

Министерство науки и высшего образования РФ
Федеральное государственное бюджетное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Кафедра «Информационная безопасность систем и технологий»

Отчет
по практической работе №2
«Реализация интерфейса с параметрами в командной строке»

Дисциплина: Методы и средства
программирования

Группа: 22ПИ2

Выполнила: _____ Рыжманова Е.Д.

Отметка о сдаче (количество баллов):

Дата сдачи:

Принял: Сидоров Н.А.

2023

1 Цель работы является получить навыки по обработке данных из командной строки.

2 Задания к лабораторной работе.

2.1 Создать на своём аккаунте на github.com репозиторий с именем TxtIntr для выполнения данного задания и сделайте его локальную копию.

2.2 Создайте в репозитории каталог calculator для реализации программы калькулятора. Подготовьте в каталоге файлы .gitignore для управления исключениями и MakeFail для управления сборкой.

2.3 Разработайте и реализуйте программу-калькулятор, получающую входные данные из командной строки и выполняющую заданные вычисления. Калькулятор должен выполнять заданный набор операций над одним или несколькими операндами. Операция должна задаваться именованным оператором, например -o summa или -operation summa. Операнды должны задаваться позиционными безымянными параметрами, например 45 13 -2 Калькулятор должен выполнять за один запуск одну операцию над заданным набором операторов. Калькулятор при запуске без параметров должен выдавать справку по использованию.

2.4 Протестируйте работоспособность разработанной программы на различных тестовых примерах.

2.5 Сделайте отчёт о выполненной работе, включив в него ссылку на свой репозиторий на github.com.

3 Ход работы:

3.1 Был создан репозиторий с именем TxtIntr для выполнения данной практической работы на сайте github.com, а так же была сделана копия на ПК. Результат на рисунке 1.

