Министерство науки и высшего образования Российской Федерации

Чайковский филиал

федерального государственного автономного образовательного учреждения

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Кафедра Автоматизации, информационных и инженерных технологий

Направление: 09.03.01 Информатика и вычислительная техника

Направленность: Автоматизированные системы обработки информации и управления

**О Т Ч Е Т**

**по лабораторной работе №5**

«Классы и объекты»

по дисциплине

«Теория алгоритмов и структуры данных»

|  |  |  |
| --- | --- | --- |
|  |  | Выполнили: |
|  |  | студенты группы АСУ-22-1б ЧФ |
|  |  | *Базуев Д.О., Гимадеев Р.Р.* |
|  |  | *(Фамилия Имя Отчество)* |
|  |  |  |
|  |  | Проверил: |
|  |  | *Ст. преподаватель* |
|  |  | *(должность, ученая степень, ученое звание)* |
|  |  | *Сухих И.И.* |
|  |  | *(Фамилия Имя Отчество)* |
|  |  | *(дата, подпись)* |

Чайковский 2023

Цель работы: Получение практических навыков при работе с классами и объектами.

Программное обеспечение: Microsoft Visual Studio 2022.

Задание 1. Реализовать (в отдельном файле) определение нового класса (закрытые атрибуты, свойства, конструкторы, инициализация и вывод атрибутов).Для демонстрации работы с объектами написать главную функцию, в которой создаются объекты класса и выводится информация, которая содержится в атрибутах. Написать функцию, реализующую указанное в варианте действие. Рассмотреть два варианта:1) статическую функцию; 2) метод класса. Используя статическую компоненту класса подсчитать количество созданных в программе объектов.

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | Diapason | Double x, double y (начало и конец диапазона) | Определить попадает ли заданное число в диапазон. Результат должен быть типа double. |

Исходный код Diapason:

#include<iostream>

#include<windows.h>

class Diapason {

static int counter;

public:

// 1 задание

Diapason() {

counter++; // увеличиваем счетчик при каждом создании объекта

}

~Diapason() {

counter--; // уменьшаем счетчик при каждом удалении объекта

}

Static int getCounter() {

return counter; // метод для получения текущего значения счетчика

}

double x;

double y;

double z;

double intersection() {

return ((z >= x) && (z <= y));

}

};

int Diapason::counter = 0;

Исходный код основной функции:

#include <iostream>

#include "D:\Информатика\ТАиСД\Лабораторные за 1 семестр 2 курса\Diapason\Diapason.cpp"

using namespace std;

double inter(double x, double y, double z);

double InputValidation(double num, string s);

int main()

{

setlocale(0, "");

Diapason D;

double z;

D.x = InputValidation(D.x, "Введите начало диапазона: ");

D.y = InputValidation(D.y, "Введите конец диапазона: ");

cout << "Введите число: ";

cin >> z;

cout << "Ваш диапазон: [" << D.x << " ; " << D.y << "]" << endl;

cout << "\n\nРезультаты проверки на вхождение в диапазон через метод: " << D.intersection(D, z)

<< "\nРезультаты проверки на вхождение в диапазон через функцию: " << inter(D.x, D.y, z)

<< "\n\nВсего было создано " << Diapason::getCounter() << " объектов класса\n";

}

double inter(double x, double y, double z)

{

return ((z >= x) && (z <= y));

}

double InputValidation(double num, string s)

{

bool b = false;

do {

b = false;

cout << s;

if (!(cin >> num)) {

cin.clear();

cin.ignore((numeric\_limits<streamsize>::max)(), '\n');

cout << "Ошибка ввода!\n";

b = true;

}

} while (b);

return num;

}

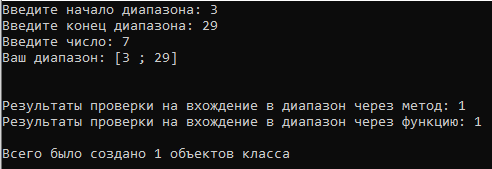


Рисунок 1 – Результат работы программы с учётом вхождения числа

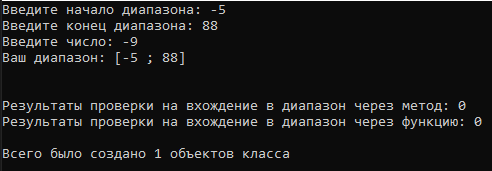


Рисунок 2 – Результат работы программы без вхождения числа

Задание 2. Добавить к реализованному классу указанные в варианте перегруженные операции. Написать демонстрационную программу, в которой создаются объекты пользовательских классов и выполняются указанные операции.

