|  |  |
| --- | --- |
|  | Министерство науки и высшего образования РФ федеральное государственное образовательное учреждение высшего образования  **«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова» (БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова»)** |
| БГТУ.СМК-Ф-4.2-К5-01 |

|  |  |  |
| --- | --- | --- |
| Факультет | О | Естественнонаучный |
| Кафедра | шифр  О7 | Наименование  Информационные системы и программная инженерия |
|  | шифр | Наименование |

Дисциплина Информационные технологии и программирование

Практическая работа № 4

|  |
| --- |
| Тема работы |
| Изучение взаимодействия управляемого |
| и неуправляемого кода |
| Вариант 9 |

|  |  |
| --- | --- |
| Выполнил студент группы | О724Б |
| Кокорев М.Н. |  |
| Фамилия И.О. |  |
| **ПРЕПОДАВАТЕЛЬ** | |
| Землянская Е.Р. | |
| Фамилия И.О. Подпись | |
| Оценка |  |
| « » | 2023 г. |

## САНКТ-ПЕТЕРБУРГ 2023 г.

СОДЕРЖАНИЕ

[**1 Постановка задачи 3**](#_Toc149579619)

[**2 Реализация 4**](#_Toc149579620)

[**3 Результат работы программы 10**](#_Toc149579621)

# Постановка задачи

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.

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

Вариант 9.

Вычислить сумму элементов вещественного массива М(11), значения которых лежат введенном с клавиатуры диапазоне [X,Y].

# Реализация

Первая часть.

Текст динамической библиотеки

#include "pch.h"

#include "file.h"

DLL\_EXPORT void input(float arr[])

{

std::cout << "Введите 11 чисел: " << std::endl;

for (int i = 0; i < 11; i++) {

std::cin >> arr[i];

}

}

DLL\_EXPORT void output(float arr[])

{

std::cout << "Ваш массив: " << std::endl;

for (int i = 0; i < 11; i++) {

std::cout << arr[i]<<" ";

}

}

DLL\_EXPORT void summa(float arr[], int i, int j)

{

float sum = 0;

for (i; i <= j; i++) {

sum += arr[i];

}

cout <<"сумма ="<< sum;

}

Текст программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.InteropServices;

using System.Text;

using System.Threading.Tasks;

namespace c\_4MISHA

{

internal class Program

{

[DllImport("DLLForSharp.dll", CallingConvention = CallingConvention.Cdecl)]

public static extern void input(double[] array);

[DllImport("DLLForSharp.dll", CallingConvention = CallingConvention.Cdecl)]

public static extern void output(double[] array);

[DllImport("DLLForSharp.dll", CallingConvention = CallingConvention.Cdecl)]

public static extern void summa(double[] array,int i,int j);

static void Main()

{

double[] array = new double[11];

int menu;

int X, Y;

input(array);

do

{

Console.Clear();

Console.WriteLine("1.Ввод массива");

Console.WriteLine("2.Вывод массива");

Console.WriteLine("3.Обработка массива");

Console.WriteLine("0.Выход");

if (!Int32.TryParse(Console.ReadLine(), out menu))

{ Console.WriteLine("Некорректный ввод");

menu = -1;

}

switch (menu)

{

case 1:

Console.Clear();

input(array);

Console.ReadKey();

break;

case 2:

Console.Clear();

output(array);

Console.ReadKey();

break;

case 3:

Console.Clear();

do

{

Console.WriteLine("Введите X: ");

} while (!Int32.TryParse(Console.ReadLine(), out X)||X<0||X>9);

do

{

Console.WriteLine("Введите Y: ");

} while (!Int32.TryParse(Console.ReadLine(), out Y) || Y <= X || Y > 10);

summa(array,X,Y);

Console.ReadKey();

break;

}

} while (menu != 0);

}

}

}

## Вторая часть.

Текст программы  
using System;

using System.Drawing;

using System.Windows.Forms;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing.Drawing2D;

class CanvasForm : Form

{

private Bitmap canvas;

public CanvasForm()

{

// Установка размера окна

this.ClientSize = new Size(600, 600);

// Создание холста размером 600x550

canvas = new Bitmap(600, 600);

// Обработчик события Paint, вызываемого при отрисовке формы

this.Paint += PaintCanvas;

}

private void PaintCanvas(object sender, PaintEventArgs e)

