







//Задание:

//1.Вывести на экран поясняющую надпись для приведённой ниже задачи

//внутри закрашенного определённым пользователем образцом закраски

//многоугольника (количество углов = номер варианта\*2+1 для вариантов 1–7,

//количество углов = номер варианта для вариантов 8–15).

//2.Вывести данные, приведённые в номере варианта, в виде столбчатой

//диаграммы. Рисунок должен включать в себя линию нулевого уровня, метки

//элементов диаграмм, вспомогательные линии (горизонтальные штриховые

//полосы для сравнения высот стержней диаграмм). Прямоугольники столбчатой

//диаграммы заполнить, используя разные виды кистей SolidBrush, HatchBrash и

//TextureBrush.

//4. Результаты восьми экспериментов.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Drawing.Drawing2D;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

    public partial class Form1 : Form

    {

        public Form1()

        {

            InitializeComponent();

            this.Show();

            // Создаем графический контекст формы

            Graphics g = this.CreateGraphics();

            // Выводим подсказку

            g.DrawString("Кликните мышкой по элементу PictureBox",

            new Font("Arial", 10, FontStyle.Regular), Brushes.Red, 0, 0);

            // Уничтожаем графический контест формы

            g.Dispose();

        }

        private void Form1\_Load(object sender, EventArgs e)

        {

        }

        private void pictureBox1\_Click(object sender, EventArgs e)

        {

            // Очищаем форму

            this.Refresh();

            // Создаем графический контекст pictureBox1

            Graphics g = pictureBox1.CreateGraphics();

            // Задаем координаты точек многоугольника

            Point[] point = new Point[9] { new Point(120, 20),

            new Point(180, 30), new Point(240, 20),

            new Point(420, 30), new Point(420, 80),

            new Point(240, 90), new Point(180, 80),

            new Point(120, 90), new Point(130, 55)

            };

            //Point[] point = new Point[10] { new Point(120, 20),

            //new Point(180, 30), new Point(240, 20),

            //new Point(420, 30), new Point(420, 80),

            //new Point(240, 90), new Point(180, 80),

            //new Point(120, 90), new Point(120, 55),

            //new Point(100, 25)

            //};

            //Point[] point = new Point[13] { new Point(120, 20),

            //new Point(180, 30), new Point(240, 20),

            //new Point(300, 30), new Point(360, 20),

            //new Point(420, 30), new Point(420, 80),

            //new Point(360, 90), new Point(300, 80),

            //new Point(240, 90), new Point(180, 80),

            //new Point(120, 90), new Point(135, 55)

            //};

            // Рисуем многоугольник

            g.DrawPolygon(new Pen(Color.Blue, 2), point);

            // Задаем фонт и выранивание по центру

            Font fn = new Font("Arial", 10, FontStyle.Bold);

            StringFormat sf = new StringFormat();

            sf.Alignment = StringAlignment.Center;

            sf.LineAlignment = StringAlignment.Center;

            // Задаем и выводим поясняющую надпись

            string s = "Результаты восьми экспериментов";

            g.DrawString(s, fn, Brushes.Red, new Rectangle(125, 20, 300, 70), sf);

            // Рисуем рамку по периметру pictureBox1

            g.DrawRectangle(new Pen(Color.Blue, 1), 0, 0, pictureBox1.Width - 1, pictureBox1.Height - 1);

            // Задаем координаты начала и конца осей x и y

            int nachalo\_x = 30; int konec\_x = pictureBox1.Width - 10;

            int nachalo\_y = 120; int konec\_y = pictureBox1.Height - 20;

            // Рисуем ось x

            g.DrawLine(new Pen(Color.Black, 1), nachalo\_x, konec\_y,

            konec\_x, konec\_y);

            // Рисуем ось y

            g.DrawLine(new Pen(Color.Black, 1), nachalo\_x, nachalo\_y,

            nachalo\_x, konec\_y);

