# Практическая работа № 3 Построение графиков

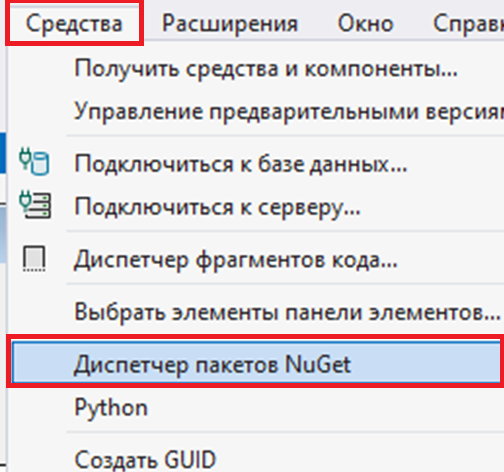
**Цель практической работы**

Получение навыков по рисованию графиков на C#.

**Постановка задачи**

**Построение графиков с помощью ScottPlot.**

Рассмотрим пример добавления к проекту библиотеки для построения графиков ScottPlot. Необходимо открыть диспетчер пакетов **NuGet** (рисунок 1).



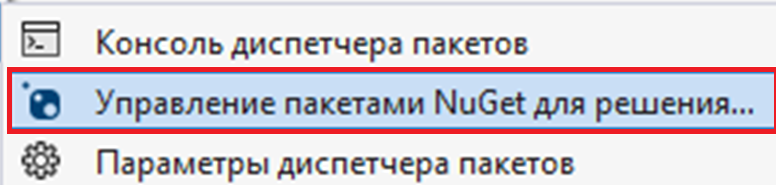


Рисунок 1 – Установка пакетов с помощью NuGet

Необходимо перейти на вкладку «Обзор» найти библиотеки ScottPlot и ScottPlot.WinForms, и нажать «Установить» (рисунок 2).

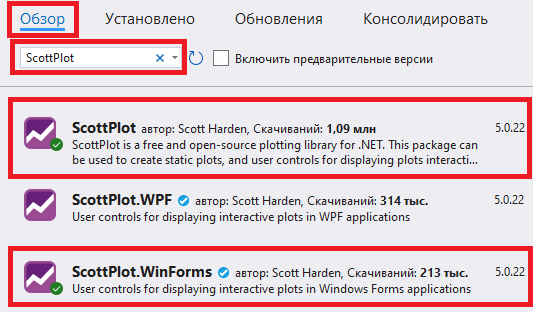


Рисунок 2 – Установка пакета ScottPlot

Элемент для построения графиков (FormsPlot) появится в панели элементов (рисунок 3). После установки пакета **возможно придется перезапустить приложение** (если появляется ошибка при добавлении элемента FormsPlot).

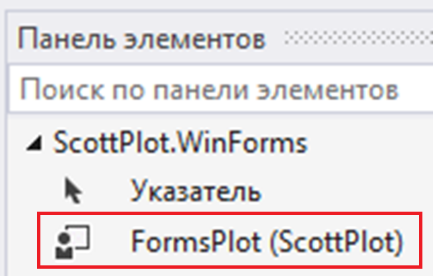


Рисунок 3 – Отображение ScottPlot в панели элементов

Добавим подписи осей, название графика и отобразим что получилось (рисунок 4).

formsPlot1.Plot.XLabel("X, м");

formsPlot1.Plot.YLabel("Y, м");

formsPlot1.Plot.Title("График");

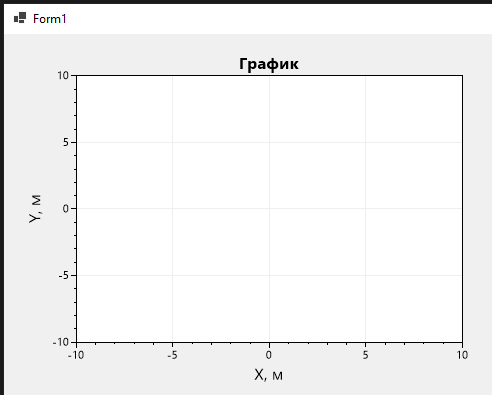


Рисунок 4 – Пустой график ScottPlot на форме

Построим график функции *f(x) = x*2 на участке от 0 до 100 (рисунок 5).

var xs = new List<double>();

var ys = new List<double>();

for (int i = 0; i <= 100; i++)

{

xs.Add(i);

ys.Add(i \* i);

}

var scatter = formsPlot1.Plot.Add.Scatter(xs, ys);

// Убираем маркеры (точки) и оставляем линию

scatter.MarkerStyle = ScottPlot.MarkerStyle.None;

scatter.Label = "x^2"; // Подписии линии графика

formsPlot1.Plot.ShowLegend();

plot.Axes.AutoScale(); // Автоматически приближать к области графика

formsPlot.Refresh(); // Вызов метода перерисовки

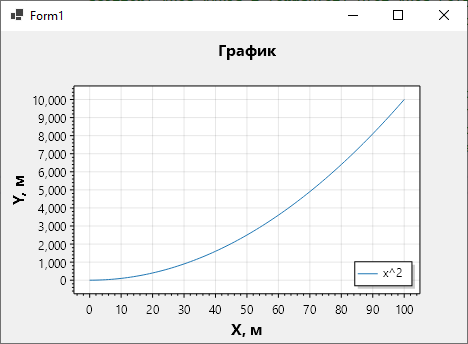


Рисунок 5 – Отображение графика функции на форме

Основное преимущество ScottPlot это **интерактивность**, график можно масштабировать и перемещаться по графику, также он сам определяет интервалы подписей у осей.

**Чтобы переместиться по графику** нужно зажать ЛКМ и двигать мышью, чтобы приблизиться (рисунок 6) или отдалиться – вращать колесико мыши, также чтобы приблизить некоторую область можно зажать колесико и выделить прямоугольник. Чтобы вернуться к исходному виду нужно нажать на колесико.

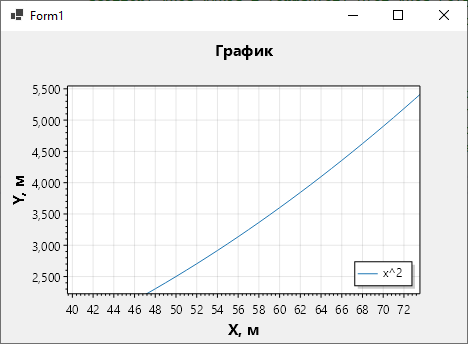


Рисунок 6 – Приближение участка графика

ScottPlot поддерживает рисование по второстепенным осям, это полезно, когда 2 набора данных сильно отличаются по значениям, но их требуется представить на одном графике (Например, давление и температура).

var xs = new List<double>();

var ys = new List<double>();

var ys2 = new List<double>();

for (int i = 0; i <= 100; i++)

{

xs.Add(i);

ys.Add(i \* i);

ys2.Add(Math.Sin(i));

}

Отобразим *x*2 на основной оси, а sin(*x*) на второстепенной оси *Oy.* Также сделаем график синуса пунктирным (рисунок 7).

var scatter1 = formsPlot1.Plot.Add.Scatter(xs, ys);

scatter1.Axes.XAxis = formsPlot1.Plot.Axes.Bottom;

scatter1.Axes.YAxis = formsPlot1.Plot.Axes.Left;

scatter1.MarkerStyle = ScottPlot.MarkerStyle.None;

scatter1.Label = "x^2";

var scatter2 = formsPlot1.Plot.Add.Scatter(xs, ys2);

scatter2.Axes.XAxis = formsPlot1.Plot.Axes.Bottom;

scatter2.Axes.YAxis = formsPlot1.Plot.Axes.Right;

scatter2.MarkerStyle = ScottPlot.MarkerStyle.None;

scatter2.LinePattern = ScottPlot.LinePattern.Dashed;

scatter2.Label = "sin(x)";

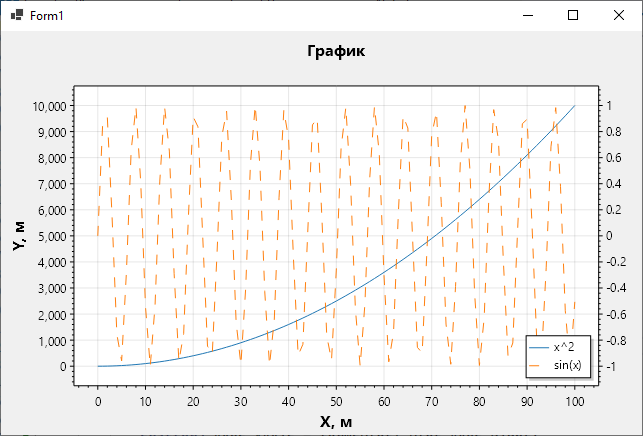


Рисунок 7 – Отображение графиков  
на главной и второстепенной осях

Рассмотрим пример построения гистограмм (рисунок 8):

var plot = formsPlot1.Plot;

double[] xs1 = { 1, 2, 3, 4 };

double[] ys1 = { 5, 10, 7, 13 };

var bars1 = plot.Add.Bars(xs1, ys1);

bars1.Label = "Alpha";

double[] xs2 = { 6, 7, 8, 9 };

double[] ys2 = { 7, 12, 9, 15 };

var bars2 = plot.Add.Bars(xs2, ys2);

bars2.Label = "Beta";

plot.ShowLegend(Alignment.UpperLeft);

// Убираем отсут от нижней границы

plot.Axes.Margins(bottom: 0);

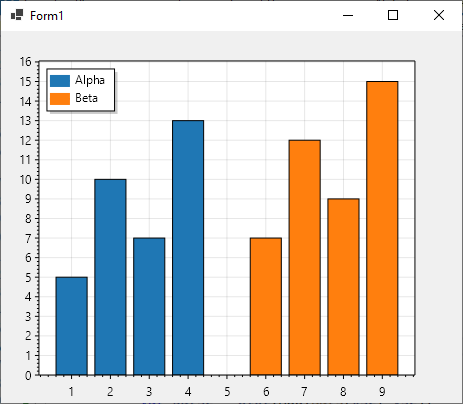


Рисунок 8 – Отображение гистограммы

Рассмотрим пример построения круговой диаграммы (рисунок 9):

var plot = formsPlot1.Plot;

List<PieSlice> slices = new()

{

new PieSlice(){ Value = 5, FillColor = Colors.Red, Label = "Red" },

new PieSlice(){ Value = 2, FillColor = Colors.Orange, Label ="Orange"},

new PieSlice(){ Value = 8, FillColor = Colors.Gold, Label ="Yellow"},

new PieSlice(){ Value = 4, FillColor = Colors.Green, Label ="Green"},

new PieSlice(){ Value = 8, FillColor = Colors.Blue, Label ="Blue" }};

var pie = plot.Add.Pie(slices);

pie.ExplodeFraction = 0.1;

plot.ShowLegend();

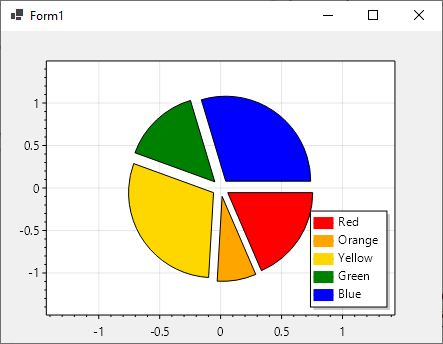


Рисунок 9 – Отображение гистограммы

**Анимация.**

Здесь необходим компонент Timer. Требуется добавить 2 кнопки – для запуска и остановки движения.

public partial class Form1 : Form

{

private double x = 0, speed = 0.1;

private List<double> xs = new();

private List<double> ys = new();

public Form1()

{

InitializeComponent();

}

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

{

timer.Start();

}

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

{

timer.Stop();

}

private double getF(double x) => Math.Sin(x);

private void timer\_Tick(object sender, EventArgs e)

{

x += speed;

xs.Add(x);

ys.Add(getF(x));

var plot = formsPlot.Plot;

plot.Clear();

var scatter = plot.Add.Scatter(xs, ys);

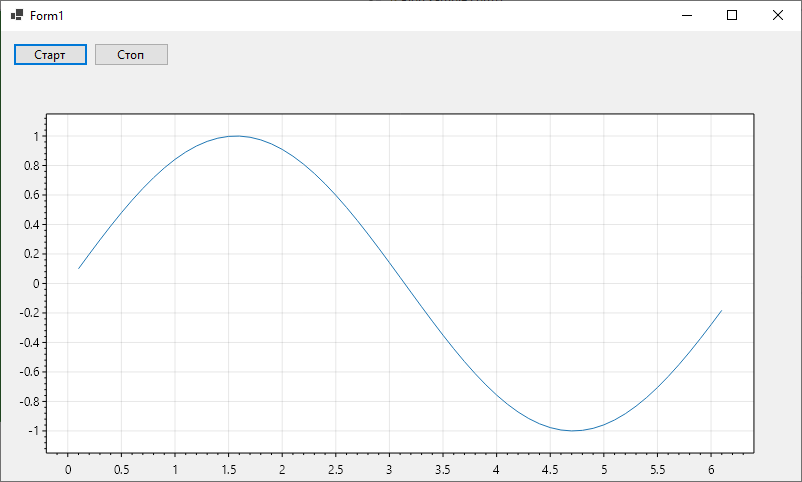
scatter.MarkerStyle = MarkerStyle.None;

plot.Axes.AutoScale();

formsPlot.Refresh();