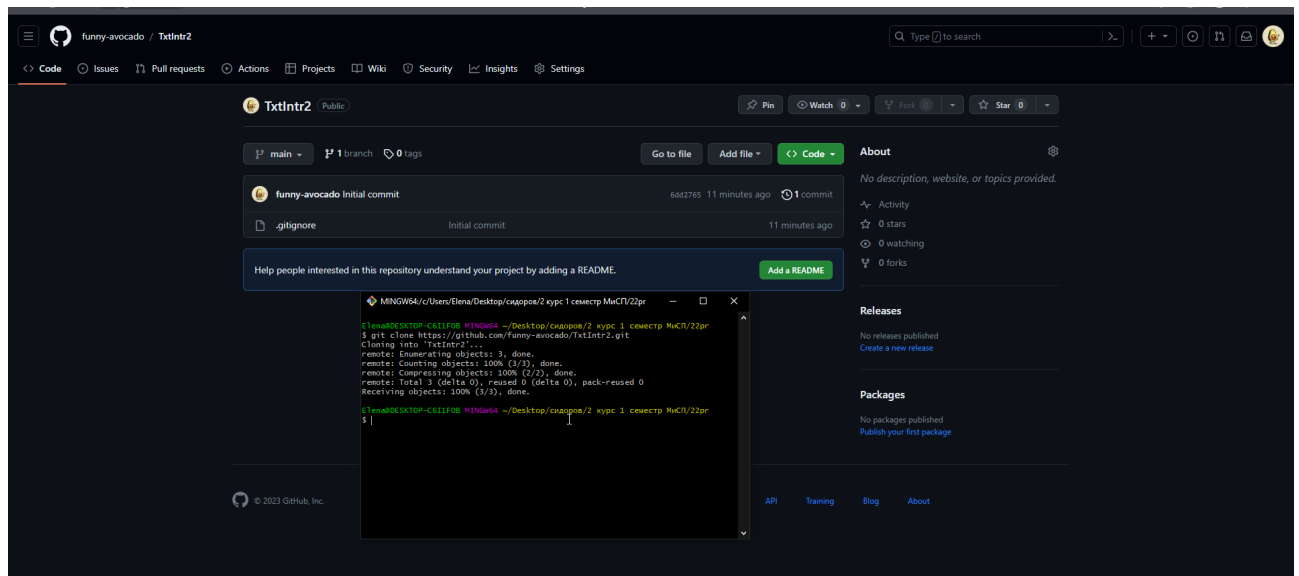
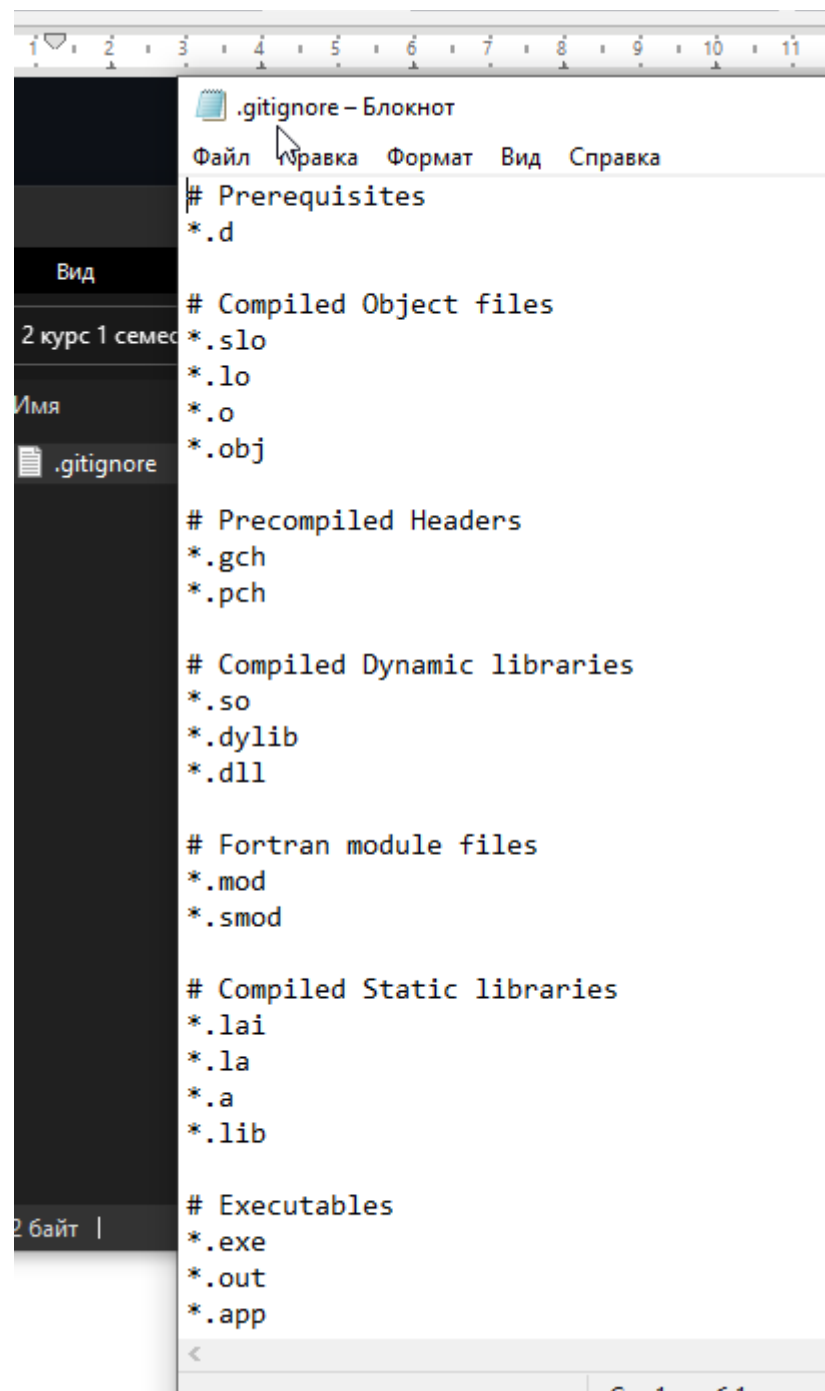


Рисунок 1 – Создание и клонирование репозитория на ПК

3.2 В репозитории был создан каталог calculator для реализации программы калькулятор.

3.2.1 Так же были подготовлены в каталоге файлы .gitignore для управления исключениями. Результат работы представлен на рисунке 2 и приложении А.



```
.gitignore – Блокнот
Файл  Правка  Формат  Вид  Справка
# Prerequisites
*.d

# Compiled Object files
*.slo
*.lo
*.o
*.obj

# Precompiled Headers
*.gch
*.pch

# Compiled Dynamic libraries
*.so
*.dylib
*.dll

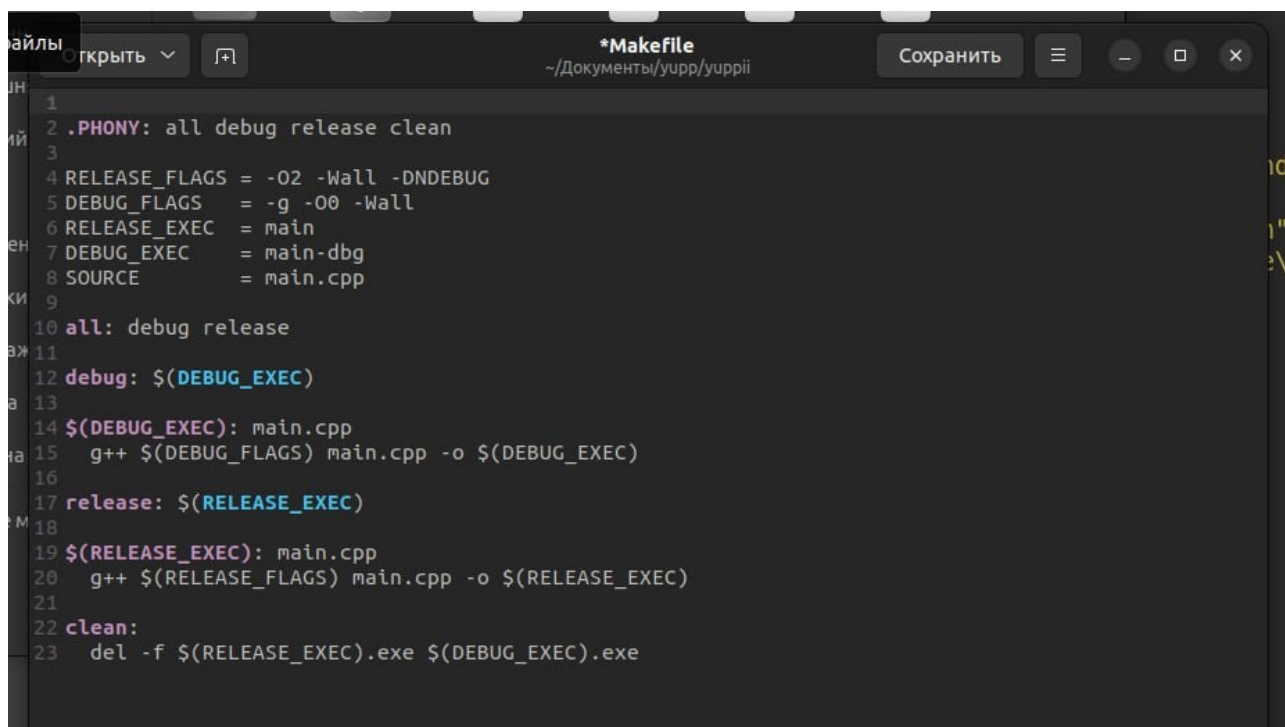
# Fortran module files
*.mod
*.smod

# Compiled Static libraries
*.lai
*.la
*.a
*.lib

# Executables
*.exe
*.out
*.app
```

Рисунок 2 – Файл .gitignore

3.2.2 Был создан MakeFail для управления сборкой, результат на рисунке 3 и приложении Б.

