МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное автономное бюджетное образовательное учреждение высшего образования

Севастопольский государственный университет

кафедра Информационных систем

**Лисянский Александр Игоревич**

Институт информационных технологий и управления в технических системах

курс 4 группа ИC/б-42(о)

09.03.02 Информационные системы (уровень бакалавриата)

КУРСОВОЙ ПРОЕКТ

по дисциплине «Интеллектуальный анализ данных»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Листов 20

Отметка о зачёте \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

Руководитель практикума

асс. Токарев А. И.

(должность) (подпись) (инициалы, фамилия)

Севастополь 2015

АННОТАЦИЯ

В документе описана поэтапная разработка приложения автоматического анализа временных рядов для получения достоверных прогнозов. В работе использовались модели AP, MA первых и вторых порядков и RWwD.

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 4](#_Toc436813926)

[1. ПОСТАНОВКА ЗАДАЧИ 5](#_Toc436813927)

[1.1 Техническое задание 5](#_Toc436813928)

[2. АНАЛИЗ И ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 6](#_Toc436813929)

[3. Временные ряды 7](#_Toc436813930)

[4. ВЫБОР ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ 10](#_Toc436813931)

[5. РАЗРАБОТКА И ТЕСТИРОВАНИЕ ПРОГРАММЫ 11](#_Toc436813932)

[5.1 Проектирование интерфейса 11](#_Toc436813933)

[5.2 Тестирование программы 11](#_Toc436813934)

[ЗАКЛЮЧЕНИЕ 14](#_Toc436813935)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 15](#_Toc436813936)

[ПРИЛОЖЕНИЕ А 16](#_Toc436813937)

Изм.

Лист

№ докум.

Подпись

Дата

Лист

3

КУРСОВОЙ ПРОЕКТ

Разраб.

Криворучко Ю. Э.

Провер.

*Токарев А.И.*

Н. Контр.

Утв.

ПОЯСНИТЕЛЬНАЯ

ЗАПИСКА

Лит.

Листов

20

Группа ИC/б-42(о)

# ВВЕДЕНИЕ

В настоящее время существует множество современных и точных программ по построению прогнозов временных рядов. Зачастую эти программы являются коммерческим ПО и стоимость лицензии оставляет желать лучшего. Целью данного курсового проекта является написание несложной программы прогнозирования временных рядов на основании знаний, полученных ранее на той же дисциплине.

Прогнозирование предполагает построение графика зависимости одной величины от другой при помои вспомогательных коэффициентов. На основании прогнозируемых значений вычисляются допустимые интервалы.

При помощи прогнозирования аналитики выводят зависимости курсов валют, прогнозов погоды, котировок акций на бирже и т.д. Прогнозирование – мощный инструмент для получения новых сведений в узкой области для достижения своих целей и решения поставленных задач.

1. ПОСТАНОВКА ЗАДАЧИ
   1. Техническое задание

На курсовой проект было выдано задание в виде анализа временного ряда тремя различными моделями. В качестве варианта задания выбраны модели:

* AP();
* MA();
* RWwD.

На основании варианта был спроектирован следующий программный продукт с возможностями:

* Построение модели AP(0) и AP(1) на максимум 20 шагов;
* Построение модели MA(0) и MA(1) на максимум 20 шагов;
* Построение модели RWwD на максимум 20 шагов;
* Запись в файл полученного прогноза;
* Чтение из файла входного временного ряда.

1. АНАЛИЗ И ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

В качестве предметной области временного ряда был выбран ряд объемы вылова рыбной продукции за январь 2007 – сентябрь 2009. При расчетах не учитывался фактор сезонности

1. Временные ряды

Одним из методов анализа временных рядов являются модели авторегрессии и скользящего среднего, которые оказываются особенно полезными для описания и прогнозирования процессов, проявляющих однородные колебания вокруг среднего значения.

Модель AP

Для того чтобы понять смысл термина "авторегрессия", стоит провести параллель с другими подобными словами (автопортрет, автобиография), в которых первая часть слова является указанием на первое лицо. Таким образом, авторегрессия - это построение такого уравнения регрессии, в котором функциональная зависимость проявляется между двумя зависимыми величинами, а не между одной зависимой и одной независимой величинами, как в линейной регрессии. Иначе говоря, обычная линейная регрессия применительно к спекулятивному рынку - это зависимость цен от времени, а авторегрессия - это зависимость текущих цен от цен, зафиксированных в предыдущие периоды.