{

// Получаем объект для рисования на холсте

Graphics g = this.CreateGraphics();

g.Clear(Color.Wheat);

// уши

// Рисуем уши

// Координаты вершин первого треугольника

Point[] triangle1Points = new Point[]

{

new Point(210, 105),

new Point(170, 220),

new Point(250, 210)

};

// Координаты вершин второго треугольника

Point[] triangle2Points = new Point[]

{

new Point(410, 120),

new Point(370, 220),

new Point(490, 210)

};

// Поворот треугольников на 30 градусов

PointF center1 = new PointF(200, 170); // Центр первого треугольника

PointF center2 = new PointF(400, 170); // Центр второго треугольника

float angle1 = 50;// Угол поворота в градусах

float angle2 = 90;

// Поворот первого треугольника

Matrix matrix1 = new Matrix();

matrix1.RotateAt(angle1, center1);

matrix1.TransformPoints(triangle1Points);

//Поворот второго треугольника

Matrix matrix2 = new Matrix();

matrix2.RotateAt(angle2, center2);

matrix2.TransformPoints(triangle2Points);

// Рисуем первый треугольник

g.FillPolygon(Brushes.Sienna, triangle1Points);

g.DrawPolygon(Pens.Sienna, triangle1Points);

// Рисуем второй треугольник

g.FillPolygon(Brushes.Sienna, triangle2Points);

g.DrawPolygon(Pens.Sienna, triangle2Points);

//тело

g.FillEllipse(Brushes.SaddleBrown, 150, 260, 300, 450);

g.DrawEllipse(Pens.SaddleBrown, 150, 260, 300, 450);

// голова

g.FillEllipse(Brushes.SaddleBrown, 200, 120, 200, 200);

g.DrawEllipse(Pens.SaddleBrown, 200, 120, 200, 200);

// глаза

g.FillEllipse(Brushes.Peru, 250, 200, 20, 20);

g.DrawEllipse(Pens.Gray, 250, 200, 20, 20);

g.FillEllipse(Brushes.Peru, 340, 200, 20, 20);

g.DrawEllipse(Pens.Gray, 340, 200, 20, 20);

//зрачки

g.FillEllipse(Brushes.Black, 255, 205, 8, 8);

g.DrawEllipse(Pens.Black, 255, 205, 8, 8);

g.FillEllipse(Brushes.Black, 345, 205, 8, 8);

g.DrawEllipse(Pens.Black, 345, 205, 8, 8);

// нос

GraphicsPath gn1 = new GraphicsPath();

gn1.AddLine(new Point(280, 235), new Point(330, 235));

gn1.AddLine(new Point(330, 235), new Point(305, 240));

gn1.CloseFigure();

g.FillPath(Brushes.IndianRed, gn1);

e.Graphics.DrawPath(Pens.MediumVioletRed, gn1);

// рот

g.DrawLine(Pens.MediumVioletRed, new Point(305, 240), new Point(305, 250));

g.DrawLine(Pens.MediumVioletRed, new Point(305, 250), new Point(295, 253));

g.DrawLine(Pens.MediumVioletRed, new Point(295, 253), new Point(291, 253));

g.DrawLine(Pens.MediumVioletRed, new Point(291, 253), new Point(285, 248));

g.DrawLine(Pens.MediumVioletRed, new Point(305, 250), new Point(315, 253));

g.DrawLine(Pens.MediumVioletRed, new Point(315, 253), new Point(319, 253));

g.DrawLine(Pens.MediumVioletRed, new Point(319, 253), new Point(325, 248));

// усы

g.DrawLine(Pens.Black, new Point(280, 240), new Point(250, 240));

g.DrawLine(Pens.Black, new Point(280, 240), new Point(230, 225));

g.DrawLine(Pens.Black, new Point(280, 240), new Point(230, 255));

g.DrawLine(Pens.Black, new Point(330, 240), new Point(360, 240));

g.DrawLine(Pens.Black, new Point(330, 240), new Point(380, 225));

g.DrawLine(Pens.Black, new Point(330, 240), new Point(380, 255));

// брови

g.DrawLine(Pens.Black, new Point(250, 195), new Point(270, 200));

g.DrawLine(Pens.Black, new Point(340, 200), new Point(360, 195));

