

ЛАБОРАТОРНАЯ РАБОТА № 7
«ИССЛЕДОВАНИЕ СРЕДСТВ УПРАВЛЕНИЯ ПОТОКАМИ
ВВОДА-ВЫВОДА. ИССЛЕДОВАНИЕ МЕХАНИЗМА
ОБРАБОТКИ ИСКЛЮЧЕНИЙ»

Цель работы: Изучить способы реализации и особенности управления потоками ввода/вывода, исследовать способы генерации и обработки исключений.

Вариант задания

Вариант 11

1. Написать программу вычисления длины периметра и площади прямоугольника. Входные данные поступают с клавиатуры. Результат сохраняется в файле в формате: 15 позиций; точность 5 символов; заполняющий символ “#”. Предусмотреть обработку ошибок.

2. Найти значение математического выражения, описав перехват ошибок вычислений: $y = 12x + \sqrt{x - 8}$

2. Код программы на языке C++

```
#include <iostream>
#include <fstream>
using namespace std;

class ExAbsClass
{
protected:
    const std::string _message;
public:
    ExAbsClass(const std::string message) : _message(message) {}
    const std::string getMessage() const { return _message; }
};

class ZeroEx : public ExAbsClass { public: ZeroEx(std::string message) :
ExAbsClass(message) {} };
class MinusEx : public ExAbsClass { public: MinusEx(std::string message) :
ExAbsClass(message) {} };
class InFormEx : public ExAbsClass { public: InFormEx(std::string message) :
ExAbsClass(message) {} };
class FStringEmpty : public ExAbsClass { public: FStringEmpty(std::string message)
: ExAbsClass(message) {} };
class FileNotOpen : public ExAbsClass { public: FileNotOpen(std::string message) :
ExAbsClass(message) {} };
class InvalidLogArgumentException : public ExAbsClass { public:
InvalidLogArgumentException(std::string message) : ExAbsClass(message) {} };

void FirstZadanie();
void SecZadanie();

int main()
{
    system("chcp 1251");
    FirstZadanie();
    SecZadanie();
}

void FirstZadanie() {
    double a = 0, b = 0;
    try
    {
        cout << "Введите значение a= "; cin.precision(5); cin >> a;
        if (cin.rdstate() == 0x02) throw InFormEx("Периметр и площадь");
        cout << "Введите значение b= "; cin.precision(5); cin >> b;
        if (cin.rdstate() == 0x02) throw InFormEx("Периметр и площадь");
        if (a == 0 || b == 0) throw ZeroEx("Периметр и площадь");
        if (a < 0 || b < 0) throw MinusEx("Периметр и площадь");
        double per = a * 2 + b * 2;
        float floor = a * b;
        string s;
        cout << "Введите название файла, в который данные будут сохранены --> " << endl;
        cin >> s;
        if (s.empty()) throw FStringEmpty("Периметр и площадь");
        ofstream file(s);
        if (!file) throw FileNotOpen("Периметр и площадь");
        file << "Периметр = : "; file.fill('#'); file.width(15); file.precision(5); file
        << per << endl;
        file << "Площадь = : "; file.fill('#'); file.width(15); file.precision(5); file
        << floor << endl;
        file.close();
        cout << "Данные сохранены в файл : " << s << endl;
        cout << "Следующие значения : " << s << endl;
        cout << "Периметр = : "; cout.fill('#'); cout.width(15); cout.precision(5); cout
        << per << endl;
        cout << "Площадь = : "; cout.fill('#'); cout.width(15); cout.precision(5); cout
        << floor << endl;
    }
}
```

```

    }
    catch (InFormEx& e)
    {
        cerr << "ОШИБКА: неверный формат ввода данных." << endl << " * Получено в блоке: «" << e.getMessage() << "»." << endl;
    }
    catch (ZeroEx& e)
    {
        cerr << "ОШИБКА: одна из сторон равна 0." << endl << " * Получено в блоке: «" << e.getMessage() << "»." << endl;
    }
    catch (MinusEx& e)
    {
        cerr << "ОШИБКА: одна из сторон равна отрицательная." << endl << " * Получено в блоке: «" << e.getMessage() << "»." << endl;
    }
    catch (FileStringEmpty& e)
    {
        cerr << "ОШИБКА: строка имени файла пуста." << endl << " * Получено в блоке: «" << e.getMessage() << "»." << endl;
    }
    catch (FileNotOpen& e)
    {
        cerr << "ОШИБКА: не удалось открыть файл для сохранения." << endl << " * Получено в блоке: «" << e.getMessage() << "»." << endl;
    }
    catch (...)
    {
        cerr << "Поймано необрабатываемое исключение" << endl;
    }
}