При таком подходе шкала времени, присутствующая на свечном представлении графика цен, оказывается не нужной, а период времени между соседними значениями цен может быть переменным. Отсюда вытекает еще одно следствие, связанное с прогнозированием. Так, если прогноз, составленный на основании уравнения линейной регрессии, распространяется на конкретное будущее время, то при использовании авторегрессии прогноз составляется для будущего значения цены, время появления которого не оговаривается.

Простейшее уравнение авторегрессии называют уравнением первого порядка. Оно является точной копией уравнения линейной регрессии (формула (1) [первой части статьи](http://www.admiralmarkets.ru/mqlabs/07.10.2013-mqlabs-regressionnyy-analiz)), которое здесь мы запишем с использованием других обозначений:

(1),

В общем виде решение нахождения коэффициентов авторегрессии выглядит так для уравнений первого и второго порядка

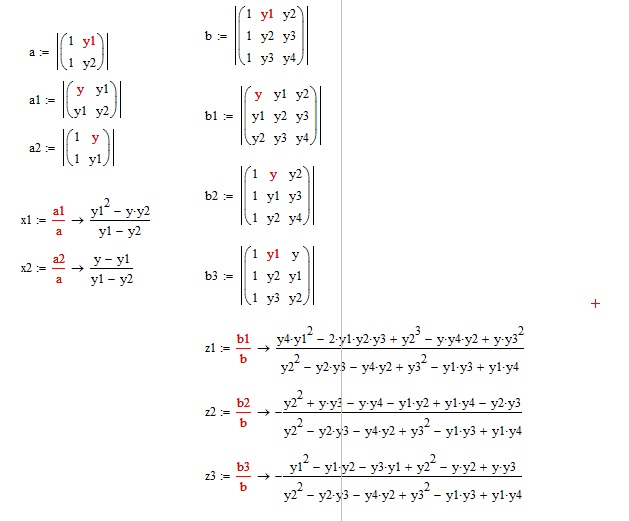


Рисунок 3.1 – Нахождение коэффициентов авторегрессии

Модель MA

В модели скользящего среднего в отличие от предыдущего способа предполагают, что каждый элемент ряда подвержен суммарному воздействию случайных предыдущих ошибок :

, (2),

где  – значение  в момент времени ;  – коэффициенты уравнения (*i*=1,2,…,q); *q* – порядок модели скользящего среднего; – случайная величина.

Модель RWwD

Модель случайного блуждания основана на предыдущем значении, значении дрейфа (сдвига) и на случайной составляющей. В общем случае формула модели выглядит так:

(3),

где – сдвиг или дрейф, – случайная составляющая.

На основании этого можно предполагать, что модель зависит от случайной составляющей больше чем две вышеописанные модели.

1. ВЫБОР ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ

В качестве платформы разработки был выбран язык C#. Он обладает мощным потенциалом, имеет множество встроенных сложных типов данных, хорошо адаптирован под написание оконных приложений, не требует очистки памяти

1. РАЗРАБОТКА И ТЕСТИРОВАНИЕ ПРОГРАММЫ
   1. Проектирование интерфейса

Для данного курсового проекта был разработан внешний вид оконного приложения, для удобного использования пользователем и простого понимания результатов прогнозирования. В качестве подложки для отображения был выбран объект ImageBox. У объекта есть встроенные методы рисования и простой интерфейс.

