|  |  |
| --- | --- |
| *voenmeh* | МИНОБРНАУКИ РОССИИ  федеральное государственное бюджетное образовательное учреждение  высшего образования  **«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»**  **(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова»)** |
| БГТУ.СМК-Ф-4.2-К5-02 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Факультет |  | О |  | Естественнонаучный |
|  |  | шифр |  | наименование |
| Кафедра |  | О7 |  | Информационные системы и программная инженерия |
|  |  | шифр |  | наименование |
| Дисциплина |  | Информационные технологии и программирование | | |

|  |
| --- |
| ПРАКТИЧЕСКАЯ РАБОТА №4 |
| ИЗУЧЕНИЕ ВЗАИМОДЕЙСТВИЯ УПРАВЛЯЕМОГО |
| И НЕУПРАВЛЯЕМОГО КОДА |
| Вариант №24 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент группы | | | |  | И924Б |
| Фокин В. К. | | | | | |
| Фамилия И.О. | | | | | |
| **Преподаватель:** | | | | | |
| Гладевич А. А. | |  |  | | |
| Фамилия И.О. Подпись | | | | | |
| Оценка |  | | | |  |
| «\_\_\_\_\_» |  | | | | 2023г. |

САНКТ-ПЕТЕРБУРГ

2023 г.

**Постановка задачи**

Работа состоит из двух частей: первая посвящена работе с неуправляемым кодом, вторая – графике.

**Первая часть**

Создать с использованием C или C++ динамическую библиотеку для работы с массивом, которая позволяет вводить массив, выводить массив и выполнять над ним действие согласно индивидуальному варианту. Далее необходимо подключить её к программе, написанной на C#, и продемонстрировать её возможности, соответствующим образом учтя разницу в используемых типах данных.

**Тексты программ**

Файл arraylib.h:

extern "C" \_\_declspec(dllexport) double\* fillArray();

extern "C" \_\_declspec(dllexport) void deleteArr(double\* arr);

extern "C" \_\_declspec(dllexport) void outputArray(double\* arr);

extern "C" \_\_declspec(dllexport) void sumAbs(double\* arr);

Файл arraylib.cpp:

#include "arrlib.h"

#include "pch.h"

#include <iostream>

#include <random>

#include <ctime>

using namespace std;

extern "C" \_\_declspec(dllexport) double\* fillArray() {

srand(static\_cast<double>(time(nullptr)));

double\* array = new double[50];

double min = -500.0, max = 500.0;

for (int i = 0; i < 50; i++) {

double randomValue = (static\_cast<double>(rand()) / RAND\_MAX) \* (max - min) + min;

array[i] = randomValue;

}

return array;

}

extern "C" \_\_declspec(dllexport) void deleteArr(double\* arr) {

delete[] arr;

}

extern "C" \_\_declspec(dllexport) void outputArray(double\* arr) {

for (int i = 0; i < 50; i++) {

cout << fixed << arr[i] << endl;

}

}

extern "C" \_\_declspec(dllexport) void sumAbs(double\* arr) {

double sumP = 0, sumN = 0, dif;

for (int i = 0; i < 50; i++) {

double num = arr[i];

if (num < 0) {

sumN += num;

}

else {

sumP += num;

}

}

cout << "sum of negative numbers: " << sumN << endl;

cout << "sum of positive numbers: " << sumP << endl;

dif = fabs(sumN) - sumP;

cout << "module difference: " << dif << endl;

}

Файл Program.cs:

using System;

using System.Runtime.InteropServices;

namespace task1

{

internal class Program

{

[DllImport("C:/c#/pr4/arrlib/x64/Debug/arrlib.dll", CallingConvention = CallingConvention.Cdecl)]

public static extern IntPtr fillArray();

[DllImport("C:/c#/pr4/arrlib/x64/Debug/arrlib.dll", CallingConvention = CallingConvention.Cdecl)]

public static extern void deleteArr(IntPtr arr);

[DllImport("C:/c#/pr4/arrlib/x64/Debug/arrlib.dll", CallingConvention = CallingConvention.Cdecl)]

public static extern void outputArray(IntPtr arr);

[DllImport("C:/c#/pr4/arrlib/x64/Debug/arrlib.dll", CallingConvention = CallingConvention.Cdecl)]

public static extern void sumAbs(IntPtr arr, int len);

static void Main(string[] args)

{

string menu;

IntPtr arr = IntPtr.Zero;

do

{

Console.WriteLine("----------------------------------------------------------------");

Console.WriteLine("| МЕНЮ |");

Console.WriteLine("----------------------------------------------------------------");

Console.WriteLine("| 1. Создать массив и заполнить 50 элементами |");

Console.WriteLine("| 2. Вывести массив |");

Console.WriteLine("| 3. Разность модулей сумм всех отрицательных |");

Console.WriteLine("| и всех положительных элементов вещественного массива |");

Console.WriteLine("| 0. Выход |");

Console.WriteLine("----------------------------------------------------------------");

menu = Console.ReadLine();

switch (menu)

{

case "1":

arr = fillArray();

Console.WriteLine("Массив создан");

break;

case "2":

if (arr != IntPtr.Zero)

{

outputArray(arr);

}

else

{

Console.WriteLine("Массив ещё не создан");

}

break;

case "3":

if (arr != IntPtr.Zero)

{

sumAbs(arr, 50);

}

else

{

Console.WriteLine("Массив ещё не создан");

}

break;

case "0":

deleteArr(arr);

break;

default:

Console.WriteLine("Пункта меню не существует");

break;

}

} while (menu != "0");

}

}

}

**Результат работы программы**

При запуске программы выводится главное меню. Меню показано на рисунке 1.