}

}

****

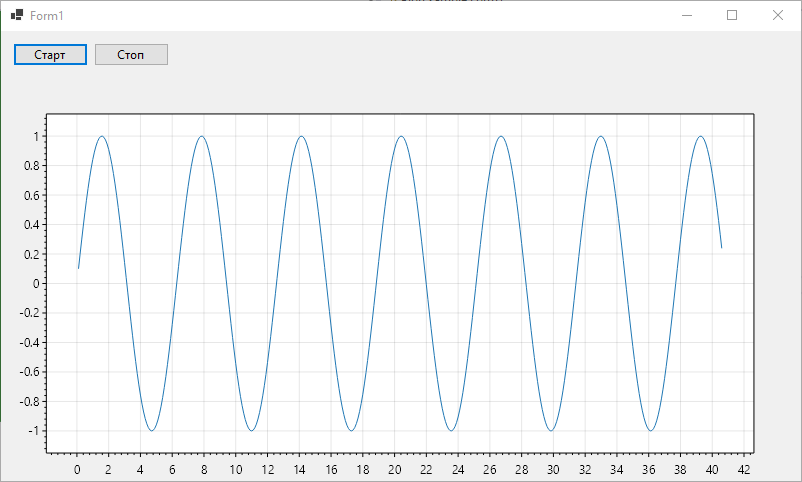
****

Рисунок 10 – Отображение анимации синусоиды

Примеры построения графиков и работы в ScottPlot приводятся в документации на сайте: <https://scottplot.net/cookbook/5.0/>

**Задание на практическую работу**

1. Разработать WindowsForms-приложение для построения графиков.
2. Предусмотреть понятный, аккуратный интерфейс ввода и вывода данных.

**Задание.**

Используя метод CubicSpline.InterpolateNatural найти промежуточные значения табличных данных и построить график функции.

В качестве данных используем таблицу стандартной атмосферы. Независимая переменная (*x*) – высота, м. Зависимая переменная (*y*) – по варианту.

**Варианты:**

1. температура,
2. скорость звука,
3. давление,
4. плотность,
5. кинематическая вязкость.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Высота, H, м | Температура, T, К | Скорость звука, a, м/с | Давление, P, Па | Плотность, ρ, кг/м3 | Кинематическая вязкость, ν, м2/с |
| -2000 | 301,2 | 347,9 | 127783 | 1,4782 | 1,2525⋅10−5 |
| -1500 | 297,9 | 346,0 | 120696 | 1,4114 | 1,3009⋅10−5 |
| -1000 | 294,7 | 344,1 | 113931 | 1,3470 | 1,3516⋅10−5 |
| -500 | 291,4 | 342,2 | 107478 | 1,2849 | 1,4048⋅10−5 |
| 0 | 288,2 | 340,3 | 101330 | 1,2250 | 1,46⋅10−5 |
| 500 | 284,9 | 338,4 | 95464 | 1,1673 | 1,52⋅10−5 |
| 1000 | 281,7 | 336,4 | 89877 | 1,1117 | 1,58⋅10−5 |
| 1500 | 278,4 | 334,5 | 84559 | 1,0581 | 1,65⋅10−5 |
| 2000 | 275,2 | 332,5 | 79499 | 1,0065 | 1,71⋅10−5 |
| 2500 | 271,9 | 330,6 | 74690 | 0,9569 | 1,79⋅10−5 |
| 3000 | 268,7 | 328,6 | 70123 | 0,9093 | 1,86⋅10−5 |
| 4000 | 262,2 | 324,6 | 61661 | 0,8194 | 2,03⋅10−5 |
| 5000 | 255,7 | 320,6 | 54052 | 0,7365 | 2,21⋅10−5 |
| 6000 | 249,2 | 316,5 | 47217 | 0,6601 | 2,42⋅10−5 |
| 7000 | 242,7 | 312,3 | 41106 | 0,59 | 2,65⋅10−5 |
| 8000 | 236,2 | 308,1 | 35653 | 0,5258 | 2,9⋅10−5 |
| 9000 | 229,7 | 303,9 | 30801 | 0,4671 | 3,2⋅10−5 |
| 10 000 | 223,3 | 299,6 | 26500 | 0,4135 | 3,53⋅10−5 |
| 11 000 | 216,8 | 295,2 | 22700 | 0,3648 | 3,9⋅10−5 |
| 12 000 | 216,7 | 295,1 | 19399 | 0,3119 | 4,56⋅10−5 |
| 14 000 | 216,7 | 295,1 | 14170 | 0,2279 | 6,24⋅10−5 |
| 16 000 | 216,7 | 295,1 | 10353 | 0,1665 | 8,54⋅10−5 |
| 18 000 | 216,7 | 295,1 | 7565 | 0,1216 | 1,17⋅10−4 |
| 20 000 | 216,7 | 295,1 | 5529 | 0,0889 | 1,6⋅10−4 |
| 24 000 | 220,6 | 297,7 | 2971 | 0,0469 | 3,07⋅10−4 |
| 28 000 | 224,5 | 300,4 | 1616 | 0,0251 | 5,84⋅10−4 |
| 32 000 | 228,5 | 303 | 889 | 0,0136 | 1,1⋅10−3 |
| 36 000 | 239,3 | 310,1 | 499 | 7,26⋅10−3 | 2,13⋅10−3 |
| 40 000 | 250,4 | 317,2 | 287 | 4,00⋅10−3 | 4,01⋅10−3 |
| 50 000 | 270,7 | 329,8 | 80 | 1,03⋅10−3 | 0,0166 |
| 60 000 | 247 | 315,1 | 22 | 3,00⋅10−4 | 0,0511 |
| 80 000 | 198,6 | 282,5 | 1 | 1,85⋅10−5 | 0,716 |

