Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики» (СибГУТИ)

Кафедра ПМиК

**Лабораторная работа № 9**

**по дисциплине «Современные технологии программирования»**

**«Редактор простых дробей»**

Выполнил: студент 4 курса ф. ИВТ, гр. ИП-813

Пещеров Вячеслав Александрович

Проверил: ассистент

к. ПМиК Агалаков А.А.

Новосибирск, 2021

Оглавление

[**Задание** 3](#_Toc90048879)

[**Реализация** 4](#_Toc90048880)

[**Демонстрация работы** 5](#_Toc90048881)

[**Вывод** 10](#_Toc90048882)

[**Список литературы** 11](#_Toc90048883)

[**Приложение** 12](#_Toc90048884)

[Листинг 1. TEditor.h 12](#_Toc90048885)

[Листинг 2. TEditor.cpp 13](#_Toc90048886)

[Листинг 3. Main.cpp 15](#_Toc90048887)

# **Задание**

1. Разработать и реализовать класс TEditor «Ввод и редактирование простых

дробей», используя класс С++.

2. Протестировать каждую операцию, определенную на типе данных,

используя средства модульного тестирования по критерию С1.

3. Если необходимо, предусмотрите возбуждение исключительных

ситуаций.

# **Реализация**

В данной работе мы создали редактор простых дробей и все необходимые для работы с ним операции, такие как: добавить 0, написать число, изменить знак, удалить символ, очистить, вывод числа, ввести число в строку, сменить режим.