```

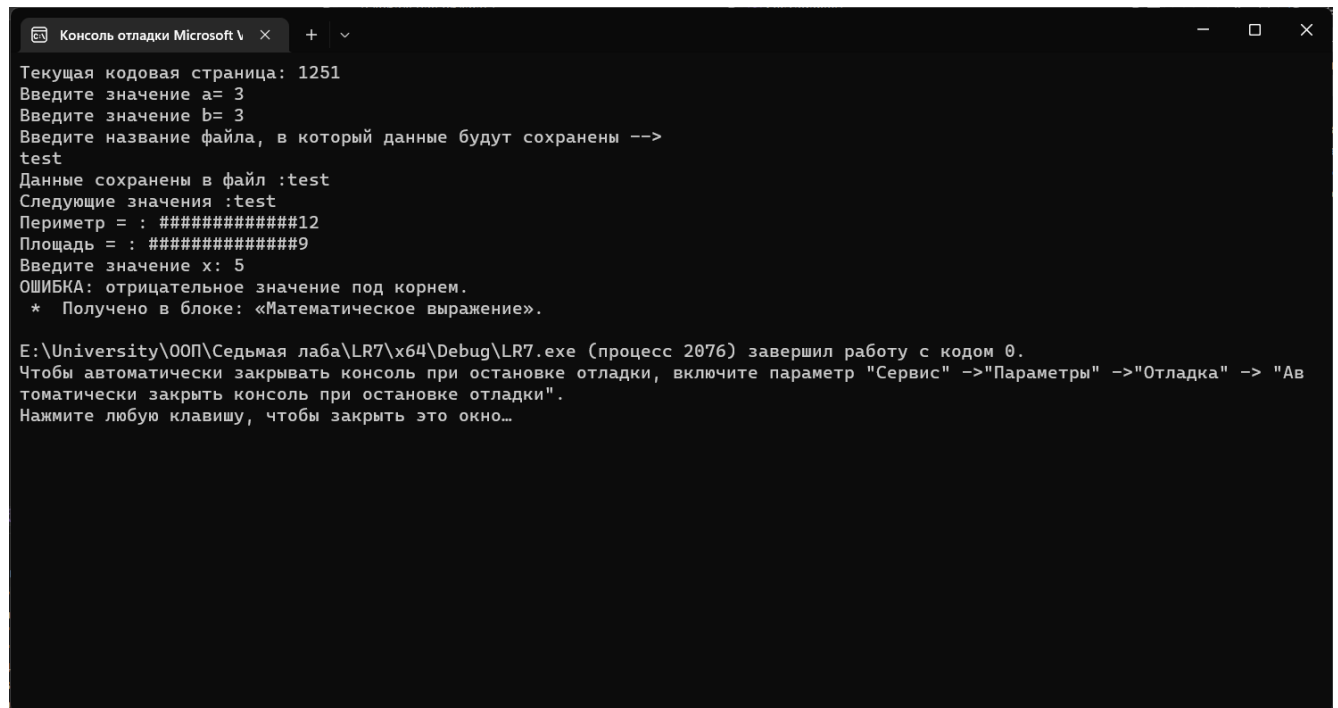
```

void SecZadanie() {
    try
    {
        double x, y;
        cout << "Введите значение x: "; cin >> x;
        if (cin.rdstate() == 0x02) throw InFormEx("Математическое выражение");
        double temp = x - 8; if (temp < 0) throw MinusEx("Математическое выражение");
        y = 12 * x + sqrt(temp);
        cout << "Полученное значение: y= " << y;
    }
    catch (InFormEx& e)
    {
        cerr << "ОШИБКА: неверный формат ввода данных." << endl << " * Получено в блоке: «" << e.getMessage() << "»." << endl;
    }
    catch (MinusEx& e)
    {
        cerr << "ОШИБКА: отрицательное значение под корнем." << endl << " * Получено в блоке: «" << e.getMessage() << "»." << endl;
    }
    catch (...)
    {
        cerr << "Поймано необрабатываемое исключение" << endl;
    }
}

```

3. Тестирование и отладка

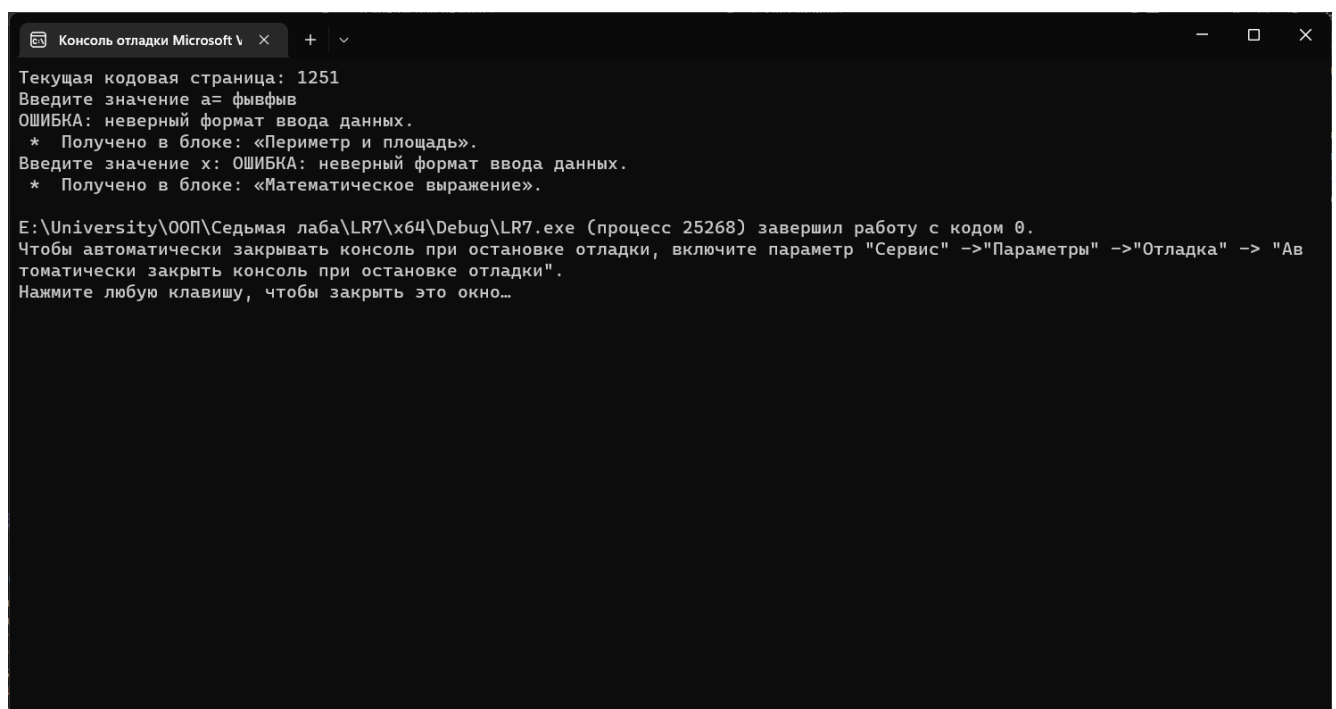
Тестирование данной программы происходит в два этапа. Сначала происходит тестирование части отвечающей за расчет площади и периметра прямоугольника и сохранения этих данных в файл, а после происходит тестирование части, связанной с расчетом значений функции.



```
Консоль отладки Microsoft Visual Studio
Текущая кодовая страница: 1251
Введите значение a= 3
Введите значение b= 3
