**Самченко ІТІНФ-20-1 В-21**

Создать класс Point, разработав следующие элементы класса:

* + Поля:
    - int x, y;
  + Конструкторы, позволяющие создать экземпляр класса:
    - с нулевыми координатами;
    - с заданными координатами.
  + Методы, позволяющие:
    - вывести координаты точки на экран;
    - рассчитать расстояние от начала координат до точки;
    - переместить точку на плоскости на вектор (a, b).
  + Свойства:
    - получить-установить координаты точки (доступное для чтений и записи);
    - позволяющие умножить координаты точки на скаляр (доступное для чтений и записи).

**Class Point:**

using System;

using System.Collections.Generic;

namespace Laba4 {

/// <summary>

/// Класс Points используется для хранения информации об n-коичестве точек.

/// </summary>

[Serializable]

public class Points {

/// <summary>

/// Конструктор по умолчанию класса Points.

/// </summary>

public Points() {}

public List<Point> PointList { get; set; } = new List<Point>();

}

/// <summary>

/// Класс Point используется для обработки данных конкретной точки.

/// </summary>

[Serializable]

public class Point {

public int x { get; set; }

public int y { get; set; }

public String name { get; set; }

/// <summary>

/// Конструктор по умолчанию класса Point.

/// </summary>

public Point() {

x = 0;

y = 0;

name = "noname" ;

}

/// <summary>

/// Конструктор с параметрами класса Point.

/// </summary>

public Point(string name, int x, int y) {

this.x = x;

this.y = y;

this.name = name;

}

/// <summary>

/// Конструктор копирования класса Point.

/// </summary>

public Point(string name) {

this.x = 0;

this.y = 0;

this.name = name;

}

/// <summary>

/// Метод для расчета дистанции от начала координат до местоположения точки.

/// </summary>

/// <example>

/// Как используется метод Distance().

/// <code>

/// class TestClass {

/// static int Main() {

/// Point point;

/// point.Distance();

/// }

/// }

/// </code>

/// </example>

public double Distance() {

return Math.Sqrt(x \* x + y \* y);

}

/// <summary>

/// Метод для перемещения точки по осям х и у.

/// </summary>

/// <example>

/// Как используется метод Move().

/// <code>

/// class TestClass {

/// static int Main() {

/// Point point;

/// point.Move(2, 5);

/// }

/// }

/// </code>

/// </example>

public void Move(int a, int b) {

x += a;

y += b;

}

/// <summary>

/// Метод для умножения значений х и у точки на заданный скаляр.

/// </summary>

/// /// Как используется метод Scalar().

/// <code>

/// class TestClass {

/// static int Main() {

/// Point point;

/// point.Scalar(2);

/// }

/// }

/// </code>

/// </example>

public void Scalar(int value) {

x = x \* value;

y = y \* value;

}

}

}

**Form1:**

using System;

using System.Windows.Forms;

using System.Xml.Serialization;

using System.IO;

namespace Laba4 {

public partial class Form1 : Form {

public Form1() {

InitializeComponent();

Clear();

}

/// <summary>

/// Функция очищающая все TextBox на форме

/// </summary>

private void Clear() {

textBoxName.Clear();

textBoxX.Clear();

textBoxY.Clear();

textBoxLong.Clear();

textBoxMoveX.Clear();

textBoxMoveY.Clear();

textBoxScalyar.Clear();

}

/// <summary>

/// Функция отвечающая за добавление нового эдемента в список ListViewItem (LVI)

/// </summary>

/// <param name="point"></param>

private void Add(Point point) {

ListViewItem LVI = new ListViewItem(point.name);

LVI.Tag = point;

listView1.Items.Add(LVI);

}

/// <summary>

/// Функция отвечающая за Сериализацию данных из списка ListViewItem, открывая SaveFileDialog .

/// Запись сериализацинных данных в файл

/// Сохраняемый тип .xml

/// </summary>

/// <param name="points"></param>

private void SerializeXML(Points points) {

XmlSerializer xml = new XmlSerializer(typeof(Points));

SaveFileDialog saveFile = new SaveFileDialog();

saveFile.Filter = "XML Files|\*.xml";

saveFile.Title = "Save a Xml File";

saveFile.ShowDialog();

if (saveFile.FileName != "") {

using (FileStream fs = (FileStream)saveFile.OpenFile()) {

xml.Serialize(fs, points);

}

}

MessageBox.Show("Файл сохранен");

}

/// <summary>

/// Функция отвечающая за Десериализацию данных из файла, открывая OpenFileDialog .

