

ИШИТР ОИТ

09.03.01 «Информатика и вычислительная техника»

ОТЧЁТ

по Лабораторной работе № 6

**Программирование графики**

дисциплина:

**Компьютерная графика**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Исполнитель:** |  | | | | |
| студент группы 8В14 | | | В.О.Маслюк |  | 11.09.2023 |
|  |  |  |  |  | Дата сдачи |
| **Руководитель:** |  | | | | |
| к.т.н. доцент ОИТ ИШИТР А.Ю.Демин | | | | | |
|  |  |  |  |  |  |

Томск – 2023

**Цель работы**: изучить возможности Visual Studio по создание простейших графических изображений. Написать и отладить программу построения на экране различных графических примитивов

**Задание:**

Изучите с помощью справки MSDN методы и свойства классов Graphics, Color, Pen и SolidBrush. Создайте собственное приложение выводящий на форму рисунок, состоящий из различных объектов (линий, многоугольников, эллипсов, прямоугольников и пр.), не закрашенных и закрашенных полностью. Используйте разные цвета и стили линий (сплошные, штриховые, штрих-пунктирные).

**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;

using System.Xml;

namespace lab6

{

public partial class Form1 : Form

{

Pen BLACK\_PEN\_BOLD = new Pen(Brushes.Black, 3);

Pen BLACK\_PEN\_REGULAR = new Pen(Brushes.Black, 2);

Pen YELLOW\_PEN\_BOLD = new Pen(Color.FromArgb(255, 255, 255, 50), 3);

SolidBrush BROWN\_BRUSH = new SolidBrush(Color.FromArgb(255, 180, 67, 35));

SolidBrush GREEN\_BRUSH = new SolidBrush(Color.FromArgb(255, 42, 154, 78));

SolidBrush YELLOW\_BRUSH = new SolidBrush(Color.FromArgb(255, 255, 255, 50));

SolidBrush BLACK\_BRUSH = new SolidBrush(Color.FromArgb(255, 0, 0, 0));

SolidBrush BLUE\_BRUSH = new SolidBrush(Color.FromArgb(255, 12, 151, 231));

SolidBrush WHITE\_BRUSH = new SolidBrush(Color.FromArgb(255, 255, 255, 255));

SolidBrush GRAY\_BRUSH = new SolidBrush(Color.FromArgb(255, 196, 200, 197));

public Form1()

{

InitializeComponent();

}

private void Form1\_Paint(object sender, PaintEventArgs e)

{

Graphics g = e.Graphics;

g.Clear(Color.White);

DrawBackground(g);

DrawHouse(g, this.Width/2 + 20,this.Height-285,185, 185);

DrawSun(g);

DrawClouds(g);

DrawTree(g,350,this.Height - 300);

DrawTree(g, 120, this.Height - 300);

DrawGrass(g);

DrawFence(g);

}

private void DrawBackground(Graphics g)

{

g.FillRectangle(new SolidBrush(Color.FromArgb(180, 153, 234, 237)), 0, 0, this.Width - 1, this.Height - 1);

g.FillRectangle(new SolidBrush(Color.FromArgb(255, 69, 28, 3)), 0, this.Height - 100, this.Width - 1, this.Height - 100);

}

private void DrawHouse(Graphics g, int x\_start, int y\_start ,int width, int height)

{

//рисуем основу дома

g.FillRectangle(new SolidBrush(Color.FromArgb(220, 255, 0, 0)), x\_start + 1, y\_start + 1, width - 1, height - 1);

g.FillPolygon(new SolidBrush(Color.FromArgb(220, 255, 0, 0)),

new Point[] {

new Point(x\_start + 1,y\_start+1),

new Point(x\_start + width / 2, y\_start - height / 2),

new Point(x\_start+width - 1,y\_start - 1)

}

);

g.DrawLine(BLACK\_PEN\_BOLD, x\_start, y\_start, x\_start + width, y\_start);

g.DrawLine(BLACK\_PEN\_BOLD, x\_start + width, y\_start, x\_start + width, y\_start+ height);

g.DrawLine(BLACK\_PEN\_BOLD, x\_start + width, y\_start + height, x\_start, y\_start + height);

g.DrawLine(BLACK\_PEN\_BOLD, x\_start, y\_start + height, x\_start, y\_start);

g.DrawLine(BLACK\_PEN\_BOLD, x\_start, y\_start, x\_start + width / 2, y\_start - height / 2);

g.DrawLine(BLACK\_PEN\_BOLD, x\_start + width / 2, y\_start - height / 2, x\_start + width, y\_start);