# **Демонстрация работы**

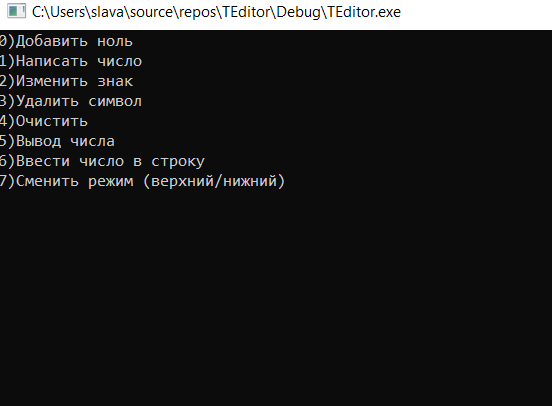


Рисунок 1. Меню

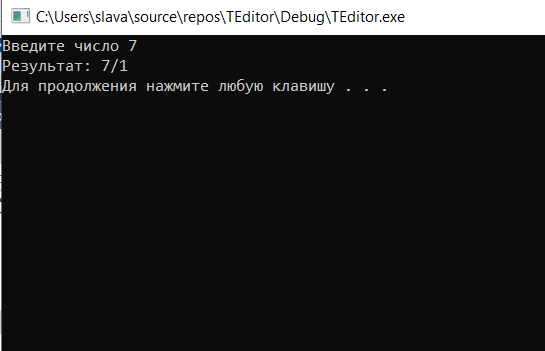


Рисунок 2. Ввод числа

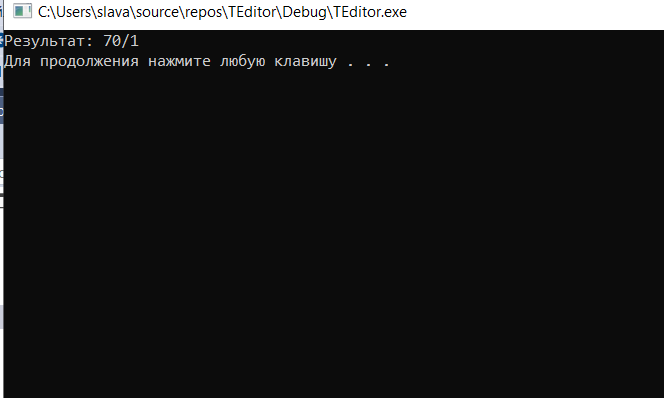


Рисунок 3. Добавить ноль

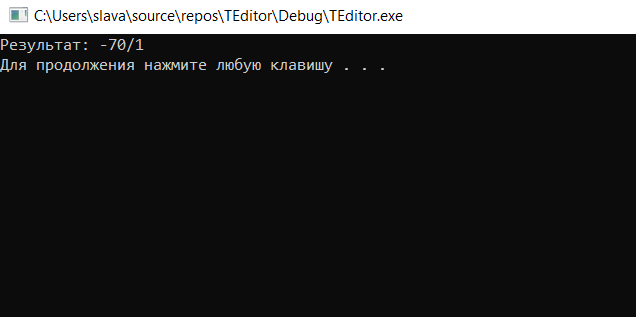


Рисунок 4. Изменить знак числа

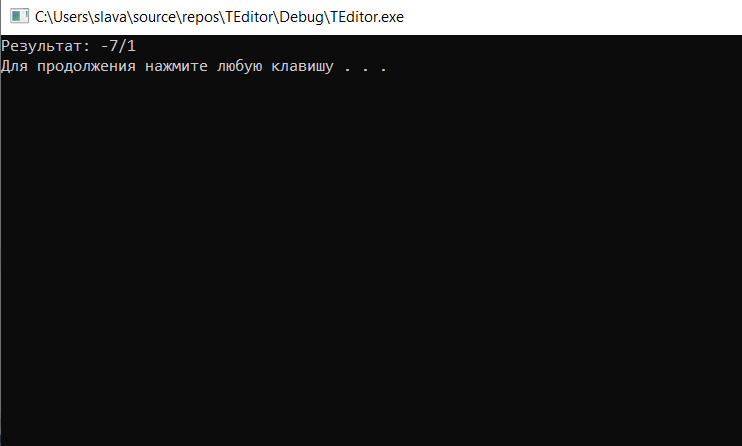


Рисунок 5. Удалить символ

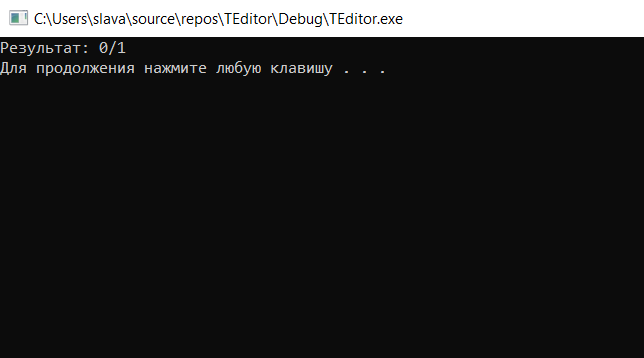


Рисунок 6. Очистить число

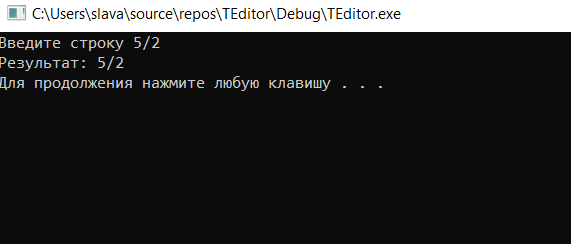


Рисунок 7. Ввод числа в строк

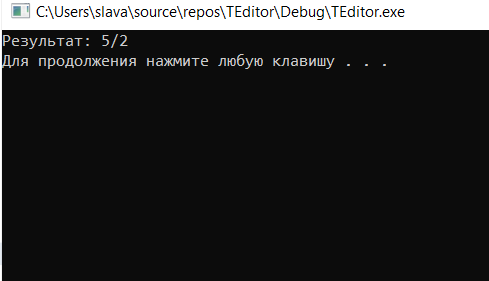


Рисунок 8. Вывод числа

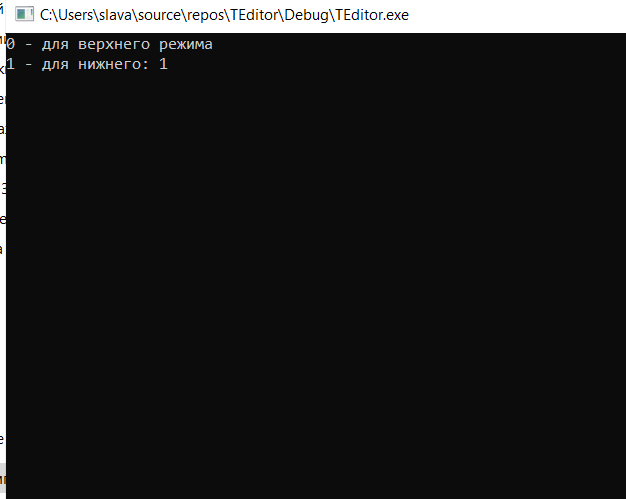


Рисунок 9. Изменение режима ввода

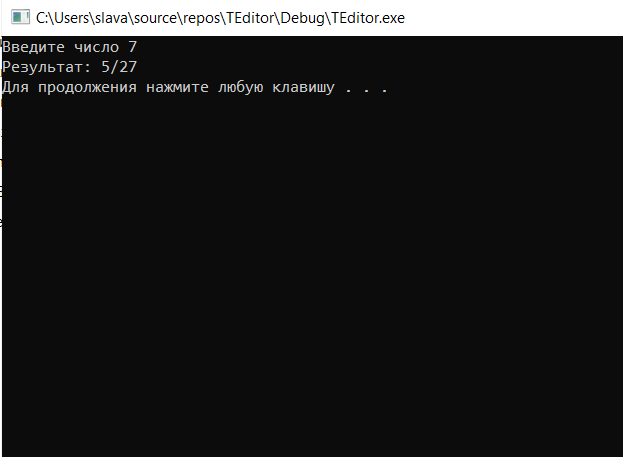


Рисунок 10. Добавление числа в знаменатель



Рисунок 11. Тесты

# **Вывод**

Мы научились работать в среде Visual Studio, а именно разрабатывать в ней модульные тесты для тестирования наших функции и классов на языке C#. Данная среда отлично подходит для выполнения модульного тестирования и автоматизации.

# **Список литературы**

1. Подбельский В.В., Фомин С.С.инт Курс программирования на языке Си: учебник. – М.:ДМК Пресс, 2012 – 384 с.
2. Павловская Т.А. C#. Программирование на языке высокого уровня: Учебник для вузов. - СПб. : Питер, 2014 - 432 с. : ил. - (Серия "Учебник для вузов").
