**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение   
высшего образования

«Национальный исследовательский Томский политехнический университет»

|  |  |
| --- | --- |
| Школа | Инженерная школа информационных технологий и робототехники |
| Обеспечивающее подразделение | Отделение информационных технологий |
| Направление подготовки | 09.03.02 Информационные системы и технологии |
| Образовательная программа | Информационные системы и технологии в бизнесе и промышленности |
| Специализация | Информационные системы и технологии в бизнесе |

**ОТЧЕТ**

**ПО КУРСОВОМУ ПРОЕКТУ**

|  |  |
| --- | --- |
| Дисциплина | Архитектура информационных систем |

|  |  |
| --- | --- |
| Выполнил обучающийся | Санжиев Даши Баирович |
| Группа | 8И8А |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись обучающегося)

Дата сдачи работы«\_\_\_»\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

Отчет принял \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Степень, звание, должность) (Ф.И.О.)

Текущий контроль в семестре

Баллы (из 40) \_\_\_\_\_\_\_\_\_\_\_

Дата защиты \_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ 20\_\_г.

Баллы (из 60) \_\_\_\_\_\_\_\_\_\_\_

Итоговые баллы (из 100)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись \_\_\_\_\_\_\_\_\_\_\_

Томск 2020

**Индивидуальное задание на курсовой проект**

Таблица 1 – Перечень работ (заданий), подлежащих выполнению

|  |
| --- |
| 1. Выбор ресурса для сбора информации. |
| 2. Анализ ресурса на предмет пригодности использования для курсовой работы. |
| 3. Разработка концептуальной и логической модели данных. |
| 4. Разработка физической модели базы данных и ее создание. |
| 5. Разработка программного компонента заполнения базы данных. |
| 6. Разработка механизма работы с данными: просмотр, редактирование, добавление, удаление. |
| 7. Разработка шаблонов. |
| 8. Разработка механизма генерации графиков на основе данных с ресурса. |
| 9. Генерация отчетов на основе выбранных данных. |

**Содержание**

[1 Введение 4](#_Toc60081127)

[2 Основная часть 5](#_Toc60081128)

[2.1 Разработка концептуальной и логической модели данных 5](#_Toc60081129)

[2.2 Разработка физической модели базы данных и ее создание 5](#_Toc60081130)

[2.3 Разработка программного компонента заполнения базы данных 6](#_Toc60081131)

[2.4 Разработка механизма работы с данными: просмотр, редактирование, добавление, удаление 6](#_Toc60081132)

[2.5 Разработка шаблонов 8](#_Toc60081133)

[2.6 Разработка механизма генерации графиков на основе данных с ресурса 9](#_Toc60081134)

[2.7 Генерация отчетов на основе выбранных данных 10](#_Toc60081135)

[3 Заключение 11](#_Toc60081136)

[Приложение А. Компонент наполнения и взаимодействия с базой данных 13](#_Toc60081137)

[Приложение Б. Компонент генерации отчетов и построения графика 24](#_Toc60081138)