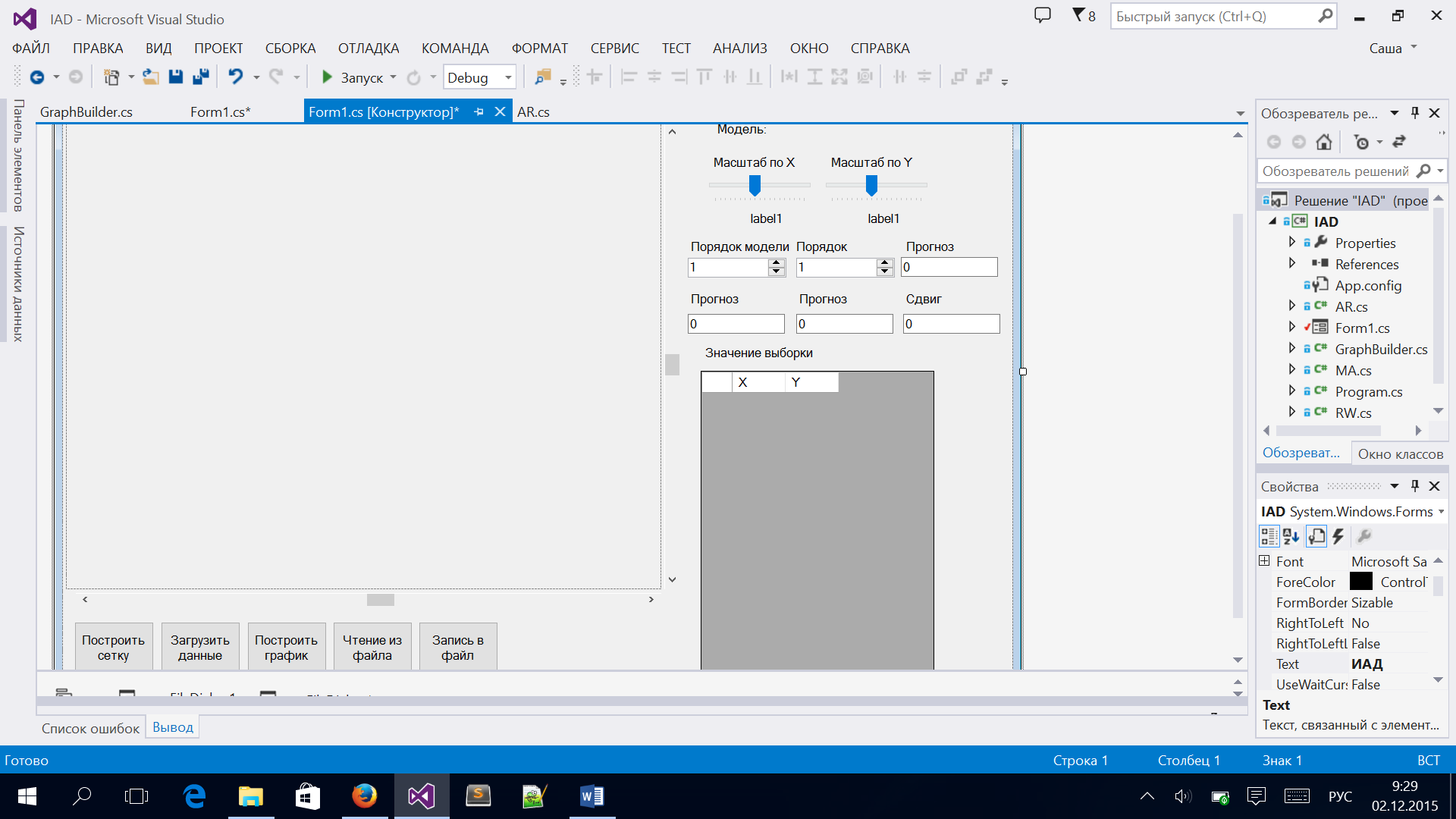


Рисунок 5.1 – Интерфейс программы

Было разработано меню пользователя для выбора модели прогнозирования.

