## Задания по теме

**ЛР 7.** Решить 2 задачи с использованием абстрактного класса или интерфейса

Часть 1

Вывести n чисел в диапазоне [a; c] на форму в виде столбчатой диаграммы,

Где n берется из таблицы 3, а и с задаются самостоятельно

Таблица 3 — Варианты заданий

|  |  |
| --- | --- |
| 1 | 10 |
| 2 | 14 |
| 3 | 20 |
| 4 | 15 |
| 5 | 21 |
| 6 | 14 |
| 7 | 22 |
| 8 | 18 |
| 9 | 20 |
| 10 | 15 |

Часть 2.

Построить график функции и вывести минимальное и максимальное значения (отменить на графике или вывести в виде текста)

Таблица 4 — Задания

|  |  |
| --- | --- |
| 1 | y=|x|+2/x |
| 2 | y=3\*x-x^2 |
| 3 | y=x^3+4 |
| 4 | y=1/(x+2) |
| 5 | y=x^5 |
| 6 | y=sin (x)+1 |
| 7 | y=3\*|x|+x^2 |
| 8 | у=2\*х+4 |
| 9 | y=x^3-2 |
| 10 | y=1/(x+3) |

## Пример 11.3 PictureBox и Chart

Создан абстрактный класс с выводом на форму столбчатой диаграммы на компонент PictureBox, графика чисел и графика функции на компонент Chart.

Исходный код можно найти в папке

<https://disk.yandex.ru/d/Hs0QHM1uXWs6yg>

using System;

using System.Collections.Generic;

using System.Data;

using System.Drawing;

using System.Globalization;

using System.Linq;

using System.Windows.Forms;

using System.Windows.Forms.DataVisualization.Charting;

namespace BD\_Prakt\_Graph2

{

public partial class Form1 : Form

{

protected WebBrowser wb;

public Form1()

{

InitializeComponent();

wb = new WebBrowser();

//инициализируем WebBrowser для вычислений

wb.ScriptErrorsSuppressed = true;//убрать на этапе отладки, чтобы видеть ошибки

//подготавливаем JS-код для вычисления произвольного выражения

string js = @"function execute(script){return eval(script);}";

wb.DocumentText =

"<html><head><script type=\"text/javascript\">" +

js + "</script></head><body>Hello World</body></html>";

}

private void buttonNumber\_Chart\_Click(object sender, EventArgs e)

{

// Очищаем данные диаграммы

chart1.Series.Clear();

var provider = new RandomDataProvider();

var data = provider.GetData();

var rend = new ChartRenderer();

rend.drawChart(data, chart1);

}

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

{

// Очищаем данные диаграммы

chart1.Series.Clear();

var grf = new Funk();

grf.PlotGraph(chart1, "Math.pow(x,2)+6", Convert.ToInt32(tbBeg.Text),

Convert.ToInt32(tbEnd.Text), 1, wb);

//для вычисления выражений JavaScript в невидимом WebBrowsere

}

private void buttonNumber\_Picture\_Click(object sender, EventArgs e)

{

var provider = new RandomDataProvider();

var data = provider.GetData();

var rend = new ChartRenderer();

rend.drawPict(data, pictureBox1);

}

}

public abstract class ADataProvider

{

public abstract List<double> GetData();

}

public abstract class AChartRenderer

//Абстрактный класс

{

public abstract void drawChart(List<double> data, Chart chart1);

public abstract void drawPict(List<double> data, PictureBox pict);

}

public class RandomDataProvider : ADataProvider

{

public override List<double> GetData()

{

var random = new Random();

var data = new List<double>();

for (int i = 0; i < 10; i++)

{

data.Add(random.NextDouble() \* 100);

}

return data;

}

}

public class ChartRenderer : AChartRenderer

{

public override void drawPict(List<double> data, PictureBox pictureBox1)

{

Bitmap bmp = new Bitmap(pictureBox1.Width, pictureBox1.Height);

Graphics g = Graphics.FromImage(bmp);

Pen pen = new Pen(Color.Black);

Font font = new Font("Arial", 8);

double maxValue = double.MinValue;

double minValue = double.MaxValue;

// Определяем минимальное и максимальное значения

foreach (double value in data)

{

if (value > maxValue)

maxValue = value;

if (value < minValue)

minValue = value;

}

// Рисуем график

int barWidth = pictureBox1.Width / data.Count;

for (int i = 0; i < data.Count; i++)

{

int barHeight = (int)(pictureBox1.Height \* (data[i] - 0) / (maxValue - 0) \* 0.7); //Уменьшение масштаба

g.FillRectangle(Brushes.Blue, i \* barWidth, pictureBox1.Height - barHeight, barWidth, barHeight);

g.DrawString(Math.Round(data[i], 2).ToString(), font, Brushes.Black, i \* barWidth, pictureBox1.Height - barHeight - 15);

}

// Определяем размер минимального и максимального столбцов

int minr = (int)(pictureBox1.Height \* (minValue - 0) / (maxValue - 0) \* 0.7);

int maxr = (int)(pictureBox1.Height \* (maxValue - 0) / (maxValue - 0) \* 0.7);