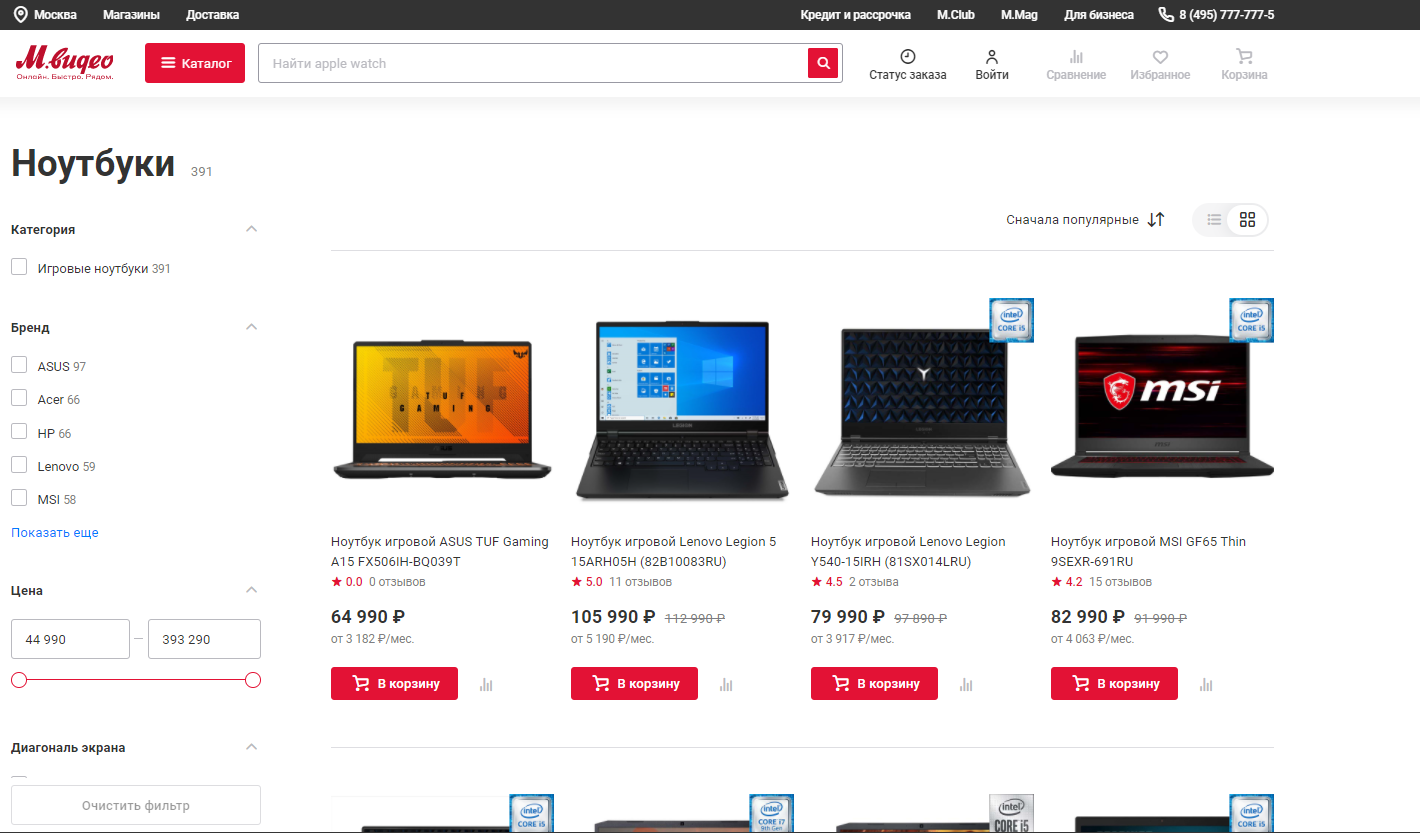
# 1 Введение

**Цель курсового проекта:** создание системы, которая будет считывать информацию с выбранного информационного ресурса, заносить в базу данных и генерировать отчеты и диаграммы на основе выбранной информации.

**Область применения:** обновление данных для поддержки актуальности, сбор и дублирование информации для генерации отчетов, обработка и распределение нужной информации.

Для выполнения курсовой работы был выбран сайт «mvideo.ru».

1. Была выбрана тематика данных, которая включает в себя разбор игровых ноутбуков. Данная тематика может служить в качестве сравнения характеристик ноутбука, что очень удобно при покупке (рисунок 1).



*Рисунок 1 – Информационный ресурс mvideo.ru*

1. Ресурс, содержит связанные данные, которые можно соединить в таблицах SQL.
2. Данный сайт имеет более 100 экземпляров, что тоже соответствует критериям курсовой работы.
3. Есть все возможности использовать парсинг, никаких проблем, блокировок не было обнаружено.

# 2 Основная часть

## **2.1 Разработка концептуальной и логической модели данных**

Используется нотация ER-модель Information Engineering (рисунки 2-3).

1. Было создано 5 таблиц:

* Model – соответствует сущности моделей игровых ноутбуков, хранит в себе основные данные ноутбука: ссылку на страницу, название модели, цену, количество бонусов, средний рейтинг, ОЗУ, информацию о SSD, диагональ экрана.
* Firm – соответствует сущности фирмы (используется как таблица – справочник).
* CPU – соответствует сущности процессора, хранит в себе информацию о названии, количестве ядер, средней тактовой частоте.
* Reviews – соответствует сущности отзывов, хранит в себе оценку и сам отзыв.
* GraphicCard – соответствует сущности видеокарт (используется как таблица – справочник).
* Все таблицы связаны связью «один ко многим».

1. Ввиду того, что информация о ноутбуках на сайте не статична (каждый ноутбук расписан по – своему, т.е у каждого ноутбука различное количество информации о характеристиках), база данных сделана наиболее «широко».