Введите название файла, в который данные будут сохранены -->
test
Данные сохранены в файл :test
Следующие значения :test
Периметр = : #####12
Площадь = : #####9
Введите значение x: 5
ОШИБКА: отрицательное значение под корнем.
* Получено в блоке: «Математическое выражение».

E:\University\ООП\Седьмая лаба\LR7\x64\Debug\LR7.exe (процесс 2076) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

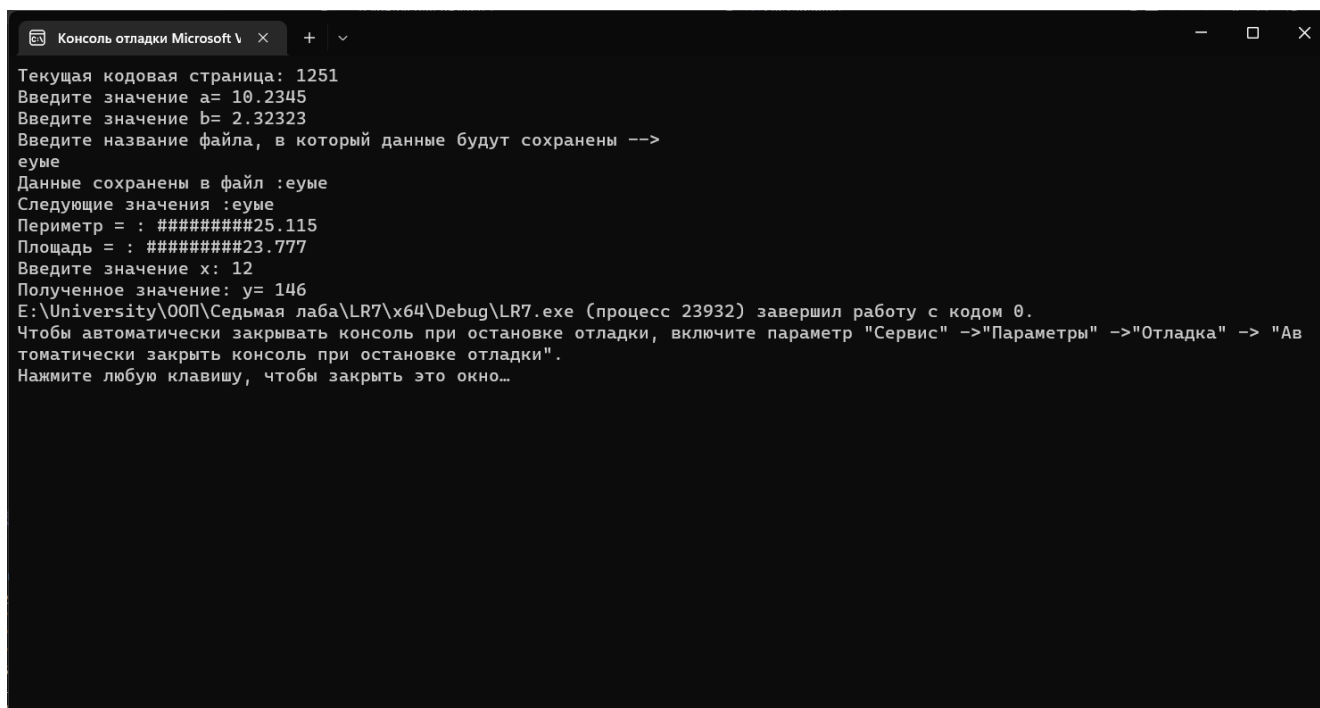
Рисунок 1 – Результат выполнения программы с неправильно введенным x.