            // Задаем значения годов

            string[] year = new string[8] { "эксперимент 1", "эксперимент 2", "эксперимент 3", "эксперимент 4", "эксперимент 5", "эксперимент 6", "эксперимент 7", "эксперимент 8" };

            // сколько открытий/тысяч долларов и т.д. принёс эксперимент

            int[] value = new int[8] { 1, 2, 3, 4, 5, 6, 7, 8 };

            // Находим максимум в массиве заработных плат

            int max = -1;

            for (int i = 0; i < value.Length; i++) { if (value[i] > max) max = value[i]; }

            // Задаем масштабный делитель

            double mash = 5.0;

            // Определяем количество точек на единицу заработной платы

            double dy = (konec\_y - nachalo\_y) / (max / mash);

            // Задаем ширину прямоугольников диаграммы

            int widthRect = ((konec\_x - nachalo\_x) / value.Length) / 2;

            // Определяем сплошную закраску

            SolidBrush sb = new SolidBrush(Color.CornflowerBlue);

            // Определяем закраску штриховкой

            HatchBrush hb = new HatchBrush(HatchStyle.BackwardDiagonal,

            Color.CornflowerBlue, Color.LightSkyBlue);

            // Определяем закраску изображением

            Image img = Image.FromFile("C:\\Users\\User\\source\\repos\\2 курс\\лабы по комп.гр\\лаба 5\\WindowsFormsApp1\\img.bmp");

            TextureBrush tb = new TextureBrush(img);

            // Задаем начальную координату x

            int x = nachalo\_x + widthRect;

            // Организуем цикл по элементам массива заработных плат

            for (int i = 0; i < value.Length; i++)

            {

                // Задаем прямоугольную область элемента диаграммы

                Rectangle rect = new Rectangle(x, konec\_y - (int)(dy \* (value[i] / mash)), widthRect, (int)(dy \* (value[i] / mash)));

                // Выводим закрашенные разными стилями и цветом прямоугольники

                if (i < 3) g.FillRectangle(sb, rect);

                if ((i >= 3) && (i < 6)) g.FillRectangle(hb, rect);

                if ((i >= 6) && (i < 8)) g.FillRectangle(tb, rect);

                // Выводим рамку вокруг прямоугольника

                g.DrawRectangle(new Pen(Color.Black, 1), rect);

                // Переходим к отображению следующего элемента

                x += 2 \* widthRect;

            }

            // Разметка по оси Y

            Pen p = new Pen(Color.Blue, 2);

            p.DashStyle = System.Drawing.Drawing2D.DashStyle.Dash;

            fn = new Font("Arial", 8, FontStyle.Bold);

            sf.Alignment = StringAlignment.Near;

            sf.LineAlignment = StringAlignment.Center;

            for (int i = 0; i < value.Length; i++)

            {

                // Рисуем штрихпунктирные линии

                g.DrawLine(p, nachalo\_x - 5, konec\_y - (int)(dy \* (value[i] / mash)), konec\_x, konec\_y - (int)(dy \* (value[i] / mash)));

                // Выводим размеры зарплат

                g.DrawString(value[i].ToString(), fn, Brushes.Black,

                new Rectangle(1, konec\_y - (int)(dy \* (value[i] / mash)) - (int)fn.Size, 30, 20), sf);

            }

            // Разметка по оси X

            sf.Alignment = StringAlignment.Center;

            x = nachalo\_x + widthRect + widthRect / 2;

            for (int i = 0; i < year.Length; i++)

            {

                // Рисуем черточки по оси X

                g.DrawLine(new Pen(Color.Black, 1), x, konec\_y - 5, x, konec\_y + 5);

                // Выводим годы

                g.DrawString(year[i], fn, Brushes.Black, new Rectangle(x - 25, konec\_y, 50, 22), sf);

                x += 2 \* widthRect;

            }

        }

    }

}

Скриншот работы программы:

