

СОВРЕМЕННЫЕ .NET-СОВМЕСТИМЫЕ СРЕДСТВА ПОСТРОЕНИЯ ДИАГРАММ

Н. А. Шульгин, студент гр. ИТ-101, V курс

Научный руководитель: Г. А. Алексеева, ст. преподаватель
Кузбасский государственный технический университет
имени Т. Ф. Горбачёва
г. Кемерово

Диаграмма является наглядным средством визуализации представления данных. Существует множество программных средств, позволяющих осуществлять построение диаграмм и графиков, однако, если речь идёт о создании собственного независимого приложения, встаёт вопрос о разработке или поиске программного модуля, осуществляющего работу с графикой.

В среде .NET в классическом оконном приложении Windows.Forms существует класс **Canvas**, позволяющий отображать графические примитивы на любом объекте пользовательского интерфейса Windows.Controls. Используя его, разработчик может создать собственное средство отображения диаграмм.

В быстроизменяющихся условиях средство, применяемое для отображения диаграмм, должно обеспечивать работу с динамическими данными. Для графического представления таких данных в обозначенных условиях разработки, указанное средство должно удовлетворять следующим критериям:

- совместимость с выбранной средой разработки – .NET, Windows.Forms (поддержка технологии web-приложений ASP.NET также будет плюсом);
- открытый код компонента или свободное распространение;
- высокое быстродействие – визуализация не должна задерживать действия пользователя, необходима возможность оперативно перестраивать диаграмму, без задержек осуществлять масштабирование и смещение по временной оси.

Рассмотрим наиболее быстродействующие [1] из свободно-распространяемых компонентов, сравнительная оценка которых представлена в таблице 1.

Таблица 1 – Сравнение компонентов построения диаграмм

Название	Требуется Framework	Проект развивается	Наличие документации	Размер библиотеки, КБ
DisLin	3.5	да	да	1 352
ZedGraph	2.0	да	да	300
NPlot	2.0	нет	нет	176

DisLin – кроссплатформенная библиотека (UNIX, Windows, FreeBSD и Mac), обладающая широчайшим функционалом (в том числе и для 3D построений). Разработана германским институтом исследований Солнечной системы имени М. Планка [2]. При программировании использован функциональный подход. Портирована на C, C++, C#, Fortran, Perl, Python, Ruby, Tgl, GCL. Бесплатна для некоммерческого использования.

Требует для работы Framework 3.5, ранние версии для скачивания не предлагаются.

На рис. 1 представлен пример использования модуля DisLin.

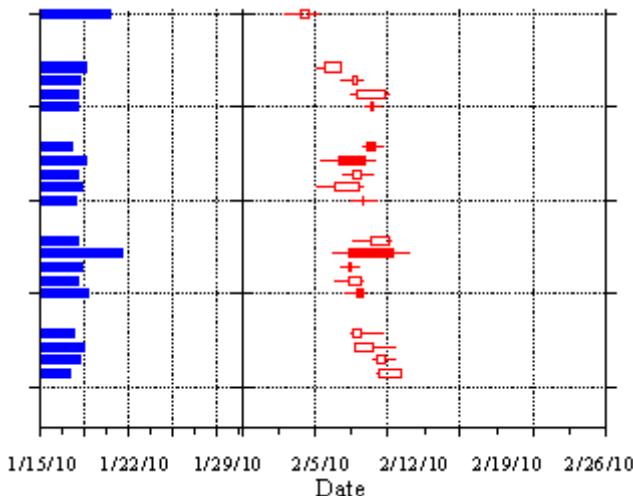


Рисунок 1 – DisLin: пример построения диаграммы

ZedGraph – открытая развивающаяся программная библиотека для платформы .NET (Windows.Forms и ASP.NET) для рисования на плоскости. Разработана на C# американскими программистами Крисом и Джоном Чемпионами, при участии ирландского программиста Рона О Салливана, с применением объектно-ориентированного подхода [3]. Существующие классы обеспечивают высокую степень гибкости – практически все настройки отображения могут быть изменены пользователем. Большинство значений имеют базовые опции по умолчанию, что позволяет уделять минимум внимания на конфигурирование компонента. Библиотека позволяет отображать точки, линии, круговые и столбчатые диаграммы.