|  |  |  |
| --- | --- | --- |
| 5 | Diapason | Унарные операции:  ! - вычислить длину диапазона, результат должен быть типа double;  ++ увеличить координаты диапазона на 1.  Операции приведения типа:  int (явная) – результатом является целая часть координаты х;  double (неявная) – результатом является координата y.  Бинарные операции:  + int d, целое число (лево- и право-сторонняя операция, координаты увеличиваются на число d);  <Diapason d, целое число – результат равен true, если целое число попадает в заданный диапазон и false – в противном случае. |

Исходный код Diapason:

#include <iostream>

#include <windows.h>

class Diapason {

static int counter;

public:

// 1 задание

Diapason() {

counter++; // увеличиваем счетчик при каждом создании объекта

}

~Diapason() {

counter--; // уменьшаем счетчик при каждом удалении объекта

}

static int getCounter() {

return counter; // метод для получения текущего значения счетчика

}

double x;

double y;

bool b = false;

double intersection(Diapason D, double z) {

if ((z >= D.x) && (z <= D.y)) b = true;

return ((z >= D.x) && (z <= D.y));

}

// 2 задание

double size() {

double Max = max(x, y);

double Min = min(x, y);

if (Max < 0) {

Max = abs(Max);

Min = abs(Min);

std::swap(Max, Min);

}

return Max - Min;

}

int Convert(double x) {

return (int)x; // неявное приведение типа double к int (целая часть)

}

double Convert(int y) {

return (double)y; // явное приведение типа int к double

}

double binar\_operation(int num, double x) {

x += num;

return x;

}

double unar\_operation(double y) {

y++;

return y;

}

bool binar\_operation(Diapason D, double z) {

intersection(D, z);

return b;

}

};

int Diapason::counter = 0;

Исходный код основной функции:

#include <iostream>

#include <windows.h>

#include "E:\Информатика\ТАиСД\Лабораторные за 1 семестр 2 курса\Diapason\Diapason.cpp"

using namespace std;

double InputValidation(double num, string s);

int InputValidation(int num, string s);

int main() {

setlocale(0, "");

Diapason D, d;

double z = 0;

int num = 0;

D.x = InputValidation(D.x, "Введите начало диапазона: ");

D.y = InputValidation(D.y, "Введите конец диапазона: ");

z = InputValidation(z, "Введите число для проверки на вхождение: ");

num = InputValidation(num, "Введите на сколько нужно увеличить координаты: ");

int increase, decrease;

cout << "\nВаш диапазон:"

<< "\n" << D.x << " ... " << D.y

<< "\n\nУнарные операции:\nДлина диапазона: " << D.size()

<< "\nОперация по увелечению координат на 1: "

<< "\nКоордината X: " << D.unar\_operation(D.x)

<< "\nКоордината Y: " << D.unar\_operation(D.y)

<< "\n\nОперации приведения типа:\nЦелая часть от X: " << D.Convert(D.x)

<< "\nКоордината Y: " << D.Convert(D.y)

<< "\n\nБинарные операции:\n1.Операция по увелечению координат на заданное число:"

<< "\nКоординаты диапазона после увеличения: "

<< "\nКоордината X: " << D.binar\_operation(num, D.x)

<< "\nКоордината Y: " << D.binar\_operation(num, D.y)

<< "\n\n2.Операция по проверке на вхождение числа:"

<< "\nРезультат: " << D.binar\_operation(D, z) << '\n';

}

double InputValidation(double num, string s) {

bool b = false;

do {

b = false;

cout << s;

if (!(cin >> num)) {

cin.clear();

cin.ignore((numeric\_limits<streamsize>::max)(), '\n');

cout << "Ошибка ввода!\n";

b = true;

}

} while (b);

return num;