Рисунок 2 – Концептуальная модель

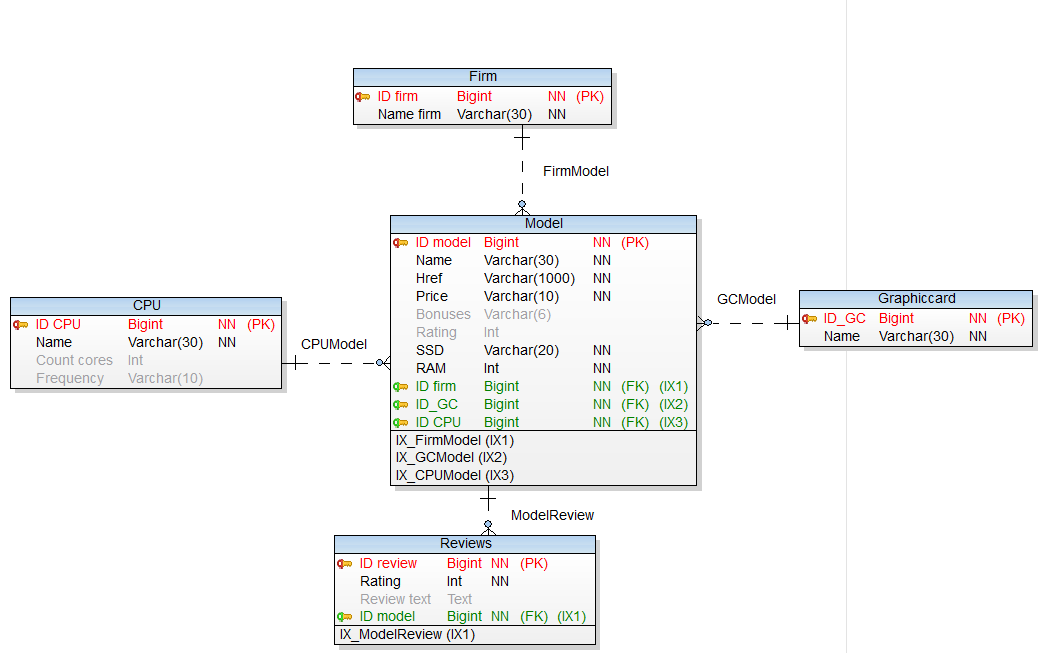


Рисунок 3 – Логическая модель

## **2.2 Разработка физической модели базы данных и ее создание**

В результате переноса схемы из TDM в MSSQL была получена следующая физическая модель (рисунок 4).

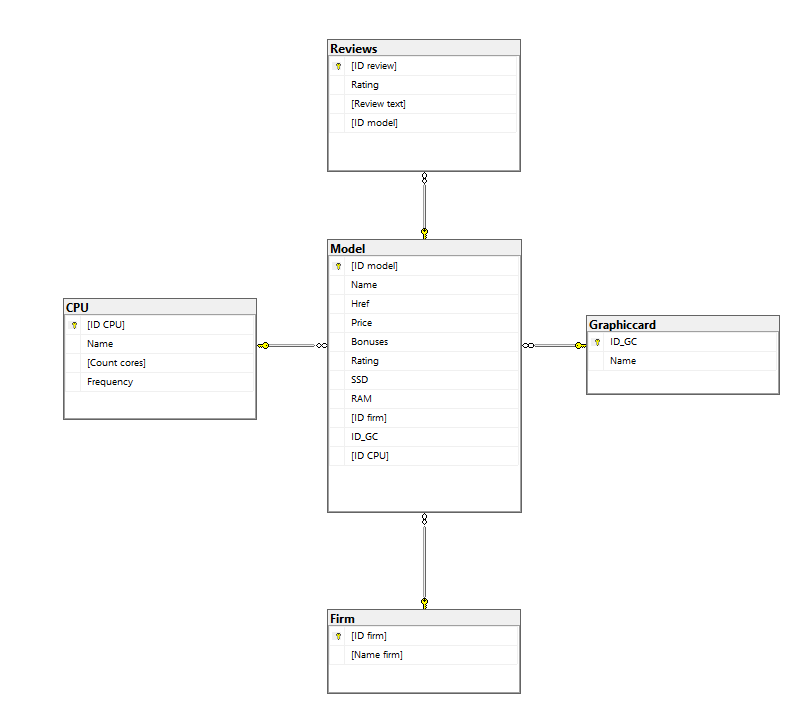


Рисунок 4 – Физическая модель

## **2.3 Разработка программного компонента заполнения базы данных**

Сначала пользователь выбирает страницу, с которой считывается информация. После данная ссылка поступает в метод, где считываются все ссылки на соответствующие ноутбуки. Далее в цикле каждая ссылка передается в метод, которая считывает всю информацию и добавляет запись в базу данных. Вспомогательная информация взята из 1 источника.

## **2.4 Разработка механизма работы с данными: просмотр, редактирование, добавление, удаление**

Далее показан графический интерфейс программы (рисунок 5-17).

Весь код реализованного функционала представлен в Приложении А. Вспомогательная информация взята из 2 и 3 источника.