На рис. 2 представлен пример использования модуля ZedGraph.

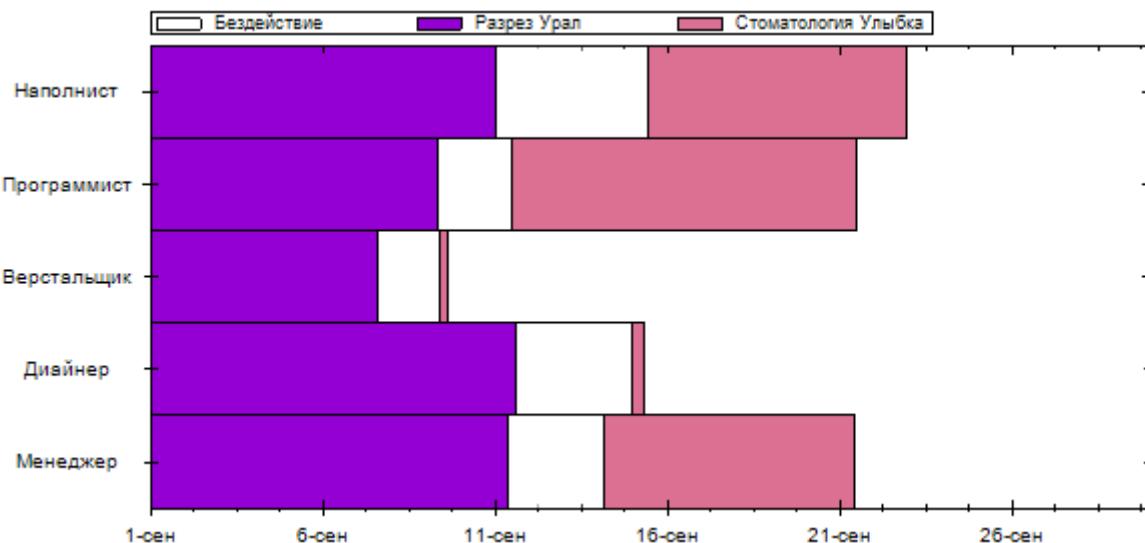


Рисунок 2 – ZedGraph: пример построения диаграммы

NPlot – открытая программная библиотека для платформы .NET (Windows.Forms и ASP.NET). Разработана британским программистом Мэттом Хоулеттом (и другими соискателями). Является незавершённым проектом (последняя версия 0.9.1 – 2007 г.) и в настоящее время не популярен. Не удалось обнаружить технической документации продукта, однако существуют англоязычные примеры по использованию с комментариями [4].

На рис. 3 представлен пример использования модуля NPlot.

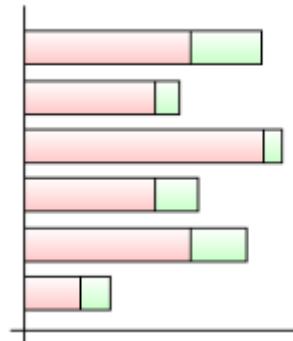


Рисунок 3 – NPlot: пример построения диаграммы

Также следует упомянуть о компоненте **Chart Controls** от Microsoft. Он имеет широкое распространение и хорошо документирован, в том числе и на русском языке, а также является бесплатным; однако не рассматривался в силу меньшего быстродействия [1]. Кроме того, отсутствует возможность скачать версию для Framework 2.0, несмотря на то, что расширенная поддержка этой версии платформы заканчивается только в 2016 году.

По результатам анализа для применения выбран компонент ZedGraph, как наиболее подходящий по функционалу, а также простой в применении; кроме того его использование увеличивает совместимость со старыми операционными системами, благодаря поддержке Framework 2.0.

