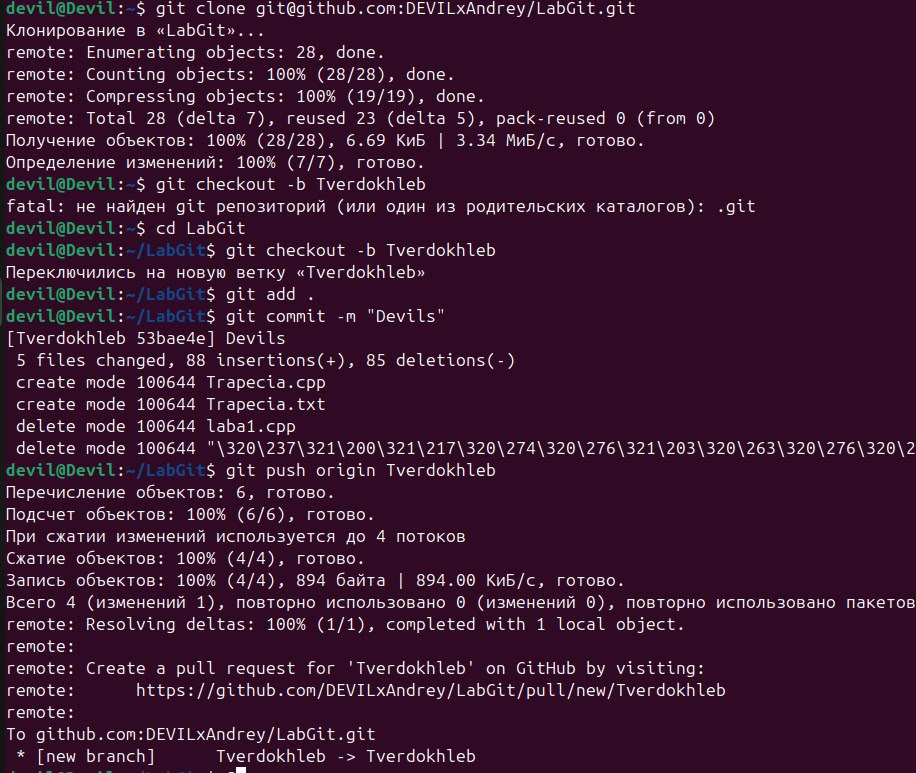


Рис. 5 git commit

6. git checkout main – перемещение в главную ветку.

7. git merge <ветка откуда сливаются изменения> - слияние веток.

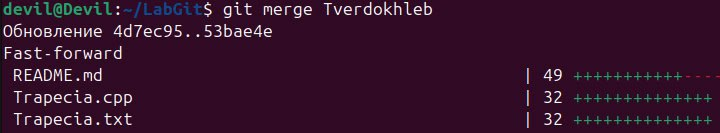


Рис. 6 git merge

8. git push <удалённый репозиторий> <ветка> — загрузка из локального репозитория в удалённый.

9. git pull <удалённый репозиторий> <ветка> — загрузка из удалённого репозитория в локальный.

10. git branch — список веток

**Вывод**

В процессе работы была изучена платформа GitHub и практическое применение системы контроля версий Git. Были освоены ключевые принципы работы с репозиториями, включая их создание, клонирование, управление ветками, слияние изменений и обновление проекта. Все поставленные задачи были успешно выполнены в рамках командного проекта.