}

int InputValidation(int num, string s) {

bool b = false;

do {

b = false;

cout << s;

if (!(cin >> num)) {

cin.clear();

cin.ignore((numeric\_limits<streamsize>::max)(), '\n');

cout << "Ошибка ввода!\n";

b = true;

}

} while (b);

return num;

}

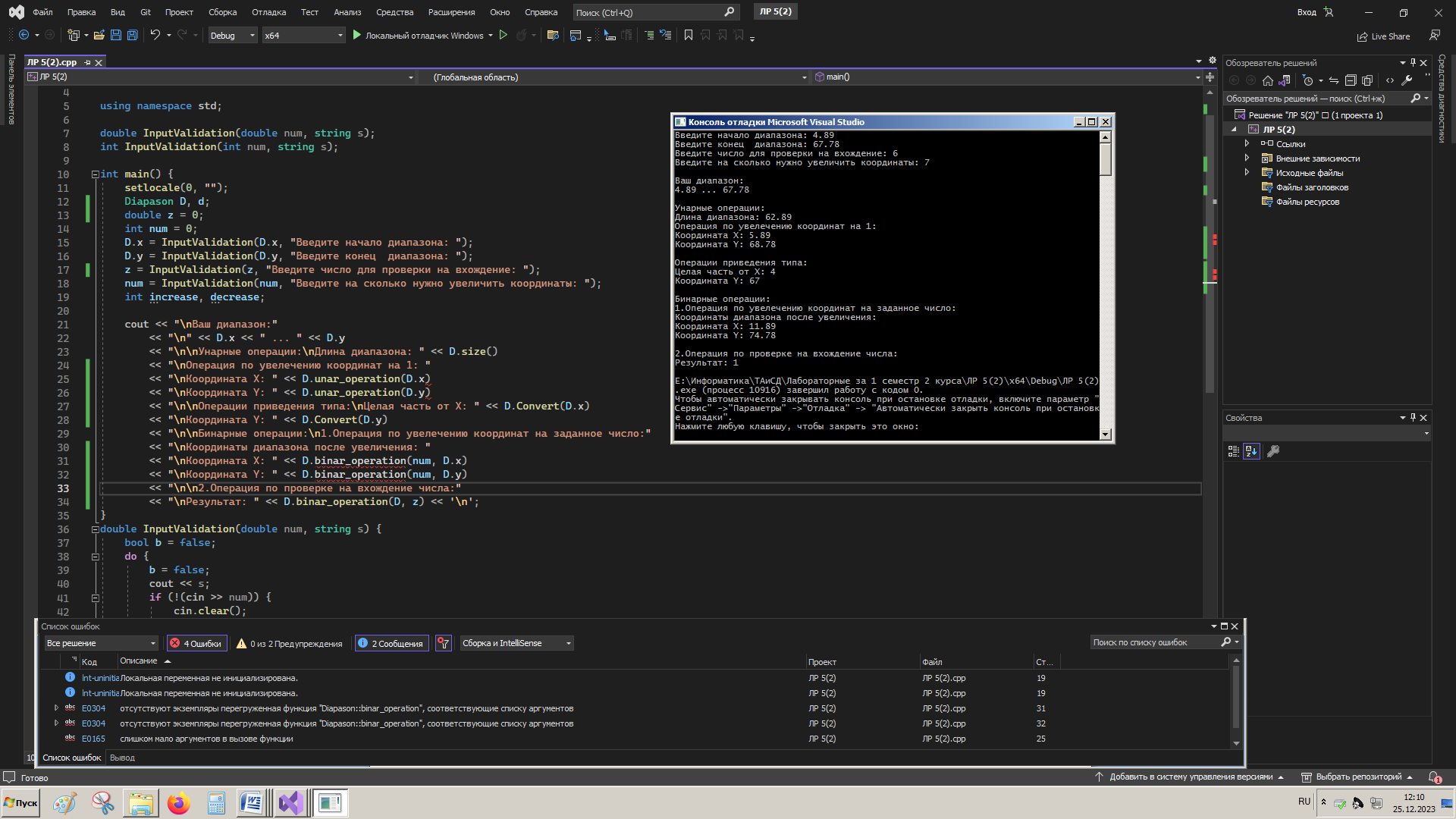


Рисунок 3 – Результат работы программы с учётом вхождения числа

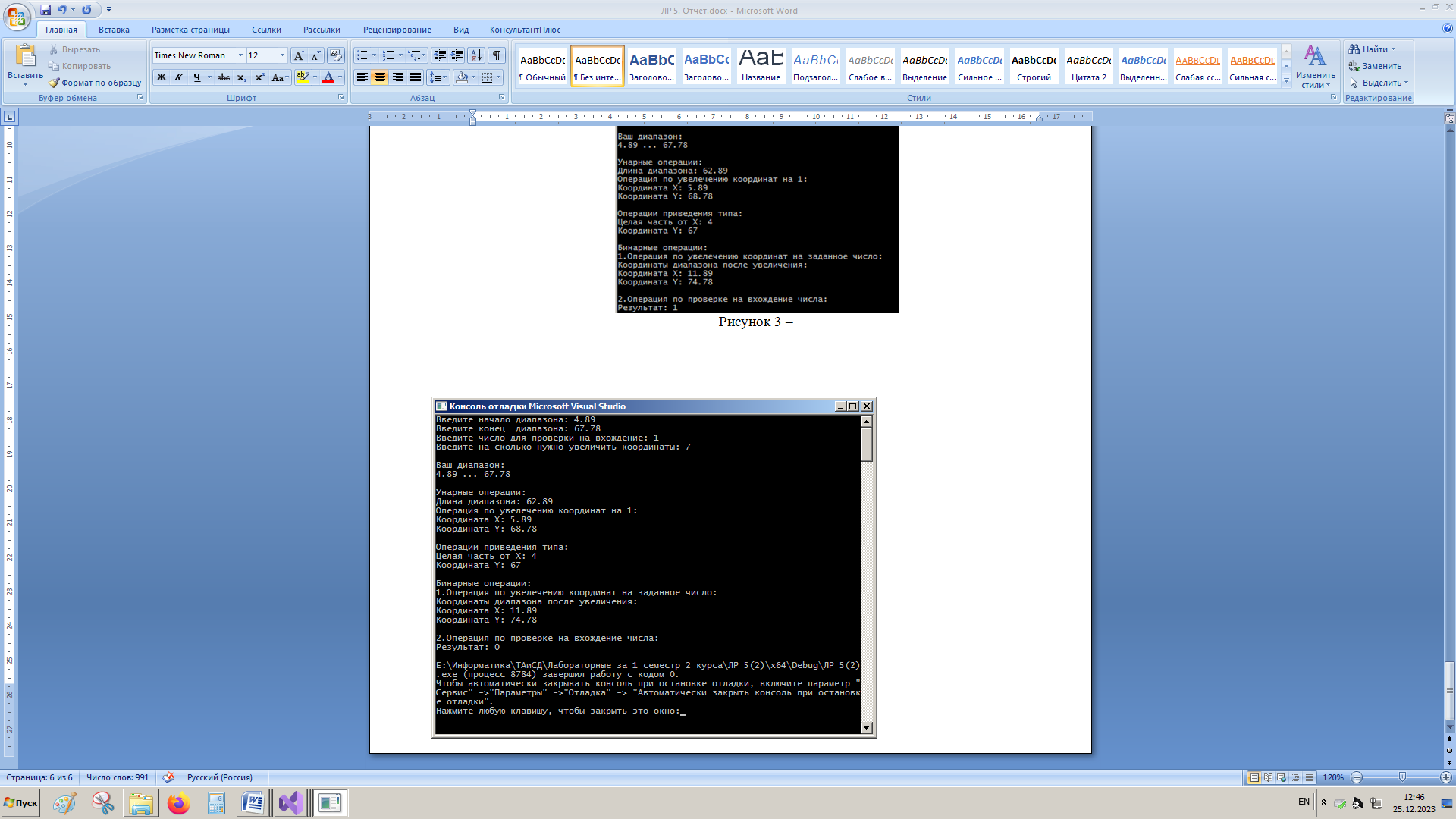


Рисунок 4 – Результат работы программы без вхождения числа

Задание 3.