//рисуем дверь

int doorWidth = 50;

int doorHeight = 90;

int doorXStart = x\_start + width - 80;

int doorYStart = y\_start + height;

g.FillRectangle(BROWN\_BRUSH, doorXStart+1, doorYStart - doorHeight + 1, doorWidth, doorHeight-2);

g.DrawLine(BLACK\_PEN\_REGULAR, doorXStart, doorYStart, doorXStart, doorYStart - doorHeight);

g.DrawLine(BLACK\_PEN\_REGULAR, doorXStart, doorYStart - doorHeight, doorXStart + doorWidth, doorYStart - doorHeight);

g.DrawLine(BLACK\_PEN\_REGULAR, doorXStart + doorWidth, doorYStart - doorHeight, doorXStart + doorWidth, doorYStart);

g.FillEllipse(BLACK\_BRUSH, doorXStart + doorWidth - 15, (doorYStart - doorHeight/2),7,7);

//рисуем окно

int windowSize = 45;

int windowXStart = x\_start + 25;

int windowYStart = y\_start + 65;

g.FillRectangle(BLUE\_BRUSH, windowXStart, windowYStart, windowSize, windowSize);

g.DrawLine(BLACK\_PEN\_REGULAR, windowXStart, windowYStart, windowXStart + windowSize, windowYStart);

g.DrawLine(BLACK\_PEN\_REGULAR, windowXStart + windowSize, windowYStart, windowXStart + windowSize, windowYStart + windowSize);

g.DrawLine(BLACK\_PEN\_REGULAR, windowXStart + windowSize, windowYStart + windowSize, windowXStart, windowYStart +windowSize);

g.DrawLine(BLACK\_PEN\_REGULAR, windowXStart, windowYStart + windowSize,windowXStart, windowYStart);

g.DrawLine(BLACK\_PEN\_REGULAR, windowXStart + windowSize/2, windowYStart, windowXStart + windowSize / 2, windowYStart+windowSize);

g.DrawLine(BLACK\_PEN\_REGULAR, windowXStart, windowYStart + windowSize/2, windowXStart + windowSize, windowYStart + windowSize / 2);

//рисуем окно на крыше

int roofWindowSize = 30;

int roofWindowXStart = x\_start + width / 2;

int roofWindowYStart = y\_start - 50;

g.FillEllipse(BLUE\_BRUSH, roofWindowXStart - roofWindowSize/2, roofWindowYStart, roofWindowSize, roofWindowSize);

g.DrawEllipse(BLACK\_PEN\_REGULAR, roofWindowXStart - roofWindowSize / 2, roofWindowYStart, roofWindowSize, roofWindowSize);

g.DrawLine(BLACK\_PEN\_REGULAR, roofWindowXStart, roofWindowYStart, roofWindowXStart, roofWindowYStart + roofWindowSize);

g.DrawLine(BLACK\_PEN\_REGULAR, roofWindowXStart - roofWindowSize/2, roofWindowYStart + roofWindowSize/2, roofWindowXStart + roofWindowSize/2, roofWindowYStart + roofWindowSize / 2);

//рисуем дымоход

int chimneyWidth = 25;

int chimneyHeight = 35;

int chimneyXStart = x\_start + 20;

int chimneyYStart = y\_start - 60;

g.FillPolygon(new SolidBrush(Color.FromArgb(255, 110, 107, 109)), new Point[]

{

new Point(chimneyXStart,chimneyYStart),

new Point(chimneyXStart, chimneyYStart + chimneyHeight+5),

new Point(chimneyXStart+chimneyWidth,chimneyYStart + chimneyHeight/2 - 1),

new Point(chimneyXStart+chimneyWidth,chimneyYStart)

});

g.DrawPolygon(BLACK\_PEN\_BOLD, new Point[]

{

new Point(chimneyXStart,chimneyYStart),

new Point(chimneyXStart, chimneyYStart + chimneyHeight+5),

new Point(chimneyXStart+chimneyWidth,chimneyYStart + chimneyHeight/2),

new Point(chimneyXStart+chimneyWidth,chimneyYStart)

});

//рисуем дым

int smokeXStart = chimneyXStart + 25;

int smokeYStart = chimneyYStart - chimneyHeight - 5;

g.FillClosedCurve(GRAY\_BRUSH, new Point[]

{

new Point(smokeXStart,smokeYStart),

new Point(smokeXStart - 20,smokeYStart + 20),

new Point(smokeXStart,smokeYStart - 20),

}) ;

}

private void DrawSun(Graphics g)

{

g.FillPie(YELLOW\_BRUSH, new Rectangle(-100, -100, 200, 200), 0, 90);

g.DrawLine(YELLOW\_PEN\_BOLD, 0, 0, 160, 80);

g.DrawLine(YELLOW\_PEN\_BOLD, 0, 0, 180, 30);

g.DrawLine(YELLOW\_PEN\_BOLD, 0, 0, 100, 150);

g.DrawLine(YELLOW\_PEN\_BOLD, 0, 0, 30, 180);

}

private void DrawClouds(Graphics g)

{

g.FillClosedCurve(WHITE\_BRUSH, GetCloudPoints(250, 60));

g.FillClosedCurve(WHITE\_BRUSH, GetCloudPoints(480, 60));

g.FillClosedCurve(WHITE\_BRUSH, GetCloudPoints(730, 60));

}

private Point[] GetCloudPoints(int x\_start, int y\_start)