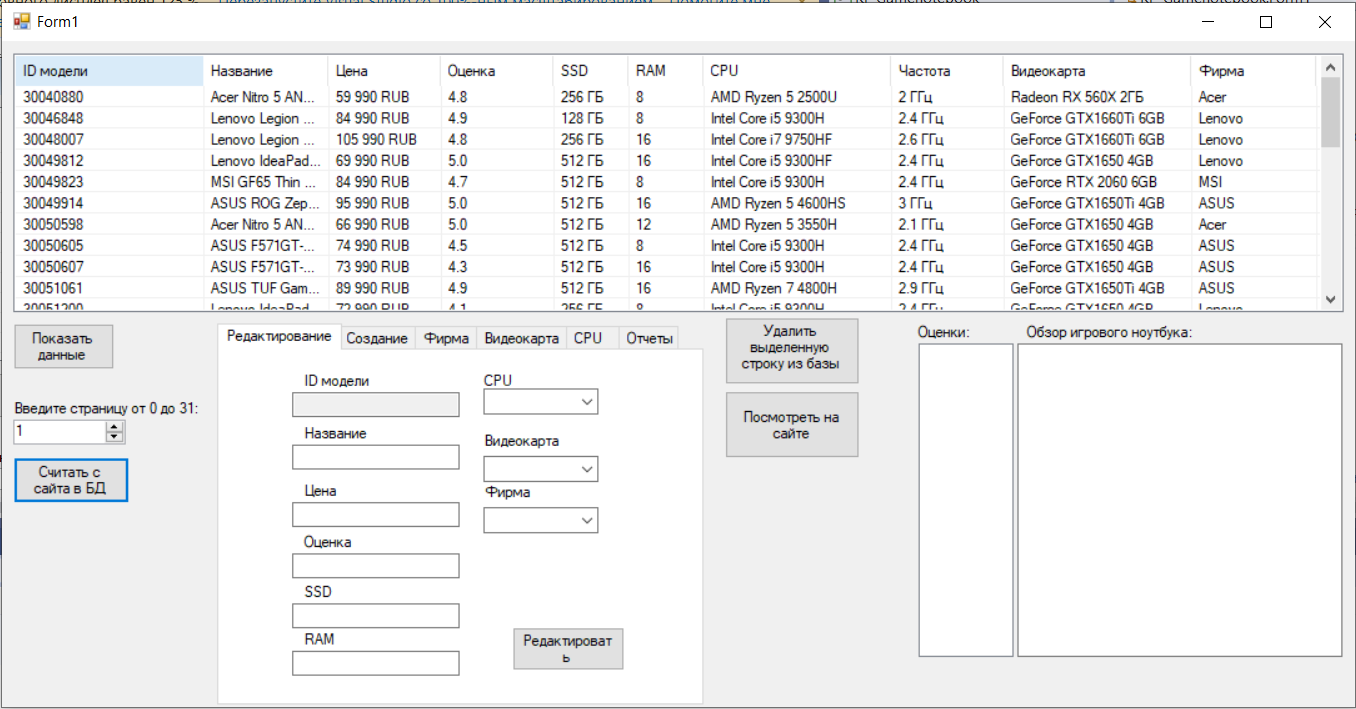


Рисунок 5 – Графический интерфейс

При нажатии на кнопку «Считать с сайта в БД» происходит вызов парсера, который наполняет БД (дубликаты не попадают).

При нажатии кнопки «Показать данные» данные из БД загружаются в Listview с которым дальше можно взаимодействовать (рисунок 6).