// Отмечаем минимальное и максимальное значения

g.FillRectangle(Brushes.Red, data.IndexOf(minValue) \* barWidth, pictureBox1.Height - minr, barWidth, minr);

g.FillRectangle(Brushes.Red, data.IndexOf(maxValue) \* barWidth, pictureBox1.Height - maxr, barWidth, maxr);

pictureBox1.Image = bmp;

}

public override void drawChart (List<double> data, Chart chart1)

{

//chart1.ChartAreas[0].AxisX.Interval = 2; //сетка

//chart1.ChartAreas[0].AxisY.Interval = 10;

Series series = chart1.Series.Add("Числ.ряд");

series.ChartType = SeriesChartType.Line;

for (int i = 0; i < data.Count; i++)

{

series.Points.AddXY(i + 1, data[i]);

}

double min = data[0];

double max = data[0];

int minIndex = 0;

int maxIndex = 0;

for (int i = 0; i < data.Count; i++)

{

if (min > data[i])

{

min = data[i];

minIndex = i;

}

if (max < data[i])

{

max = data[i];

maxIndex = i;

}

}

series.Points[minIndex].Label = "Min: " + Math.Round(min,2);

series.Points[minIndex].MarkerStyle = MarkerStyle.Circle;

series.Points[minIndex].MarkerSize = 10;

series.Points[maxIndex].Label = "Max: " + Math.Round(max, 2); ;

series.Points[maxIndex].MarkerStyle = MarkerStyle.Circle;

series.Points[maxIndex].MarkerSize = 10;

chart1.Invalidate();

}

}

}

public abstract class AFunk

{

public abstract void PlotGraph(Chart ch, string expression, double x\_min, double x\_max, double dx, WebBrowser wb);

}

public class Funk : AFunk

{

public override void PlotGraph(Chart ch, string expression, double x\_min, double x\_max, double dx, WebBrowser wb)

{

DataTable dt = new DataTable();

dt.Columns.Add("x", typeof(Double));

dt.Columns.Add("y", typeof(Double));

DataRow row;

double x;

string s;

object y;

//создаем культуру, совместимую с представлением чисел в JS

CultureInfo culture =

new CultureInfo(CultureInfo.CurrentCulture.Name);

culture.NumberFormat.NegativeSign = "-";

culture.NumberFormat.NumberDecimalSeparator = ".";

//создаем таблицу значений

for (x = x\_min; x <= x\_max; x += dx)

{

row = dt.NewRow();

row["x"] = x;

//подготавливаем выражение для JS

s = "var x=" + x.ToString(culture) + "; " + expression + ";";

//вычисляем выражение

y = wb.Document.InvokeScript("execute", new object[] { s });

//преобразуем результат из строки в Double

row["y"] = Convert.ToDouble(y, culture);

dt.Rows.Add(row);

}

// Находим максимальное значение Y

double maxY = dt.AsEnumerable().Max(r => r.Field<double>("Y"));

// Находим минимальное значение Y

double minY = dt.AsEnumerable().Min(r => r.Field<double>("Y"));

// Получаем все точки, у которых значение Y равно максимальному

var maxPoints = dt.AsEnumerable().Where(r => r.Field<double>("Y") == maxY).Select(r => new { X = r.Field<double>("X"), Y = r.Field<double>("Y") }).ToList();

// Получаем все точки, у которых значение Y равно минимальному

var minPoints = dt.AsEnumerable().Where(r => r.Field<double>("Y") == minY).Select(r => new { X = r.Field<double>("X"), Y = r.Field<double>("Y") }).ToList();

//добавляем ряд в Chart

ch.DataSource = dt;

Series ser = new Series();

ser.Name = expression;

ser.XValueMember = "x";

ser.YValueMembers = "y";

ser.ChartType = SeriesChartType.Line;

// ch.Series.Clear();

Series series = new Series("Max и min");

series.ChartType = SeriesChartType.Point;

series.Color = Color.Red;

// Добавляем максимальные точки на график в виде красных маркеров

foreach (var point in maxPoints)

{

series.Points.AddXY(point.X, point.Y);

series.Points.Last().Color = System.Drawing.Color.Red;

series.Points.Last().MarkerSize = 10;

series.Points.Last().MarkerStyle = MarkerStyle.Circle;

}

// Добавляем минимальные точки на график в виде красных маркеров

foreach (var point in minPoints)

{

series.Points.AddXY(point.X, point.Y);

series.Points.Last().Color = System.Drawing.Color.Red;

series.Points.Last().MarkerSize = 10;

series.Points.Last().MarkerStyle = MarkerStyle.Circle;

}

ch.Series.Add(ser);

ch.Series.Add(series);

ch.ChartAreas[0].AxisX.Minimum = x\_min; //начало графика- минимальное зн. х

ch.ChartAreas[0].AxisX.Interval = 5; //сетка

// ch.ChartAreas[0].AxisY.Interval = 20;

}

}

Изображение выглядит как текст, снимок экрана, диаграмма, График

Автоматически созданное описание

1. Визуализация ряда

Изображение выглядит как текст, снимок экрана, диаграмма, График

Автоматически созданное описание

1. Визуализация графика функции