3. Рихтер Дж. CLR via C#. Программирование на платформе Microsoft.NET Framework 4 на языке C# . 3-е изд.: - СПб.:Питер, 2012 - 928 с. : ил.
4. Котляров, В. П. Основы тестирования программного обеспечения : учебное пособие для СПО / В. П. Котляров. — Саратов : Профобразование, 2019 — 335 c. — ISBN 978-5-4488-0364-2. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: http://www.iprbookshop.ru/86202.html (дата обращения: 21.08.2020). — Режим доступа: для авторизир. пользователей

# **Приложение**

## Листинг 1. TEditor.h

#pragma once

#include <iostream>

#include <string>

using namespace std;

enum partToEdit { TOP, BOTTOM };

enum commandType { ADD\_ZERO, ADD\_NUMBER, ADD\_SIGN, DEL\_CHAR, CLEAR, READ, WRITE, MODE };

class TEditorclass

{

private:

const string strSeparator = "/";

const string strZero = "0/1";

partToEdit mod;

string strf;

public:

TEditorclass();

~TEditorclass();

bool isFractionZero();

string addSign();

string addNumber(int a);

string addZero();

string deleteCharacter();

string clear();

string readString();

void writeString(string a);

string edit(commandType a);

void setMode(partToEdit a);

};

## Листинг 2. TEditor.cpp

// TEditor.cpp : Этот файл содержит функцию "main". Здесь начинается и заканчивается выполнение программы.

//

#include <iostream>

#include "TEditorclass.h"

using namespace std;

int main()

{

setlocale(LC\_ALL, "RUS");

TEditorclass t;

int c;

while (1) {

cout << "0)Добавить ноль" << endl;

cout << "1)Написать число" << endl;

cout << "2)Изменить знак " << endl;

cout << "3)Удалить символ" << endl;

cout << "4)Очистить" << endl;

cout << "5)Вывод числа" << endl;

cout << "6)Ввести число в строку" << endl;

cout << "7)Сменить режим (верхний/нижний)" << endl;

cin >> c;

if (c == 8) break;

system("cls");

cout << "Результат: " << t.edit((commandType)c) << endl;

system("pause");

system("cls");

}

return 0;

}

## Листинг 3. Main.cpp

#include "TEditorclass.h"

#include <regex>

TEditorclass::TEditorclass() {//конструктор

mod = TOP;

strf = strZero;

}

TEditorclass::~TEditorclass() {}

bool TEditorclass::isFractionZero() {//добавить 0

return strf == strZero || strf == ("-" + strZero);

}

string TEditorclass::addSign() {//добавить знак

if (strf[0] == '-')

strf = strf.substr(1, strf.size());

else strf = '-' + strf;

return strf;

}

string TEditorclass::addNumber(int a) {//добавить число

if (a < 0 || a > 10)

return strf;

int indexSep = strf.find(strSeparator);

if (mod == TOP) {

if (strf[0] == '0')

strf[0] = '0' + a;

else if (strf[1] == '0' && strf[0] == '-')

strf[1] = '0' + a;

else {

if (indexSep > 0)

strf.insert(indexSep, 1, ('0' + a));

else strf += '0' + a;

}

}

else {

if (strf[indexSep + 1] == '0')

strf.erase(indexSep + 1);

strf += '0' + a;

}

return strf;

}

string TEditorclass::addZero() {//добавить 0

return addNumber(0);

}

string TEditorclass::deleteCharacter() {//удалить символ

if (mod == TOP) {

if (strf[0] == '0' || (strf[0] == '-' && strf[1] == '0'))

return strf;

else strf.erase(strf.find(strSeparator) - 1, 1);

if (strf[0] == '/' || (strf[0] == '-' && strf[1] == '/'))

strf.insert(strf.find(strSeparator), 1, '0');

}

else {

if (strf.find(strSeparator) + 1 != strf.size())

strf.pop\_back();

}

return strf;

}

string TEditorclass::clear() {//очистить

mod = TOP;

strf = strZero;

return strf;

}

string TEditorclass::readString() {//считать строку

return strf;

}

void TEditorclass::writeString(string a) {//ввести строку

regex regex("-?([0-9]+)\/[1-9][0-9]\*");

if (regex\_match(a, regex))

strf = a;

}

void TEditorclass::setMode(partToEdit a)\//установить нужный режим

{

mod = a;

}

string TEditorclass::edit(commandType a)//редактировать число

{

setlocale(LC\_ALL, "RUS");

string res, f;

switch (a) {

case ADD\_ZERO:

res = addZero();

break;

case ADD\_NUMBER:

int number;

cout << "Введите число ";

cin >> number;

res = addNumber(number);

break;

case ADD\_SIGN:

res = addSign();

break;

case DEL\_CHAR:

res = deleteCharacter();

break;

case CLEAR:

res = clear();

break;

case READ:

res = readString();

break;

case WRITE:

cout << "Введите строку ";

cin >> f;

writeString(f);

res = strf;

break;

case MODE:

cout << "0 - для верхнего режима \n1 - для нижнего: ";

int mod;

cin >> mod;

if (mod) {

setMode(BOTTOM);

}

else

{

setMode(TOP);

}

break;

default:

res = strf;

break;

}

return res;

}