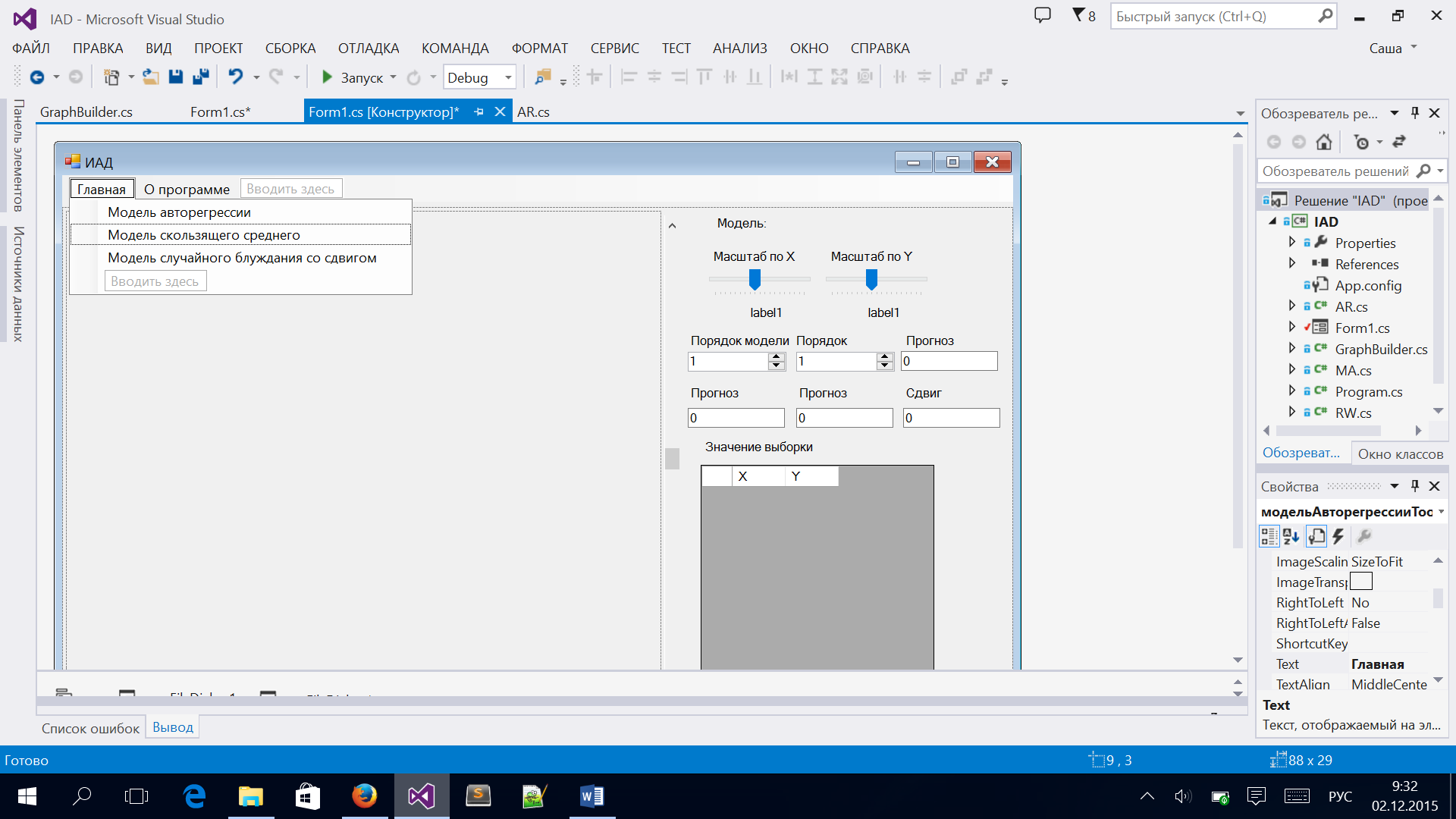


Рисунок 5.2 – Меню пользователя

В дальнейшем был разработан интерфейс загрузки и сохранения данных, вывода прогноза и модели на экран и ввод пользовательских данных. Это было опущено для экономии размера данной пояснительной записки.

* 1. Тестирование программы

При запуске программы пользователю доступны всего несколько элементов управления. При выборе модели доступные элементы управления сменяют друг друга в зависимости от выбранной модели.

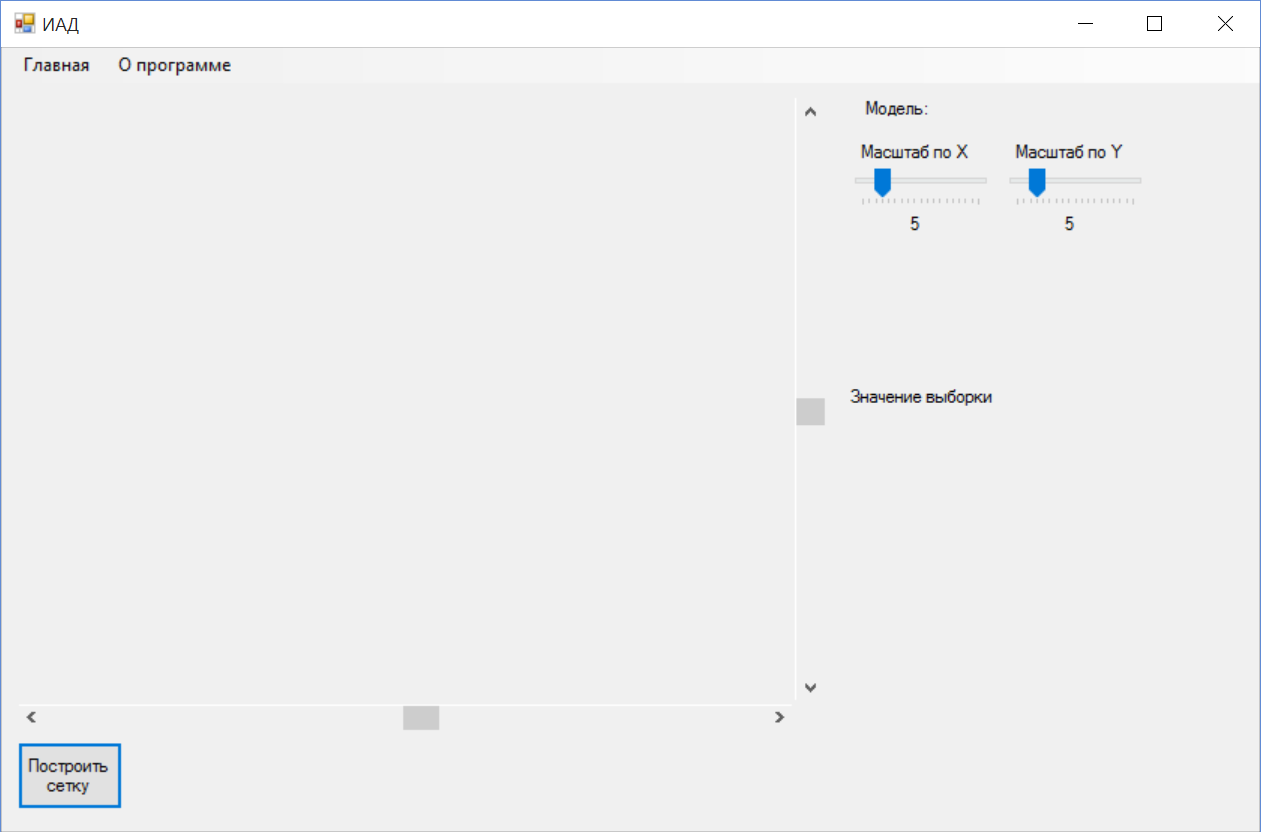


Рисунок 5.3 – Начальное состояние программы

После загрузки программы пользователь может выбрать модель коикнув по необходимому пункту меню. Например, выберем модель скользящего среднего.

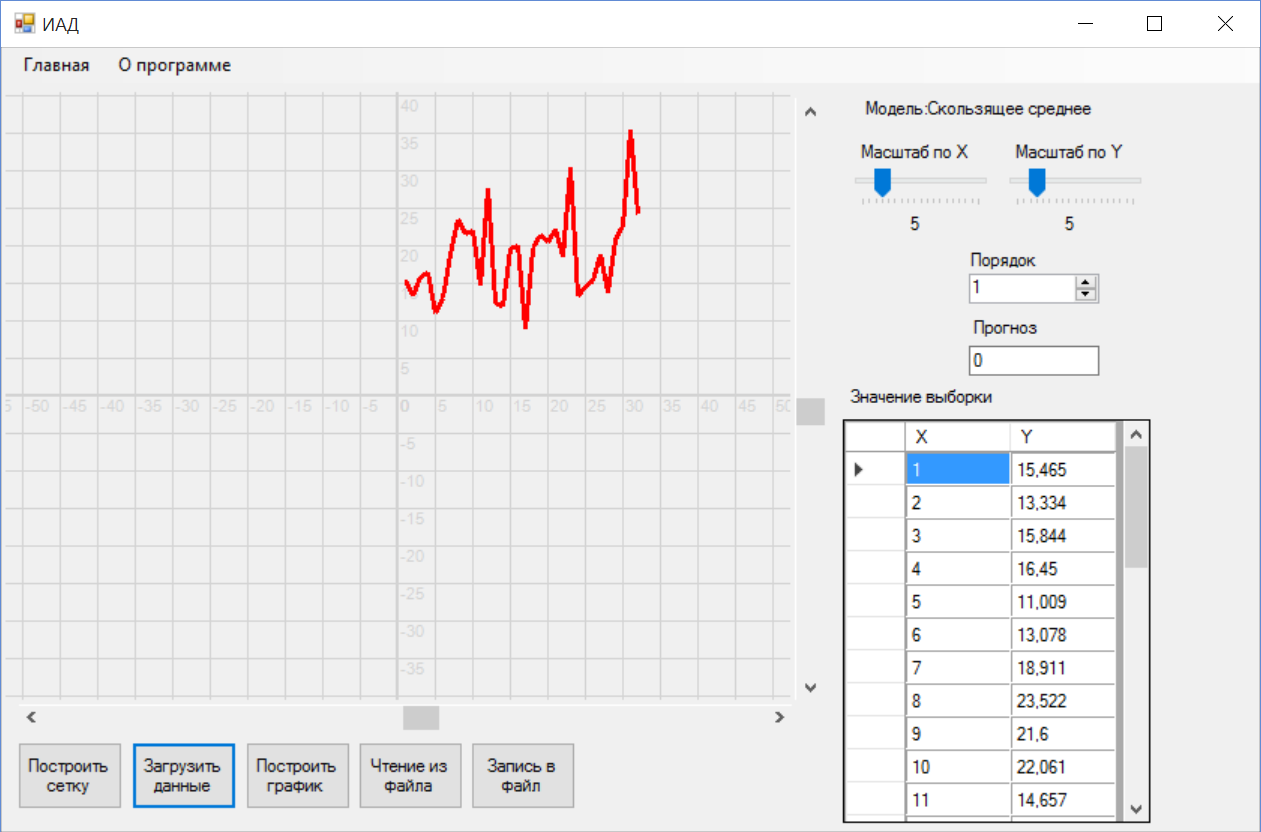


Рисунок 5.4 – Загрузка интерфейса модели скользящего среднего.

После выбора модели и загрузки данных можно приступать к прогнозированию. Возьмем для примера модель MA(1).

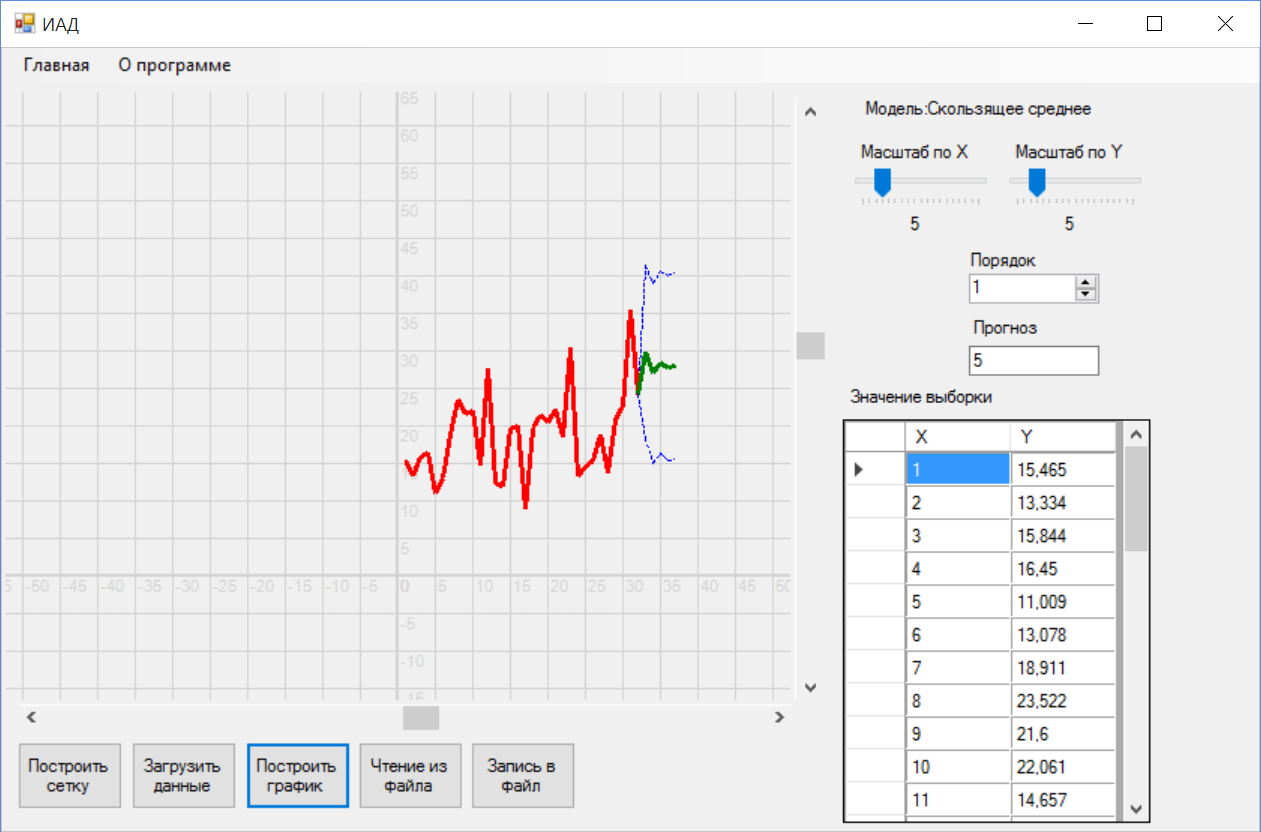


Рисунок 5.5 – Прогноз скользящего среднего.

Далее пользователь в праве записать прогноз в файл или же загрузить новые данные из файла. На примере модели скользящего среднего была продемонстрирована работа программы. Тестирование показало, что программа работает корректно, сбоев нет.

# ЗАКЛЮЧЕНИЕ

В данном курсовом проекте была реализована программа построения прогноза по нескольким выбранным моделям прогнозирования. В частности, были выбраны модели AP, MA и RWwD. Для каждой из моделей был написан и отлажен свой интерфейс и функционал, по которому работают модели. Проверка работы показала, что программа написана правильно, претензий нет.

Программа была разработана при помощи мощного инструментального средства Microsoft Visual Studio на платформе .NET. Основным языком программирования был выбран C#.

Все цели курсового проектирования были достигнуты. В результате выполнения курсового проектирования были закреплены навыки построения прогнозов временных рядов, написания функционала прогнозирования.

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Методические указания к выполнению курсовой работы по дисциплине «ИАД» для студентов направления 09.03.02 «Информационные системы и технологии» ./ Е.Л. Первухина, А.И. Токарев. — Севастополь: Изд-во СевГУ, 2015. — 19 с.
2. Регрессионный анализ. Часть 3. — Режим доступа: http://www.admiralmarkets.ru/mqlabs/27.10.2013-mqlabs-regressionnyy-analiz-chast-3 . — Последний доступ: 25.10.2015. — Название с экрана.

# ПРИЛОЖЕНИЕ А

Код модели AP

using System;

using System.Collections.Generic;

using System.Linq;

using System.Drawing;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace IAD

{

class AR

{

public float coefA;

public float coefB;

public float coefC;

private float[] determinant2x2(PointF y, PointF y1, PointF y2)

{

float[] result = new float[2];

float det1 = (y.Y - y1.Y) / (y1.Y - y2.Y);

float det2 = (y1.Y \* y1.Y - y.Y \* y2.Y) / (y1.Y - y2.Y);

result[0] = det1;

result[1] = det2;

return result;

}

private float[] determinant3x3(PointF y, PointF y1, PointF y2, PointF y3, PointF y4)

{

float[] result = new float[3];

float det1 = (y4.Y \* y.Y \* y.Y - 2 \* y1.Y \* y2.Y \* y3.Y + y2.Y \* y2.Y \* y2.Y - y.Y \* y2.Y \* y4.Y + y.Y \* y3.Y \* y3.Y) /

(y2.Y \* y2.Y - y2.Y \* y3.Y - y4.Y \* y2.Y + y3.Y \* y3.Y - y1.Y \* y3.Y + y1.Y \* y4.Y);

float det2 = (y2.Y \* y2.Y + y.Y \* y3.Y - y.Y \* y4.Y - y1.Y \* y2.Y + y1.Y \* y4.Y - y2.Y \* y3.Y) /

(y2.Y \* y2.Y - y2.Y \* y3.Y - y4.Y \* y2.Y + y3.Y \* y3.Y - y1.Y \* y3.Y + y1.Y \* y4.Y);

float det3 = (y1.Y \* y1.Y - y1.Y \* y2.Y - y1.Y \* y3.Y + y2.Y \* y2.Y - y.Y \* y2.Y + y.Y \* y3.Y) /

(y2.Y \* y2.Y - y2.Y \* y3.Y - y4.Y \* y2.Y + y3.Y \* y3.Y - y1.Y \* y3.Y + y1.Y \* y4.Y);

result[0] = det1;

result[1] = det2;

result[2] = det3;

return result;

}

public List<PointF> Forecast(List<PointF> inmass, int steps, int exponent)

{

if (steps > 0)

{

List<PointF> result = new List<PointF>(steps + 1);

List<PointF> temp = new List<PointF>(inmass.Count + steps);

result.Add(new PointF(inmass[inmass.Count - 1].X, inmass[inmass.Count - 1].Y));

float step = 0;

for (int i = 1; i < inmass.Count; i++)

step += inmass[i].X - inmass[i - 1].X;

step /= inmass.Count - 1;

if (exponent == 1)

{

float[] res = determinant2x2(result[0], inmass[inmass.Count - 2], inmass[inmass.Count - 3]);

coefA = res[0];

coefB = res[1];

result.Add(new PointF(result[0].X + step, coefB + coefA \* result[0].Y));

res = determinant2x2(result[1], result[0], inmass[inmass.Count - 2]);

coefA = res[0];

coefB = res[1];

result.Add(new PointF(result[1].X + step, coefB + coefA \* result[1].Y));

for (int i = 2; i <= steps; i++)

{

res = determinant2x2(result[i], result[i - 1], result[i - 2]);

coefA = res[0];

coefB = res[1];

result.Add(new PointF(result[i].X + step,

coefB + coefA \* result[i].Y));

}

return result;

}

if (exponent == 2 && inmass.Count >= 5)

{

float[] res = determinant3x3(result[0], inmass[inmass.Count - 2], inmass[inmass.Count - 3], inmass[inmass.Count - 4], inmass[inmass.Count - 5]);

coefA = res[0];

coefB = res[1];

coefC = res[2];

result.Add(new PointF(result[0].X + step, coefA + coefB \* result[0].Y + coefC \* inmass[inmass.Count - 2].Y));

res = determinant3x3(result[1], result[0], inmass[inmass.Count - 2], inmass[inmass.Count - 3], inmass[inmass.Count - 4]);

coefA = res[0];

coefB = res[1];

coefC = res[2];

result.Add(new PointF(result[1].X + step, coefA + coefB \* result[1].Y + coefC \* result[0].Y));

res = determinant3x3(result[2], result[1], result[1], inmass[inmass.Count - 2], inmass[inmass.Count - 3]);

coefA = res[0];

coefB = res[1];

coefC = res[2];

result.Add(new PointF(result[2].X + step, coefA + coefB \* result[2].Y + coefC \* result[1].Y));

res = determinant3x3(result[3], result[2], result[1], result[1], inmass[inmass.Count - 2]);

coefA = res[0];

coefB = res[1];

coefC = res[2];

result.Add(new PointF(result[3].X + step, coefA + coefB \* result[3].Y + coefC \* result[2].Y));

for (int i = 4; i <= steps; i++)

{

res = determinant3x3(result[i], result[i - 1], result[i - 2], result[i - 3], result[i - 4]);

coefA = res[0];

coefB = res[1];

coefC = res[2];

result.Add(new PointF(result[i].X + step, coefA + coefB \* result[i].Y + coefC \* result[i - 1].Y));

}

return result;

}

}

return inmass;

}

public List<PointF> Model(List<PointF> inmass, int exponent)

{

List<PointF> result = new List<PointF>(inmass.Count);

result.Add(new PointF(inmass[0].X, inmass[0].Y));

if (exponent == 1)

{

float[] res = determinant2x2(inmass[2], inmass[1], inmass[0]);

coefA = res[0];

coefB = res[1];

result.Add(new PointF(inmass[1].X, coefB + coefA \* inmass[0].Y));

for (int i = 2; i < inmass.Count; i++)

{

res = determinant2x2(inmass[i], inmass[i - 1], inmass[i - 2]);

coefA = res[0];

coefB = res[1];

result.Add(new PointF(inmass[i].X, coefB + coefA \* inmass[i - 1].Y));

}

return result;

}

if (exponent == 2)

{

result.Add(new PointF(inmass[1].X, inmass[1].Y));

result.Add(new PointF(inmass[2].X, inmass[2].Y));

result.Add(new PointF(inmass[3].X, inmass[3].Y));

float[] res = new float[3];

for (int i = 4; i < inmass.Count; i++)

{

res = determinant3x3(inmass[i], inmass[i-1], inmass[i-2], inmass[i-3], inmass[i-4]);

coefA = res[0];

coefB = res[1];

result.Add(new PointF(inmass[i].X, coefA + coefB \* inmass[i - 1].Y + coefC \* inmass[i - 2].Y));

}

return result;

}

return inmass;

}

}

}

Код модели MA

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace IAD

{

public class MA

{

public List<PointF> Solve(List<PointF> inmass, int q, int steps)

{

List<PointF> result = new List<PointF>(inmass.Count + steps);

List<PointF> res = new List<PointF>(steps + 1);

res.Add(new PointF(inmass[inmass.Count - 1].X, inmass[inmass.Count - 1].Y));

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

{

result.Add(inmass[i]);

}

float step = 0;

for (int i = 1; i < inmass.Count; i++)

step += inmass[i].X - inmass[i - 1].X;

step /= inmass.Count - 1;

for (int i = inmass.Count; i < inmass.Count + steps; i++)

{

List<float> sum = new List<float>();

for (int j = 0; j <= q; j++)

{

sum.Add(result[i - j - 1].Y);

}

result.Add(new PointF(result[i - 1].X + step,sum.Average()));

}

for (int i = 1; i < steps + 1; i++)

{

res.Add(new PointF(res[i - 1].X + step, result[i + inmass.Count - 1].Y));

}

return res;

}

}

}

Код модели RWwD

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace IAD

{

class RW

{

public List<PointF> forecast(List<PointF> inmass, int steps, float alfa)

{

if (steps > 0)

{

List<PointF> result = new List<PointF>(steps + 1);

result.Add(new PointF(inmass[inmass.Count - 1].X, inmass[inmass.Count - 1].Y));

float step = 0;

for (int i = 1; i < inmass.Count; i++)

step += inmass[i].X - inmass[i - 1].X;

step /= inmass.Count - 1;

Random rand = new Random();

for (int i = 1; i <= steps; i++)

{

if (alfa != 0)

result.Add(new PointF(result[i - 1].X + step,

result[i - 1].Y + alfa + 2 \* (float)rand.NextDouble() - 1));

else

result.Add(new PointF(result[i - 1].X + step,

result[i - 1].Y + 2 \* (float)rand.NextDouble() - 1));

}

return result;

}

return inmass;

}

}

}