1. Реализовать класс (в отдельном файле), полем которого является одномерный массив из элементов заданного в варианте типа. В классе реализовать:

* конструктор без параметров;
* конструктор с параметрами, заполняющий элементы случайными значениями;
* конструктор с параметрами, позволяющий заполнить массив элементами, заданными пользователем с клавиатуры;
* индексатор (для доступа к элементам массива);
* метод для просмотра элементов.

1. Написать демонстрационную программу, позволяющую создать массив разными способами и распечатать элементы массива. Подсчитать количество созданных объектов.
2. Выполнить указанное в варианте задание (если необходимо, перегрузить нужные для выполнения задачи операции или функции).

|  |  |  |
| --- | --- | --- |
| 5 | Diapason | Максимальное значение. |

Исходный код DiapasonArray:

#include <iostream>

#include <windows.h>

#include <time.h>

#include "E:\Информатика\ТАиСД\Лабораторные за 1 семестр 2 курса\Diapason\Diapason.cpp"

using namespace std;

class DiapasonArray {

static int count;

double Max = DBL\_MIN;

int MaxEl;

public:

int SizeArr;

Diapason\* arr;

DiapasonArray() {

count++;

}

DiapasonArray(int size) {

count++;

SizeArr = size;

arr = new Diapason[SizeArr];

for (int i = 0; i < size; i++)

{

cout << "Введите начало диапазона " << i + 1 << ": ";

cin >> arr[i].x;

cout << "Введите конец диапазона " << i + 1 << ": ";

cin >> arr[i].y;

}

}

DiapasonArray(int size, int min, int max) {

count++;

SizeArr = size;

arr = new Diapason[SizeArr];

for (int i = 0; i < size; i++)

{

arr[i].x = rand() % (max - min) + min;

arr[i].y = rand() % (max - min) + min;

}

}

~DiapasonArray() {

count--;

}

Diapason Max\_Element() {

for (int i = 0; i < SizeArr; i++)

{

if (Max < arr[i].x) {

Max = arr[i].x;

MaxEl = i;

}

else

{

if (Max < arr[i].y) {

Max = arr[i].y;

MaxEl = i;

}

}

}

return arr[MaxEl];

}

double MaxNum() {

return max(Max\_Element().x, Max\_Element().y);

}

Diapason NumOut(int i) {

return arr[i - 1];

}

void ArrayOut() {

for (int i = 0; i < SizeArr; i++)

{

cout << arr[i].x << "\t...\t" << arr[i].y << "\n";

}

}

static int GetCount() {

return count;

}

};

int DiapasonArray::count = 0;

Исходный код основной функции:

#include <iostream>

#include "E:\Информатика\ТАиСД\Лабораторные за 1 семестр 2 курса\DiapasonArray\DiapasonArray.cpp"

using namespace std;

DiapasonArray Arr1(int size);

DiapasonArray ArrRand(int size, int min, int max);

int main()

{

srand(time(0));

setlocale(0, "");

Diapason D;

int index, size, choice, max, min;

DiapasonArray arr;

cout << "Введите размер массива: ";

cin >> size;

cout << "Выберите тип заполнения массива:\n"

"1. Вручную\n"

"2. Рандомными числами в выбраном диапазоне\n"

"Ваш выбор: ";

cin >> choice;

switch (choice)

{

case 1:

arr = Arr1(size);

break;

case 2:

cout << "Введите желаемый, минимально возможный элемент: ";

cin >> min;

cout << "Введите желаемый, максимально возможный элемент: ";

cin >> max;

arr = ArrRand(size, min, max);

break;

default:

return 0;

break;

}

cout << "\nВаш массив:\n";

arr.ArrayOut();

cout << "\nВведите к какой элемент вы хотите посмотреть: ";

cin >> index;

D = arr.NumOut(index);

cout << "Ваш элемент: " << D.x << " ... " << D.y;

cout << "\n\nМаксимальный элемент в массиве: " << arr.MaxNum();

cout << "\nСтрока массива с максимальным элементом: "

<< arr.Max\_Element().x << " ... " << arr.Max\_Element().y;

cout << "\n\nКоличество созданных объектов класса: "

<< DiapasonArray::GetCount() << "\n";

}

DiapasonArray Arr1(int size) {

DiapasonArray arr(size);

return arr;