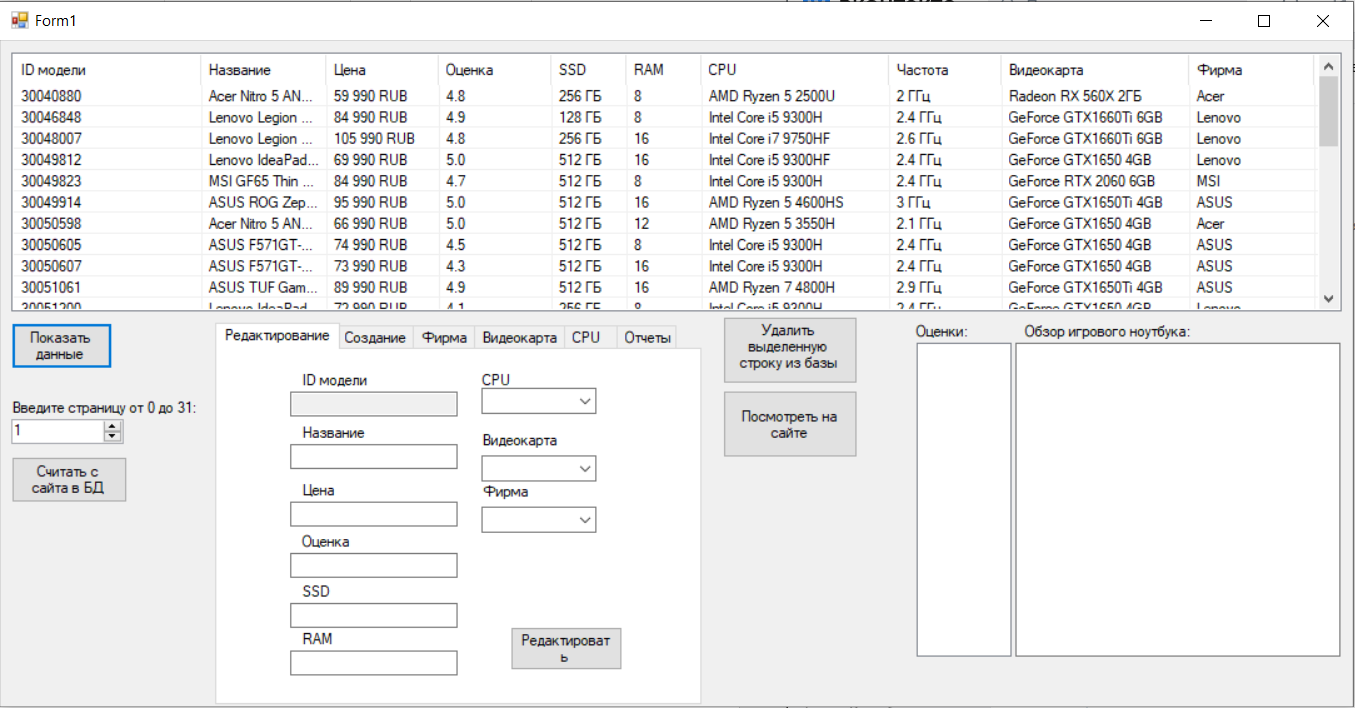


Рисунок 6 – Нажатие на «Показать данные»

После выбора какого-нибудь элемента, можно посмотреть оценки и отзывы (рисунок 7).

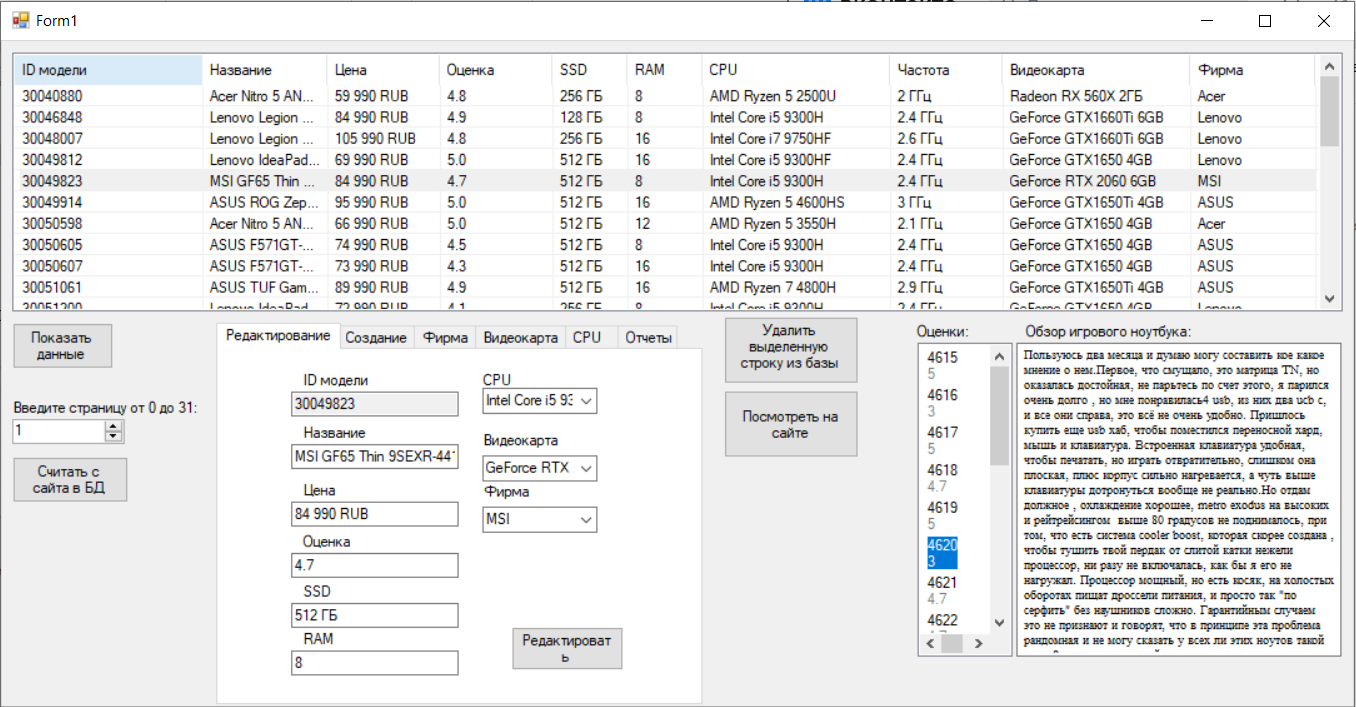


Рисунок 7 – Просмотр оценок и обзоров

При нажатии на кнопку «Удалить выделенную строку из базы» выбранная запись удаляется из БД.