**Пример разработки приложения**

**Задание.**

Пусть данные метеоизмерений заданы таблицей:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| y, м | 0 | 1 | 2 | 5 | 10 | 15 | 25 | 50 | 100 |
| T, C | 30 | 30,5 | 30,7 | 30 | 29 | 28 | 27,5 | 27 | 25 |

**Шаг 1: Установка необходимых библиотек**

Перед началом работы необходимо установить NuGet пакеты:

1. ScottPlot - для визуализации данных
2. ScottPlot.WinForms - компоненты для интеграции с Windows Forms
3. MathNet.Numerics - для математических вычислений и интерполяции

**Шаг 2: Подготовка главной формы (рисунок 11)**

На форме должны быть расположены следующие элементы:

* Элемент FormsPlot (из ScottPlot.WinForms) для отображения графика
* Кнопка loadDataButton для загрузки данных из файла

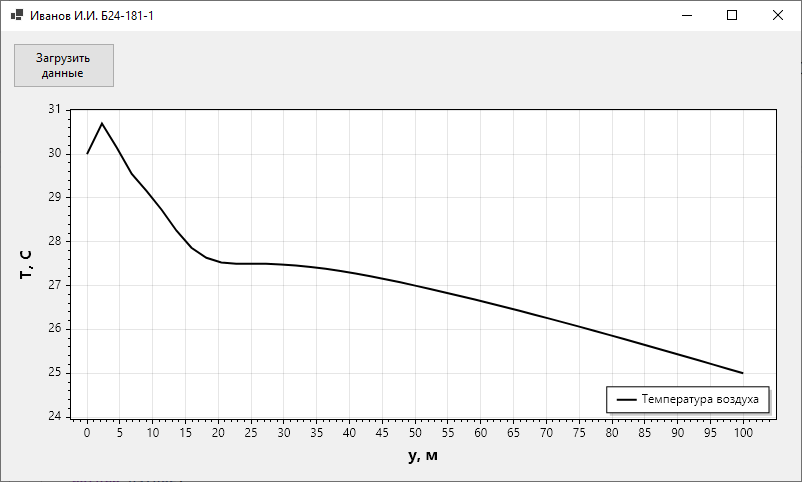


Рисунок 11 – Главное окно программы

**Шаг 3: Основной функционал формы (MainForm)**

Выделим отдельно методы для считывания данных, вычисления промежуточных значений и построения графика

1. Инициализация формы

* В методе MainForm\_Load устанавливаются подписи осей:
  + Ось X: "y, м" (высота в метрах)
  + Ось Y: "T, С" (температура в градусах Цельсия)
* Включается отображение легенды на графике

2. Загрузка данных

* При нажатии на кнопку loadDataButton открывается диалог выбора файла (поддерживаются форматы .txt и .csv)
* Файл должен содержать две строки:
  1. Первая строка - значения высот (в метрах)
  2. Вторая строка - соответствующие значения температур (в градусах Цельсия)
* Данные разделяются пробелами или табуляциями