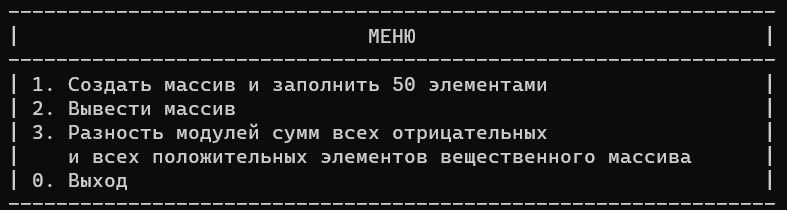


Рисунок 1 – Меню программы

При выборе пункта «1» можно создать вещественный массив, заполненный рандомно 50 элементами. Создание показано на рисунке 2.



Рисунок 2 – Создание массива

При выборе пункта «2» созданный массивы будет выведен на экран. Вывод показан на рисунке 3.

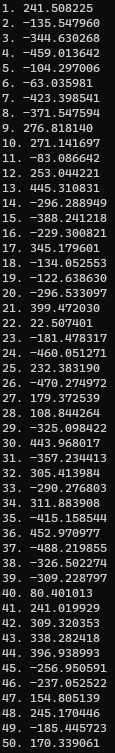


Рисунок 3 – Вывод массива

При выборе пункта «2», не создав массив до этого, выведется сообщение об ошибке. Ошибка показана на рисунке 4.



Рисунок 4 – Сообщение об ошибке

При выборе пункта «3» складываются отрицательные и положительные числа по отдельности, а затем из модуля суммы отрицательных чисел вычитается сумма положительных. Результат работы показан на рисунке 5.

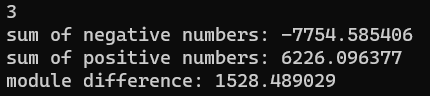


Рисунок 5 – Разность модулей

При выборе пункта «0» программа прекращает работу.

**Вторая часть. Задание**

Нарисовать кота или кошку (или капибару) программным способом. Использовать готовые изображения нельзя.

**Текст программы**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace task2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

Screen mainScreen = Screen.PrimaryScreen;

int screenWidth = mainScreen.Bounds.Width;

int screenHeight = mainScreen.Bounds.Height;

Size = new Size(screenWidth, screenHeight); // Устанавливает ширину и высоту формы

}

protected override void OnPaint(PaintEventArgs e)

{

base.OnPaint(e);

Screen mainScreen = Screen.PrimaryScreen;

int screenWidth = mainScreen.Bounds.Width;

int screenHeight = mainScreen.Bounds.Height;

Color brownBright = ColorTranslator.FromHtml("#A9775B");

Graphics graph = e.Graphics;

Pen brown = new Pen(Color.SaddleBrown, 3);

Pen penc = new Pen(Color.Black, 2);

SolidBrush brownBrush = new SolidBrush(brownBright);

SolidBrush blackBrush = new SolidBrush(Color.Black);

Point[] points = new Point[]

{

new Point(755, 96),

new Point(633, 87),

new Point(607, 92),

new Point(439, 184),

new Point(437, 200),

new Point(469, 347),

new Point(480, 361),

new Point(600, 363),

new Point(615, 455),

new Point(658, 583),

new Point(714, 660),

new Point(714, 800),

new Point(1395, 800),

new Point(1395, 695),

new Point(926, 225),

new Point(849, 224)

};

graph.FillEllipse(new SolidBrush(Color.SaddleBrown), 600, 55, 85, 85);

graph.FillPolygon(brownBrush, points);

graph.FillEllipse(new SolidBrush(Color.SaddleBrown), 710, 70, 85, 85);

graph.FillPie(brownBrush, 455, 225, 940, 940, 270, 90);

graph.FillPie(brownBrush, 605, 87, 46, 15, 185, 165);

graph.FillPie(brownBrush, 436, 182, 15, 30, 90, 180);

graph.FillPie(brownBrush, 469, 323, 69, 44, 145, 90);

graph.FillPie(brownBrush, 644, 320, 196, 347, 100, 50);

graph.DrawLine(brown, 439, 184, 607, 92);

graph.DrawLine(brown, 633, 87, 755, 96);

graph.DrawLine(brown, 755, 96, 849, 224);

graph.DrawLine(brown, 849, 224, 926, 225);

graph.DrawLine(brown, 437, 200, 469, 347);

graph.DrawLine(brown, 480, 361, 600, 363);

graph.DrawLine(brown, 615, 455, 658, 583);