При нажатии на «Редактирование ноутбуков» нужно также выделить определенный элемент из Listview, тогда все данные скопируются в открывшуюся panel (рисунок 8).

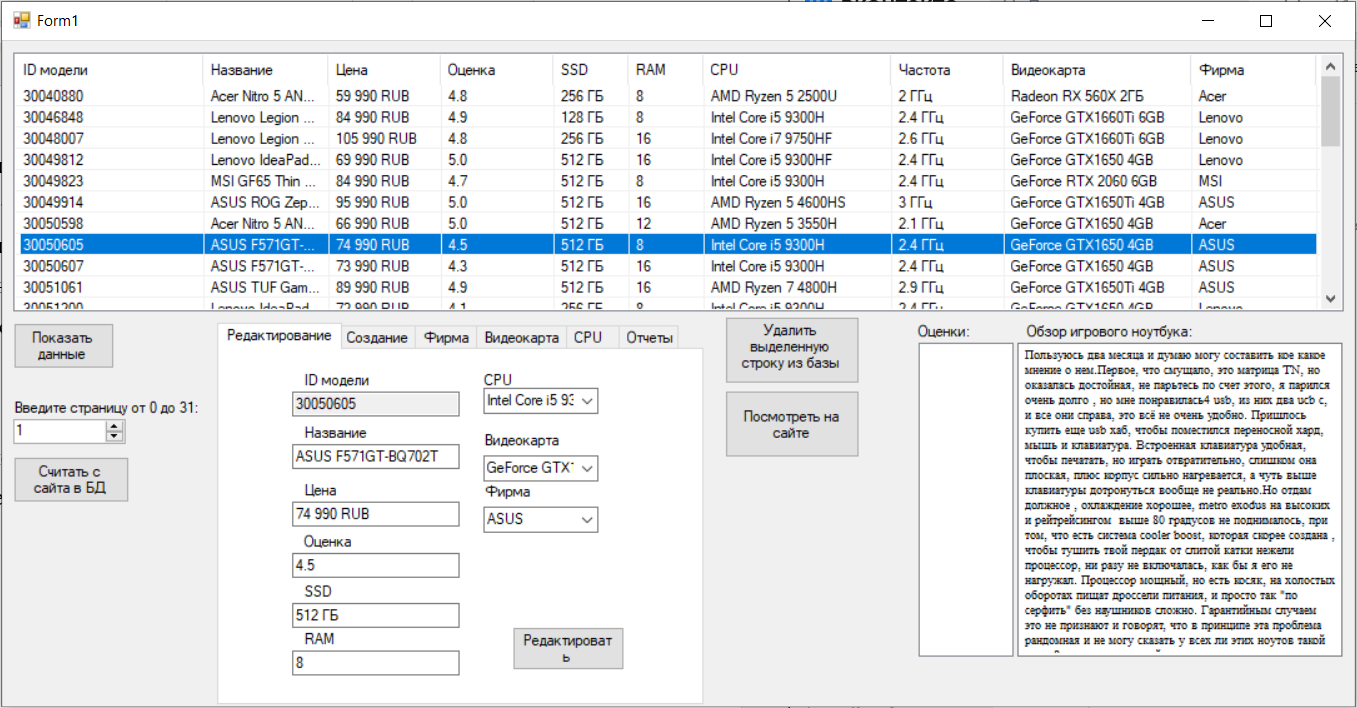


Рисунок 8 – Открывшаяся panel

Далее можно изменить определенные данные и после нажатия на кнопку «Редактировать» данные в БД будут обновлены (рисунок 9).