```
Консоль отладки Microsoft Visual Studio
Текущая кодовая страница: 1251
Введите значение a= фывфыв
ОШИБКА: неверный формат ввода данных.
* Получено в блоке: «Периметр и площадь».
Введите значение x: ОШИБКА: неверный формат ввода данных.
* Получено в блоке: «Математическое выражение».

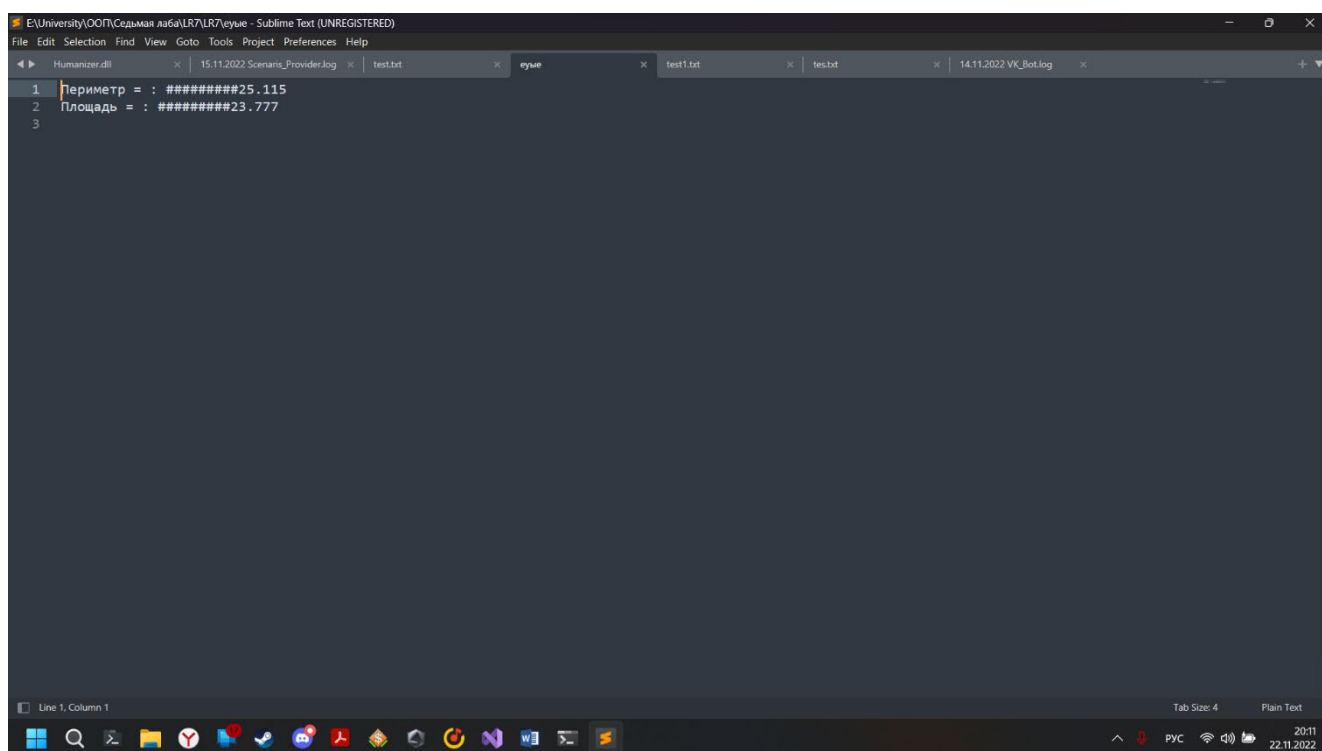
E:\University\ООП\Седьмая лаба\LR7\x64\Debug\LR7.exe (процесс 25268) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

Рисунок 2 – Проверка работоспособности отлова ошибок неправильного типа ввода



```
Консоль отладки Microsoft Visual Studio
Текущая кодовая страница: 1251
Введите значение a= 10.2345
Введите значение b= 2.32323
Введите название файла, в который данные будут сохранены -->
еуые
Данные сохранены в файл :еуые
Следующие значения :еуые
Периметр = : #####25.115
Площадь = : #####23.777
Введите значение x: 12
Полученное значение: y= 146
E:\University\ООП\Седьмая лаба\LR7\х64\Debug\LR7.exe (процесс 23932) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

Рисунок 3 – Результат работы программы при корректном вводе данных.



```
E:\University\ООП\Седьмая лаба\LR7\еуые - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Humanizer.dll x 15.11.2022 Scenario_Provider.log x test.txt x еуые x test1.txt x test.txt x 14.11.2022 VK_bot.log x
1 Периметр = : #####25.115
2 Площадь = : #####23.777
3
```

Рисунок 4 – Данные, сохраненные в файле

В результате тестирования программы, форматирование вывода и отлов ошибок работают корректно.

Вывод

При выполнении данной лабораторной работы были получены навыки работы с механизмом создания и обработки исключений. Были выявлены положительные стороны использования механизма исключений.