Министерство образования и науки Кыргызской Республики

Кыргызский государственный технический университет

им. И.Раззакова

Факультет информационных технологий

Кафедра «Программное обеспечение компьютерных систем»

Направление: 710400 «Программная инженерия»

Дисциплина: «Объектно – ориентированное программирование»

**ОТЧЕТ**

По лабораторной работе №9.

Тема: «Исключение»

                                                                       Выполнила: студент группы

ПИ(б)-2-21 Марат к. Жылдыз

Проверил: Мусабаев Э.Б.

**Бишкек – 2024**

**Лабораторная работа №9.**

**Тема: Исключение.**

**Задания:**

**1.**Добавьте класс исключений к программе ARROVER1, чтобы индексы, выходящие за пределы массива, вызывали генерацию исключения. Блок-ловушка может выводить пользователю сообщение об ошибке.

Данные помещаются в массив с помощью метода putel () и выводятся на экран с помощью метода getel (). Безопасность массива реализована с помощью вывода сообщений об ошибке при попытке использования индекса, не входящего в границы массива.

**Код программы:**

#include <iostream>

#include<process.h>

using namespace std;

const int LIMIT = 20;

class safearray {

private:

int arr[LIMIT];

public:

class Error {

};

void putel(int n, int elvalue)

{

if (n < 0 || n >= LIMIT) {

throw Error();

}

arr[n] = elvalue;

}

int getEl(int n) const

{

if (n < 0 || n >= LIMIT) {

throw Error();

}

return arr[n];

}

};

int main() {

system("chcp 1251");

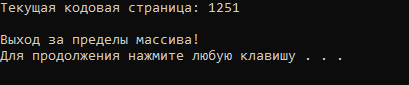
cout << "\nВыход за пределы массива!" << endl;

system("pause");

return 0;

}

**Результат программы:**



1. Задание 3 из лабораторной работы №4 выполнить с использованием механизма исключений.

Код программы:

#include<iostream>

#include<Windows.h>

using namespace std;

class rectangle {

private:

float a;

float b;

public:

rectangle() :a(0.0), b(0.0) {} // Конструктор по умолчанию

~rectangle() { cout << endl << "Деструктор завершен" << endl; } // Деструктор

// Перегружаемые функции

class Error {

};

float rect\_area(float a, float b) { // функция для прямоуголника

if (a <= 0 || b <= 0) {

throw Error();

}

return a \* b;

}

float rect\_area(float a) { // функция для квадрата

if (a < 0 || b < 0) {

throw Error();

}

return pow(a, 2);

}

};

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setlocale(LC\_ALL, "RUS");

float st1 = 0, st2 = 0;

int val = 0;

rectangle obj;

try

{

while (val != 3 && val < 3) // Пока не равно или пока меньше трех

{

cout << "\tМеню" << endl;

cout << "1) Прямоугольник " << "\n2) Квадрат " << "\n3) Выход" << endl;

cin >> val;

if (val == 1) {

cout << "Введите сторону a: ";

cin >> st1;

cout << "Введите сторону b: ";

cin >> st2;

cout << endl << "Площадь прямоуголника равна: " << obj.rect\_area(st1, st2) << endl; // Вызов и вывод на экран результата перегружаемой функции

}

else if (val == 2) {

cout << "Введите сторону a: ";

cin >> st1;

cout << endl << "Площадь квадрата равна: " << obj.rect\_area(st1) << endl; // Вызов и вывод на экран результата перегружаемой функции

}

}

}

catch (rectangle::Error)

{

cout << "Числа не могут быть отрицательными!" << endl;

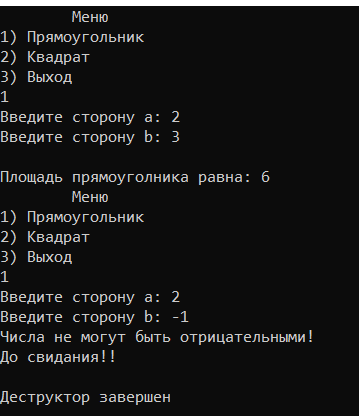
}

cout << "До свидания!!" << endl;

}

**Результат программы:**

**Рис.1. Для прямоугольника:**



**Рис.2. Для квадрата.**