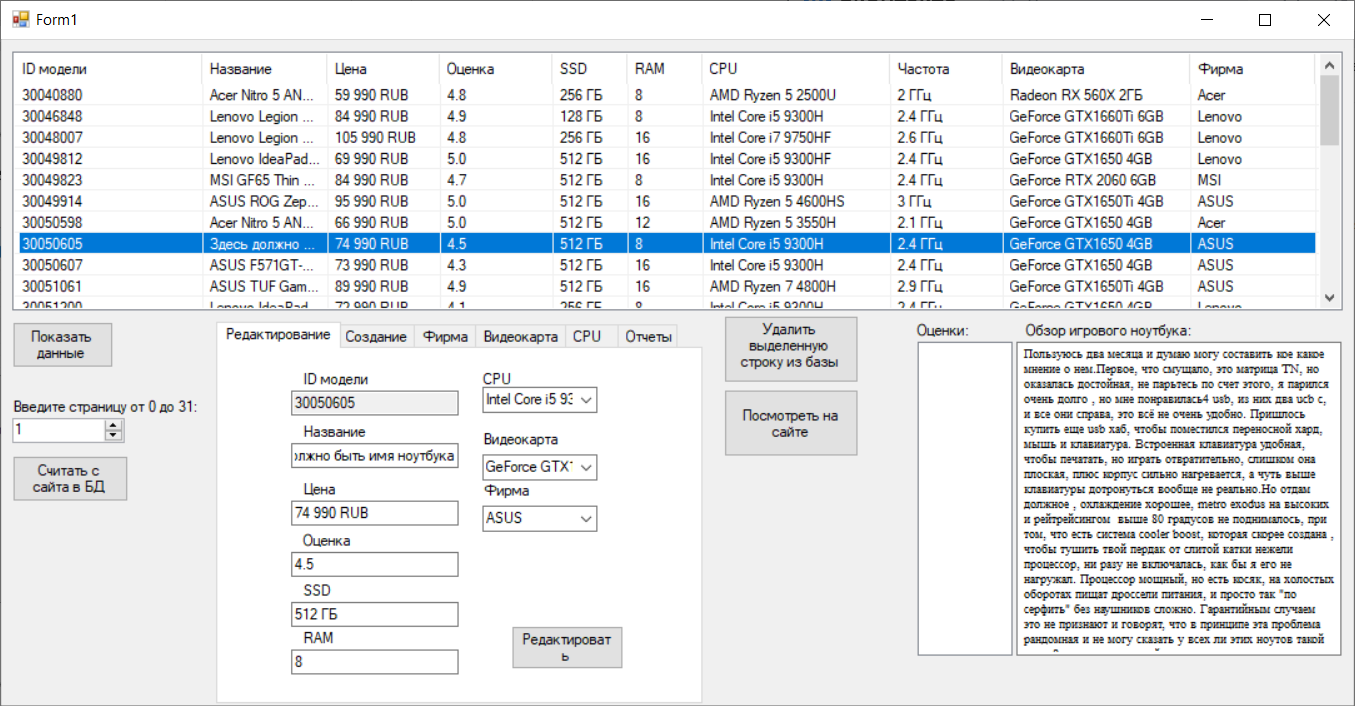


Рисунок 9 – Открывшаяся panel

Далее можно добавлять данные в БД с помощью кнопки «Создание» (рисунок 10-11).