{

Point[] cloudPoints = new Point[14];

int x\_increment = 30;

int y\_increment = 30;

cloudPoints[0] = new Point(x\_start, y\_start);

for (int i = 1; i < cloudPoints.Length; i++)

{

if(i == cloudPoints.Length / 2)

{

x\_increment = -x\_increment;

y\_increment = -y\_increment;

cloudPoints[i] = new Point(cloudPoints[i - 1].X, y\_start);

continue;

}

if(i % 2 != 0)

{

cloudPoints[i] = new Point(cloudPoints[i - 1].X + x\_increment, y\_start - y\_increment);

}

else

{

cloudPoints[i] = new Point(cloudPoints[i - 1].X + x\_increment, y\_start - y\_increment / 2);

}

}

cloudPoints[cloudPoints.Length - 1].Y = y\_start;

return cloudPoints;

}

private void DrawGrass(Graphics g)

{

int x\_start = 1;

int y\_start = this.Height - 100;

int pieceHeight = 10;

int pieceWidth = 15;

for (int i = 0; i < 34; i++)

{

g.FillPolygon(GREEN\_BRUSH, new Point[]

{

new Point(x\_start, y\_start),

new Point(x\_start, y\_start - pieceHeight),

new Point(x\_start + pieceWidth / 2, y\_start - pieceHeight - 10),

new Point(x\_start + pieceWidth, y\_start - pieceHeight),

new Point(x\_start + pieceWidth, y\_start)

}

);

x\_start += 15;

}

}

private void DrawFence(Graphics g)

{

int x = this.Width / 2 + 206;

int y = this.Height - 100;

for(int i = 0; i < 8; i++)

{

Point[] points = getFencePiecePoints(x, y);

g.FillPolygon(BROWN\_BRUSH, points);

g.DrawPolygon(BLACK\_PEN\_REGULAR, points);

x += 40;

}

}

private Point[] getFencePiecePoints(int x\_start, int y\_start)

{

Point[] fencePiecePoints = new Point[5];

int pieceHeight = 70;

int pieceWidth = 40;

fencePiecePoints[0] = new Point(x\_start, y\_start);

fencePiecePoints[1] = new Point(x\_start, y\_start - pieceHeight);

fencePiecePoints[2] = new Point(x\_start + pieceWidth/2, y\_start - pieceHeight - 10);

fencePiecePoints[3] = new Point(x\_start + pieceWidth, y\_start - pieceHeight);

fencePiecePoints[4] = new Point(x\_start + pieceWidth, y\_start);

return fencePiecePoints;

}

private void DrawTree(Graphics g, int x\_start, int y\_start)

{

g.FillRectangle(BROWN\_BRUSH, x\_start, y\_start, 40, 200);

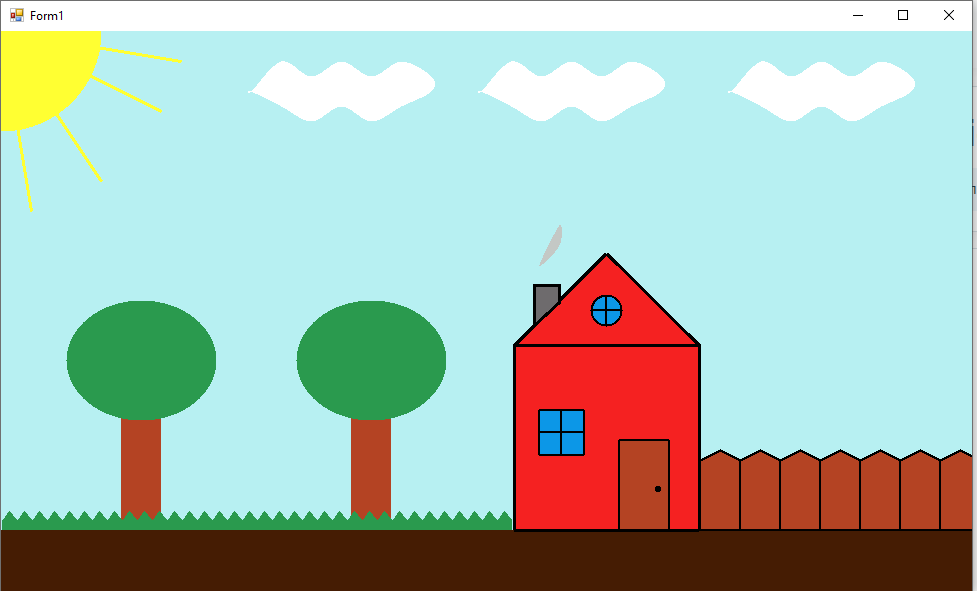
g.FillEllipse(GREEN\_BRUSH, x\_start-55, y\_start-30,150,120);

}

}

}

**Результат работы программы:**



**Вывод:** в ходе лабораторной работы изучили возможности Visual Studio по создание простейших графических изображений. Написали и отладили программу построения на экране различных графических примитивов