//стакан белое

GraphicsPath gp3 = new GraphicsPath();

gp3.AddLine(new Point(300, 350), new Point(310, 470));

gp3.AddLine(new Point(310, 470), new Point(400, 470));

gp3.AddLine(new Point(400, 470), new Point(410, 350));

gp3.CloseFigure();

g.FillPath(Brushes.Silver, gp3);

e.Graphics.DrawPath(Pens.Silver, gp3);

// стакан кофе

GraphicsPath gp2 = new GraphicsPath();

gp2.AddLine(new Point(300, 350), new Point(330, 360));

gp2.AddLine(new Point(330, 360), new Point(370, 360));

gp2.AddLine(new Point(370, 360), new Point(410, 350));

gp2.AddLine(new Point(410, 350), new Point(370, 340));

gp2.AddLine(new Point(370, 340), new Point(330, 340));

gp2.CloseFigure();

g.FillPath(Brushes.Chocolate, gp2);

e.Graphics.DrawPath(Pens.MistyRose, gp2);

// хот дог корка

GraphicsPath gp4 = new GraphicsPath();

gp4.AddLine(new Point(150, 350), new Point(250, 470));

gp4.AddLine(new Point(250, 470), new Point(265, 450));

gp4.AddLine(new Point(265, 450), new Point(270, 430));

gp4.AddLine(new Point(270, 430), new Point(230, 350));

gp4.AddLine(new Point(230, 350), new Point(200, 330));

gp4.CloseFigure();

g.FillPath(Brushes.Peru, gp4);

e.Graphics.DrawPath(Pens.Peru, gp4);

// сосиска

GraphicsPath gp5 = new GraphicsPath();

gp5.AddLine(new Point(170, 340), new Point(270, 455));

gp5.AddLine(new Point(270, 455), new Point(280, 455));

gp5.AddLine(new Point(280, 455), new Point(280, 450));

gp5.AddLine(new Point(280, 450), new Point(270, 440));

gp5.AddLine(new Point(270, 440), new Point(190, 325));

gp5.AddLine(new Point(190, 325), new Point(180, 320));

gp5.AddLine(new Point(180, 320), new Point(175, 330));

gp5.CloseFigure();

g.FillPath(Brushes.Firebrick, gp5);

e.Graphics.DrawPath(Pens.Firebrick, gp5);

// рука 1

GraphicsPath gp6 = new GraphicsPath();

gp6.AddLine(new Point(160, 360), new Point(150, 410));

gp6.AddLine(new Point(150, 410), new Point(220, 470));

gp6.AddLine(new Point(220, 470), new Point(230, 475));

gp6.AddLine(new Point(230, 475), new Point(240, 470));

gp6.AddLine(new Point(240, 470), new Point(260, 450));

gp6.AddLine(new Point(260, 450), new Point(257, 440));

gp6.AddLine(new Point(257, 440), new Point(160, 360));

gp6.CloseFigure();

g.FillPath(Brushes.SaddleBrown, gp6);

e.Graphics.DrawPath(Pens.Black, gp6);

// рука 2

GraphicsPath gp7 = new GraphicsPath();

gp7.AddLine(new Point(430, 360), new Point(450, 410));

gp7.AddLine(new Point(450, 410), new Point(380, 470));

gp7.AddLine(new Point(380, 470), new Point(370, 475));

gp7.AddLine(new Point(370, 475), new Point(360, 470));

gp7.AddLine(new Point(360, 470), new Point(300, 450));

gp7.AddLine(new Point(300, 450), new Point(303, 440));

gp7.AddLine(new Point(303, 440), new Point(430, 360));

gp7.CloseFigure();

g.FillPath(Brushes.SaddleBrown, gp7);

e.Graphics.DrawPath(Pens.Black, gp7);

}

static void Main(string[] args)

{

Application.Run(new CanvasForm());

}

}

# Результат работы программы

На рисунке 1 показано меню программы.

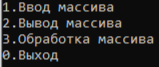


Рисунок 1 – Меню.

На рисунке 2 показан ввод массива.



Рисунок 2 – Ввод массива.

На рисунке 3 показан вывод массива.



Рисунок 3 – Вывод массива

На рисунке 4 показана сумма элементов в массиве в диапазоне.

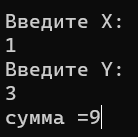


Рисунок 4 – Обработка массива

На рисунке 5 показан котик, нарисованный программным путём.

Изображение выглядит как рисунок, графическая вставка, иллюстрация, мультфильм

Автоматически созданное описание

Рисунок 5 – Кот