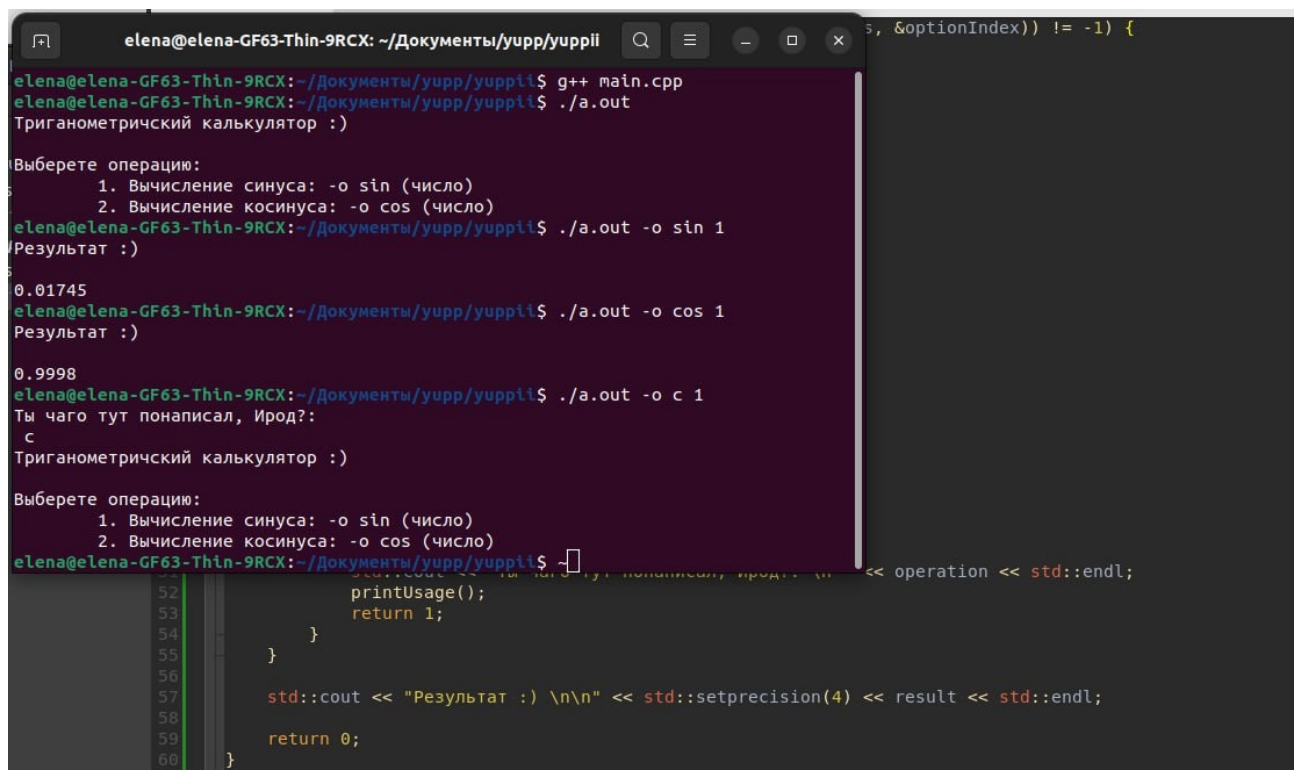


```
1
2 .PHONY: all debug release clean
3
4 RELEASE_FLAGS = -O2 -Wall -DNDEBUG
5 DEBUG_FLAGS   = -g -O0 -Wall
6 RELEASE_EXEC  = main
7 DEBUG_EXEC    = main-dbg
8 SOURCE        = main.cpp
9
10 all: debug release
11
12 debug: $(DEBUG_EXEC)
13
14 $(DEBUG_EXEC): main.cpp
15     g++ $(DEBUG_FLAGS) main.cpp -o $(DEBUG_EXEC)
16
17 release: $(RELEASE_EXEC)
18
19 $(RELEASE_EXEC): main.cpp
20     g++ $(RELEASE_FLAGS) main.cpp -o $(RELEASE_EXEC)
21
22 clean:
23     del -f $(RELEASE_EXEC).exe $(DEBUG_EXEC).exe
```

Рисунок 3 – Makefail

3.3 Был разработан и реализован тригонометрический калькулятор, получающий входные данные из командной строки и выполняющий операции \sin и \cos (режим радиусов и радиан). Результат представлен в приложении В.