}

DiapasonArray ArrRand(int size, int min, int max) {

DiapasonArray arr(size, min, max);

return arr;

}

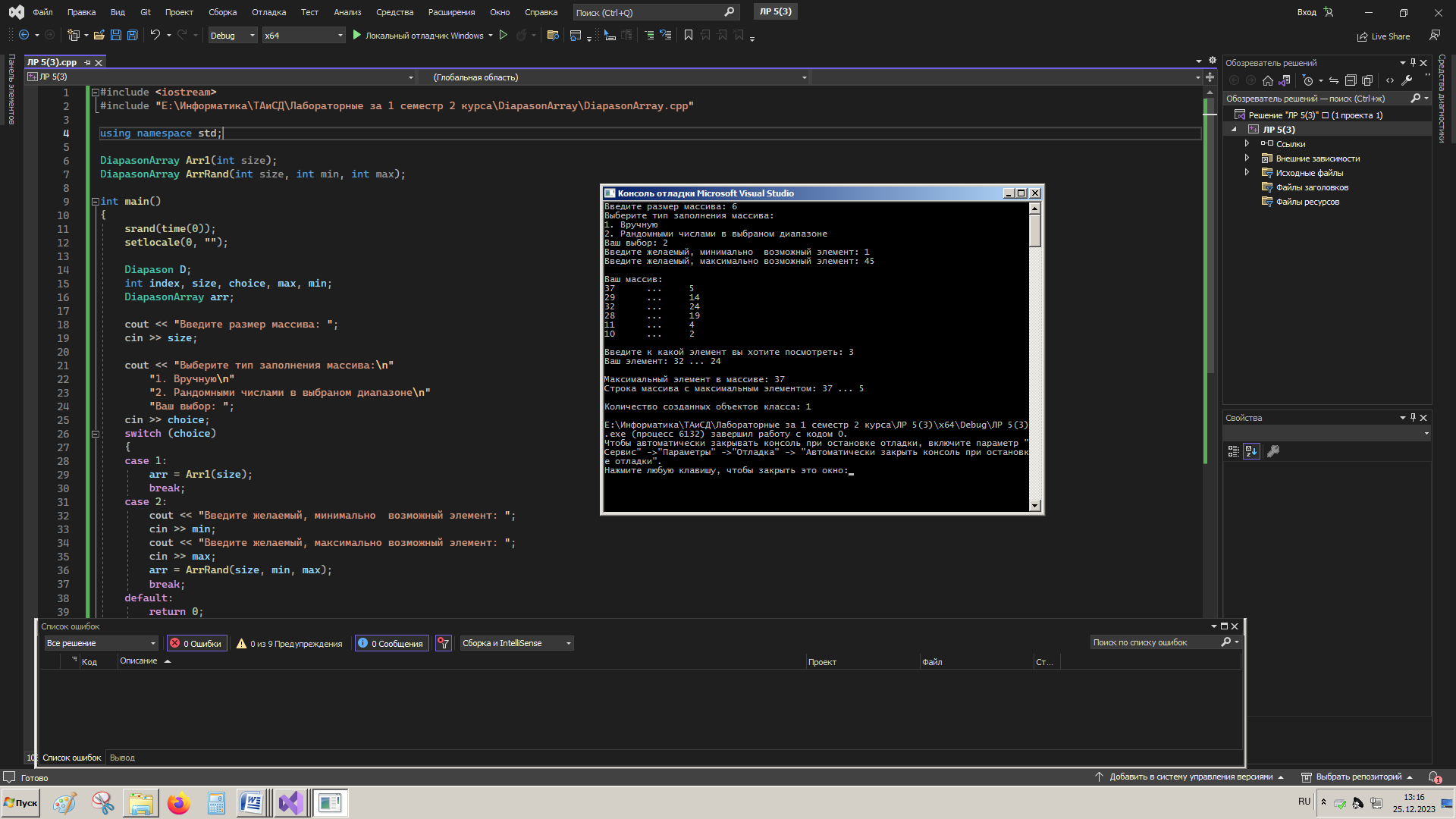


Рисунок 5 – Результат работы программы

Вывод: Были получены практические навыки при работе с классами и объектами.