Рассмотрим пример работы с ZedGraph в Microsoft Visual Studio. Для добавления компонента в проект следует в контекстном меню панели «Toolbox» выбрать пункт «Choose items...», в открывшемся окне нажать кнопку «Browse...» и в диалоговом окне выбрать файл динамической библиотеки «ZedGraph.dll». После чего можно получить доступ к компоненту, используя пространство имён ZedGraph.

Код для построения горизонтальной столбчатой диаграммы представлен на рис. 4.

```
1 // Начальная инициализация статически созданного объекта zedGC
2 GraphPane pane = zedGC.GraphPane;
3 pane.CurveList.Clear();
4 PointPairList list = new PointPairList();
5 pane.BarSettings.Base = BarBase.Y; // Задание ориентации: горизонтальной
6 pane.BarSettings.Type = BarType.Stack; // Задание вида: столбчатая
7 pane.XAxis.Type = AxisType.Date; // Задание типа шкалы оси оХ: дата
8 // Задание начала и конца координат оси оХ
9 pane.XAxis.Scale.Min = new XDate(2014, 09, 01);
10 pane.XAxis.Scale.Max = new XDate(2014, 09, 30);
11 pane.YAxis.Type = AxisType.Text; // Задание типа шкалы оси оУ: текст
12 pane.YAxis.Scale.TextLabels = new[] { "Менеджер", "Дизайнер", "Верстальщик",
13     "Программист", "Наполнитель" }; // Задание значений подписи оси
14 // Построение 5 столбцов случайной длины
15 int itemsCount = 5;
16 double[] barLength = new double[itemsCount];
17 double[] barPosition = new double[itemsCount];
18 Random rnd = new Random();
19 for (int i = 0; i < itemsCount; i++)
20 {
21     barLength[i] = (pane.XAxis.Scale.Max - pane.XAxis.Scale.Min) * 0.4
22         * rnd.NextDouble();
23     barPosition[i] = i + 1;
24 }
25 // Вызов метода добавления элемента столбцов
26 pane.AddBar("Урал", barLength, barPosition, Color.DarkViolet).Bar.Fill.Type
27     = FillType.Solid;
28 // Обновление графика
29 zedGC.AxisChange();
30 zedGC.Invalidate();
```

Рисунок 4 – Листинг кода построения диаграммы при помощи ZedGraph

Результатом построения будет диаграмма аналогичная представленной на рис. 3.

Благодаря открытой архитектуре [6] модуль ZedGraph может быть доработан под нужды конкретного приложения. Например, для внесения функционала обратной связи отдельных элементов диаграммы.

Список литературы

1. Средства построения графиков для .NET // Хабрахабр [Электронный ресурс]. Дата обновления: 03.12.2013. – URL: <http://habrahabr.ru/post/204308/> (дата обращения: 28.07.2014).
2. Dislin // Max Planck Institute for Solar System Research [Электронный ресурс] Дата обновления: 21.07.2014. – URL: <http://www.mps.mpg.de/dislin/> (дата обращения: 08.09.2014).
3. ZedGraph // SourceForge [Электронный ресурс]. Дата обновления: 19.09.2012. – URL: <http://sourceforge.net/projects/zedgraph/> (дата обращения: 30.03.2015).
4. NPlot Wiki // Netcontrols [Электронный ресурс]. Дата обновления: 08.08.2007. – URL: <http://netcontrols.org/nplot/wiki/> (дата обращения: 28.07.2014).
5. Сроки поддержки продуктов Майкрософт // Microsoft [Электронный ресурс]. Дата обновления: 16.07.2014. – URL: <https://support.microsoft.com/ru-ru/lifecycle/search?sort=PN&alpha=.NET%20Framework> (дата обращения: 28.07.2014).
6. ZedGraph Class Library Documentation // SourceForge – 2007 [Электронный ресурс]. – URL: <http://zedgraph.sourceforge.net/documentation/default.html> (дата обращения: 30.03.2015).
7. Шпаргалка по ZedGraph // Сайт Евгения Ильина [Электронный ресурс] Дата обновления: 24.01.2013. – URL: <http://jenyay.net/Programming/ZedGraph> (дата обращения: 12.09.2014).