3.4 Была проверена работоспособность разработанной программы различными тестовыми примерами, результат на рисунке 4.



```
elena@elena-GF63-Thin-9RCX: ~/Документы/yurp/yurpii
elena@elena-GF63-Thin-9RCX:~/Документы/yurp/yurpii$ g++ main.cpp
elena@elena-GF63-Thin-9RCX:~/Документы/yurp/yurpii$ ./a.out
Тригонометрический калькулятор :)

Выберете операцию:
1. Вычисление синуса: -o sin (число)
2. Вычисление косинуса: -o cos (число)
elena@elena-GF63-Thin-9RCX:~/Документы/yurp/yurpii$ ./a.out -o sin 1
Результат :)
0.81745
elena@elena-GF63-Thin-9RCX:~/Документы/yurp/yurpii$ ./a.out -o cos 1
Результат :)
0.9998
elena@elena-GF63-Thin-9RCX:~/Документы/yurp/yurpii$ ./a.out -o c 1
Ты чого тут понаписал, Ирод?:
c
Тригонометрический калькулятор :)

Выберете операцию:
1. Вычисление синуса: -o sin (число)
2. Вычисление косинуса: -o cos (число)
elena@elena-GF63-Thin-9RCX:~/Документы/yurp/yurpii$
```

```
52         std::cout << "Ты чого тут понаписал, Ирод?: " << operation << std::endl;
53         printUsage();
54         return 1;
55     }
56
57     std::cout << "Результат :) \n\n" << std::setprecision(4) << result << std::endl;
58
59     return 0;
60 }
```

Рисунок 4 – Результат выполнения программы

3.5 Был оформлен отчёт о выполненной работе, включая ссылку на свой репозиторий <https://github.com/funny-avocado/TxtIntr2>.

4 Вывод: был получен навык по обработке ввода данных из командной строки.

Листинг .gitignore

```
*.project
```


Приложение Б

(Обязательное)

Листинг Makefail

```
.PHONY: all debug release clean

RELEASE_FLAGS = -O2 -Wall -DNDEBUG
DEBUG_FLAGS   = -g -O0 -Wall
RELEASE_EXEC  = main
DEBUG_EXEC    = main-dbg
SOURCE= main.cpp

all: debug release

debug: $(DEBUG_EXEC)

$(DEBUG_EXEC): main.cpp
    g++ $(DEBUG_FLAGS) main.cpp -o $(DEBUG_EXEC)

release: $(RELEASE_EXEC)

$(RELEASE_EXEC): main.cpp
    g++ $(RELEASE_FLAGS) main.cpp -o $(RELEASE_EXEC)

clean:
    del -f $(RELEASE_EXEC).exe $(DEBUG_EXEC).exe
```


Приложение В

(Обязательное)

Листинг main.cpp

```
#include <iostream>
#include <iomanip>
#include <cmath>
#include <string>
#include <getopt.h>

void printUsage() {
    std::cout << "Тригонометрический калькулятор :) \n" << std::endl;
    std::cout << "Выберете операцию:" << std::endl;
    std::cout << "      1. Вычисление синуса: -o sin (число)" << std::endl;
    std::cout << "      2. Вычисление косинуса: -o cos (число)" << std::endl;
}

int main(int argc, char* argv[]) {
    std::string operation;
    double result = 0;

    struct option longOptions[] = {
        { "операция", required_argument, nullptr, 'o' },
        { nullptr, 0, nullptr, 0 }
    };

    int optionIndex;
    int option;
    while ((option = getopt_long(argc, argv, "o:", longOptions,
    &optionIndex)) != -1) {
        switch (option) {
            case 'o':
                operation = optarg;
                break;
            default:
                printUsage();
                return 1;
        }
    }
}
```

```

    if (argc - optind < 1 || operation.empty()) {
        printUsage();
        return 1;
    }
    for (int i = optind; i < argc; i++) {
        double operand = std::stod(argv[i]);
        if (operation == "sin") {
            result += sin(operand * M_PI / 180.0);
        }
        else if (operation == "cos") {
            result += cos(operand * M_PI / 180.0);
        }
        else {
            std::cout << "Ты чаго тут понаписал, Ирод?: \n " << operation <<
std::endl;

            printUsage();
            return 1;
        }
    }

    std::cout << "Результат :) \n\n" << std::setprecision(4) << result <<
std::endl;
    return 0;
}

```