3. Обработка данных

* Метод ParseLine разбивает строку на отдельные числовые значения
* Создается объект PlotData, содержащий пары значений высота-температура

4. Интерполяция данных

* Константа INTERPOLATION\_FACTOR = 5 определяет, во сколько раз увеличится количество точек после интерполяции
* Метод Interpolate использует натуральный кубический сплайн (CubicSpline.InterpolateNatural) для создания плавного графика
* Создаются новые массивы данных с увеличенным количеством точек

5. Визуализация данных

* Метод DrawTemperaturesScatter отображает интерполированные данные на графике:
  + Черная линия толщиной 2 пикселя
  + Без маркеров в точках данных
  + Подпись в легенде: "Температура воздуха"
* График автоматически масштабируется под данные

using MathNet.Numerics.Interpolation;

using ScottPlot;

namespace WinFormsApp2;

public partial class MainForm : Form

{

    private const int INTERPOLATION\_FACTOR = 5;

    public MainForm()

    {

        InitializeComponent();

    }

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

    {

        formsPlot.Plot.Axes.Bottom.Label.Text = "y, м";

        formsPlot.Plot.Axes.Left.Label.Text = "T, С";

        formsPlot.Plot.ShowLegend();

    }

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

    {

        var ofd = new OpenFileDialog

        {

            Filter = "Text Files|\*.txt;\*.csv"

        };

        if (ofd.ShowDialog() != DialogResult.OK)

            return;

        var lines = File.ReadAllLines(ofd.FileName);

        var heights = ParseLine(lines[0]);

        var temperatures = ParseLine(lines[1]);

        var input = new PlotData(heights, temperatures);

        var interpolated = Interpolate(input, input.X.Count \* INTERPOLATION\_FACTOR);

        DrawTemperaturesScatter(interpolated);

    }

    private PlotData Interpolate(PlotData data, int interpolatedPointsCount)

    {

        var spline = CubicSpline.InterpolateNatural(data.X, data.Y);

        var dx = (data.X.Last() - data.X.First()) / (interpolatedPointsCount - 1);

        var interpolatedX = new double[interpolatedPointsCount];

        var interpolatedY = new double[interpolatedPointsCount];

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

        {

            interpolatedX[i] = data.X.First() + i \* dx;

            interpolatedY[i] = spline.Interpolate(interpolatedX[i]);

        }

        return new PlotData(interpolatedX, interpolatedY);

    }

    private void DrawTemperaturesScatter(PlotData tempera-tures)

    {

        var scatter = formsPlot.Plot.Add.Scatter(temperatures.X.ToArray(), temperatures.Y.ToArray());

        scatter.LineColor = Colors.Black;

        scatter.MarkerShape = MarkerShape.None;

        scatter.LineWidth = 2;

        scatter.LegendText = "Температура воздуха";

        formsPlot.Plot.Axes.AutoScale();

        formsPlot.Refresh();

    }

    private double[] ParseLine(string line)

    {

        var strValues = line.Split(new[] { ' ', '\t' });

        var values = new double[strValues.Length];

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

        {

            values[i] = double.Parse(strValues[i]);

        }

        return values;

    }

}

**Шаг 4: Вспомогательный класс PlotData**

* Простая запись (record) для хранения пар значений X (высота) и Y (температура)
* Позволяет удобно передавать данные между методами

public record PlotData(ICollection<double> X, ICollection<double> Y);

**Контрольные вопросы**

1. Что такое ScottPlot и для чего он используется?
2. Как установить библиотеку ScottPlot в проект C#?
3. Как создать простой график с использованием ScottPlot?
4. Какой метод используется для отображения графика в WinForms?
5. Какие типы графиков поддерживает ScottPlot?
6. Как построить линейный график и чем он отличается от точечного графика?
7. Как изменить заголовок графика и подписи осей?
8. Как изменить цвет и стиль линий на графике?
9. Как обновить данные на уже существующем графике без его перерисовки?
10. Как добавить возможность масштабирования?
11. Как сохранить график в файл (например, в формате PNG или SVG)?
12. Какие методы доступны для экспорта графиков в другие форматы?
13. Как добавить несколько графиков на одну ось?
14. Как настроить отображение легенды для нескольких графиков?
15. Как реализовать анимацию графика, чтобы данные обновлялись в реальном времени?
16. Как изменить шрифт и размер текста на графике?
17. Как добавить аннотации или текстовые метки к определенным точкам на графике?
18. Как загрузить данные из файла (например, CSV) и построить график на их основе?