/// Запись сериализацинных данных в ListViewItem

/// Чтение производится из файлов типа .xml

/// </summary>

/// <returns></returns>

private Points DeserializeXML() {

XmlSerializer xml = new XmlSerializer(typeof(Points));

Stream myStream = null;

OpenFileDialog openFile = new OpenFileDialog();

openFile.Filter = "XML Files|\*.xml";

openFile.Title = "Open a Xml File";

openFile.ShowDialog();

using (FileStream fs = (FileStream)openFile.OpenFile()) {

return (Points)xml.Deserialize(fs);

}

}

/// <summary>

/// Нажатие на кнопку button4\_Click создает новый обьект класса Point <see cref="Point"/>,

/// с проверкой следущих полей: textboxX, TextBoxY, TextBoxName

/// Исходя из пройденной проверки используеться определенный конструктор <see cref="Point()"/>,

/// и добавляеться в LVI

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void button4\_Click(object sender, EventArgs e) {

if ((textBoxName.Text == string.Empty)&&(textBoxX.Text == string.Empty)&&(textBoxY.Text == string.Empty)) {

Point point = new Point();

Add(point);

} else if ((textBoxX.Text == string.Empty)&&(textBoxY.Text == string.Empty)) {

Point point = new Point(textBoxName.Text);

Add(point);

} else {

Point point = new Point(textBoxName.Text, int.Parse(textBoxX.Text), int.Parse(textBoxY.Text));

Add(point);

}

Clear();

}

/// <summary>

/// В listView1 находится список всех точек.

/// Нажав на любую из них можно увидеть координаты по оси х и по оси у.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void listView1\_SelectedIndexChanged(object sender, EventArgs e) {

if (listView1.SelectedItems.Count == 1) {

Point point = (Point)listView1.SelectedItems[0].Tag;

if (point != null) {

textBoxName.Text = point.name;

textBoxX.Text = Convert.ToString(point.x);

textBoxY.Text = Convert.ToString(point.y);

}

} else if (listView1.SelectedItems.Count == 0) {

Clear();

}

}

/// <summary>

/// Нажав на кнопку buttonmove выбранная точка переместится на новое местоположение

/// <see cref="Point.Move(int, int)"/>.

/// textBoxMoveX.Text и textBoxMoveУ.Text хранят в себе значения, которые будут пребавлены

/// к значениям Х и У точки соответственно.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonmove\_Click(object sender, EventArgs e) {

if (listView1.SelectedItems.Count == 1) {

Point point = (Point)listView1.SelectedItems[0].Tag;

if ((textBoxMoveX.Text != string.Empty)&&(textBoxMoveY.Text != string.Empty)) {

point.Move(int.Parse(textBoxMoveX.Text), int.Parse(textBoxMoveY.Text));

textBoxX.Text = Convert.ToString(point.x);

textBoxY.Text = Convert.ToString(point.y);

}

} else if (listView1.SelectedItems.Count == 0) {

Clear();

}

}

/// <summary>

/// Нажав на кнопку buttonlong\_Click в поле textBoxLong будет выведено

/// значение расстояния данной точки от начала координат <see cref="Point.Distance()"/>

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonlong\_Click(object sender, EventArgs e) {

if (listView1.SelectedItems.Count == 1) {

Point point = (Point)listView1.SelectedItems[0].Tag;

textBoxLong.Text = Convert.ToString(Math.Round(point.Distance(),4));

} else if (listView1.SelectedItems.Count == 0) {

Clear();

}

}

/// <summary>

/// Нажатие на кнопку buttonscalyar\_Click позволяет умножить координаты точки на скаляр

/// <see cref="Point.Scalar(int)"/>, значение которого записывается пользователем в поле textBoxScalyar

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonscalyar\_Click(object sender, EventArgs e) {

if (listView1.SelectedItems.Count == 1) {

Point point = (Point)listView1.SelectedItems[0].Tag;

point.Scalar(int.Parse(textBoxScalyar.Text));

textBoxX.Text = Convert.ToString(point.x);

textBoxY.Text = Convert.ToString(point.y);

} else if (listView1.SelectedItems.Count == 0) {

Clear();

}

}

/// <summary>

/// Нажатие на кнопку buttonSerialize\_Click позволяет преобразовать данные в поток байтов

/// <see cref="SerializeXML"/>.

/// После преобразования этот поток байтов можно записать на диск в XML файл.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonSerialize\_Click(object sender, EventArgs e) {

Points points = new Points();

foreach (ListViewItem item in listView1.Items) {

if (item.Tag != null) {

points.PointList.Add((Point)item.Tag);

}

}

SerializeXML(points);

}

/// <summary>

/// Нажатие на кнопку button1\_Click позволяет считать данные с XML файла

/// <see cref="DeserializeXML"/> и запустить их в программе

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void button1\_Click(object sender, EventArgs e) {

Clear();

Points points = DeserializeXML();

foreach (Point point in points.PointList) {

Add(point);

}

}

}

}

Изображение выглядит как текст, внутренний, снимок экрана, компьютер

Автоматически созданное описание