graph.DrawLine(brown, 714, 660, 714, 800);

graph.DrawLine(brown, 1395, 695, 1395, 800);

graph.DrawLine(brown, 600, 363, 615, 455);

graph.DrawLine(new Pen(Color.Black, 3), 830, 680, 820, 800);

graph.DrawArc(brown, 605, 87, 46, 15, 185, 160);

graph.DrawArc(brown, 455, 225, 940, 940, 270, 90);

graph.DrawArc(brown, 436, 182, 15, 30, 100, 180);

graph.DrawArc(brown, 469, 323, 69, 44, 145, 32);

graph.DrawArc(brown, 644, 320, 196, 347, 99, 40);

graph.DrawArc(new Pen(Color.Black, 7), 590, 140, 55, 30, 180, 160);

graph.DrawArc(new Pen(Color.Black, 5), 397, 300, 180, 30, 10, 135);

graph.DrawArc(new Pen(Color.Black, 3), 1100, 500, 150, 350, 120, 150);

graph.FillEllipse(blackBrush, 425, 175, 70, 40);

}

}

}

**Результат работы программы**

При запуске программы рисуется капибара. Изображение показано на рисунке 6.

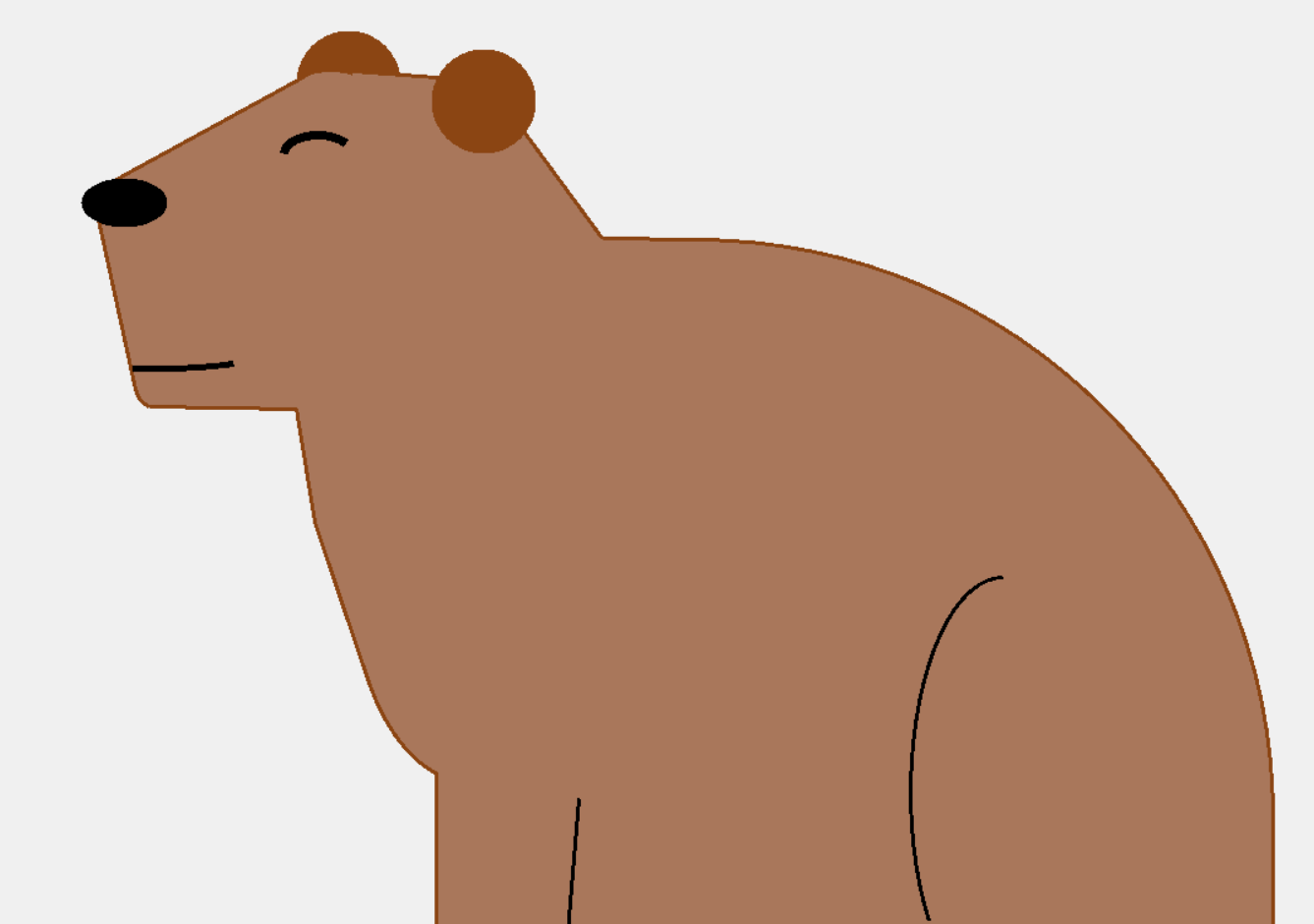


Рисунок 6 – Капибара