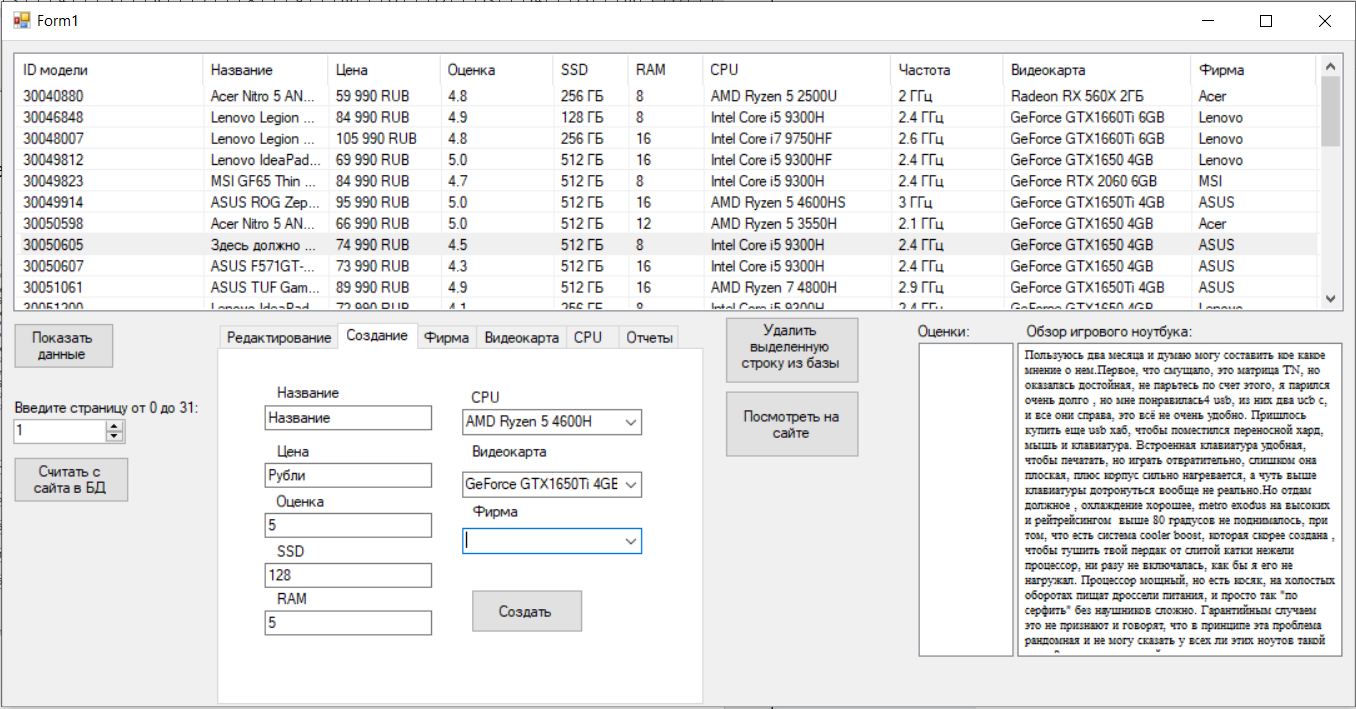


Рисунок 10 – Создание Модели

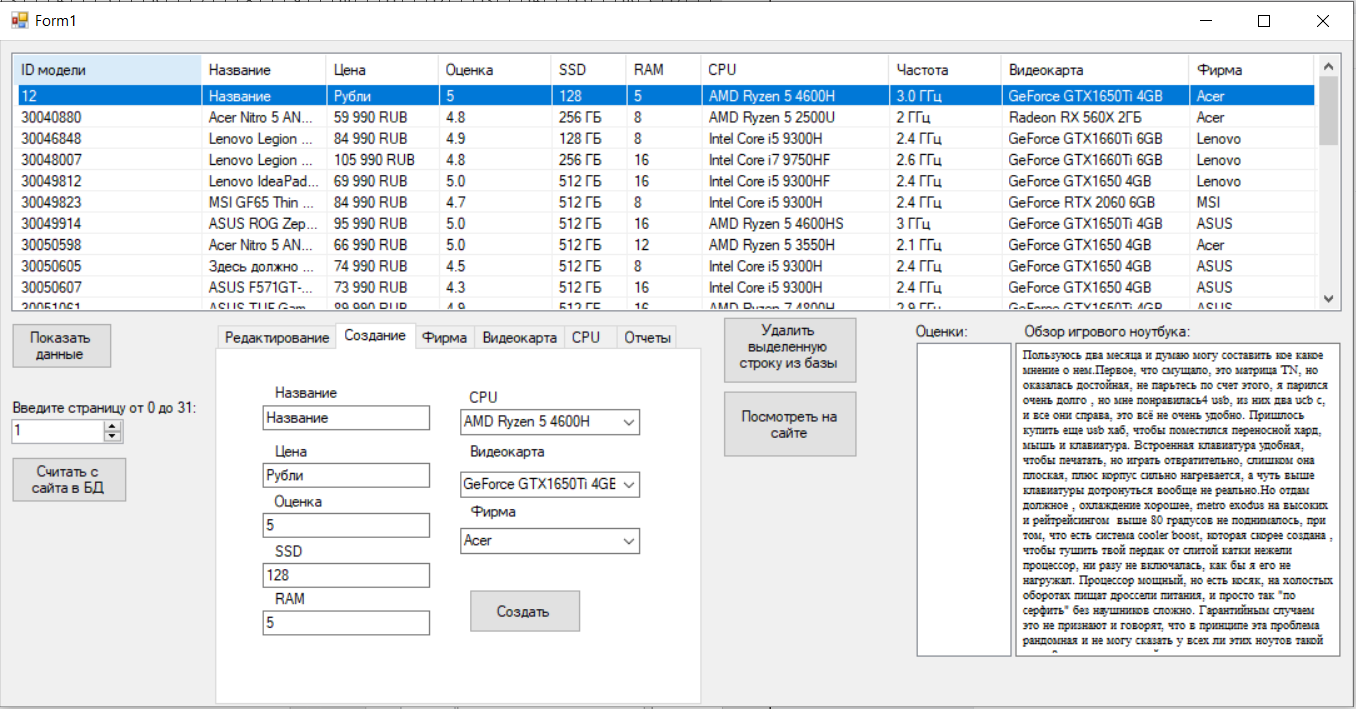


Рисунок 11 – Создавшаяся модель

Также можно редактировать значения других таблиц с помощью кнопок «Редактирование Фирмы», «Редактирование видеокарт», «Редактирование процессоров» (рисунок 12-16).

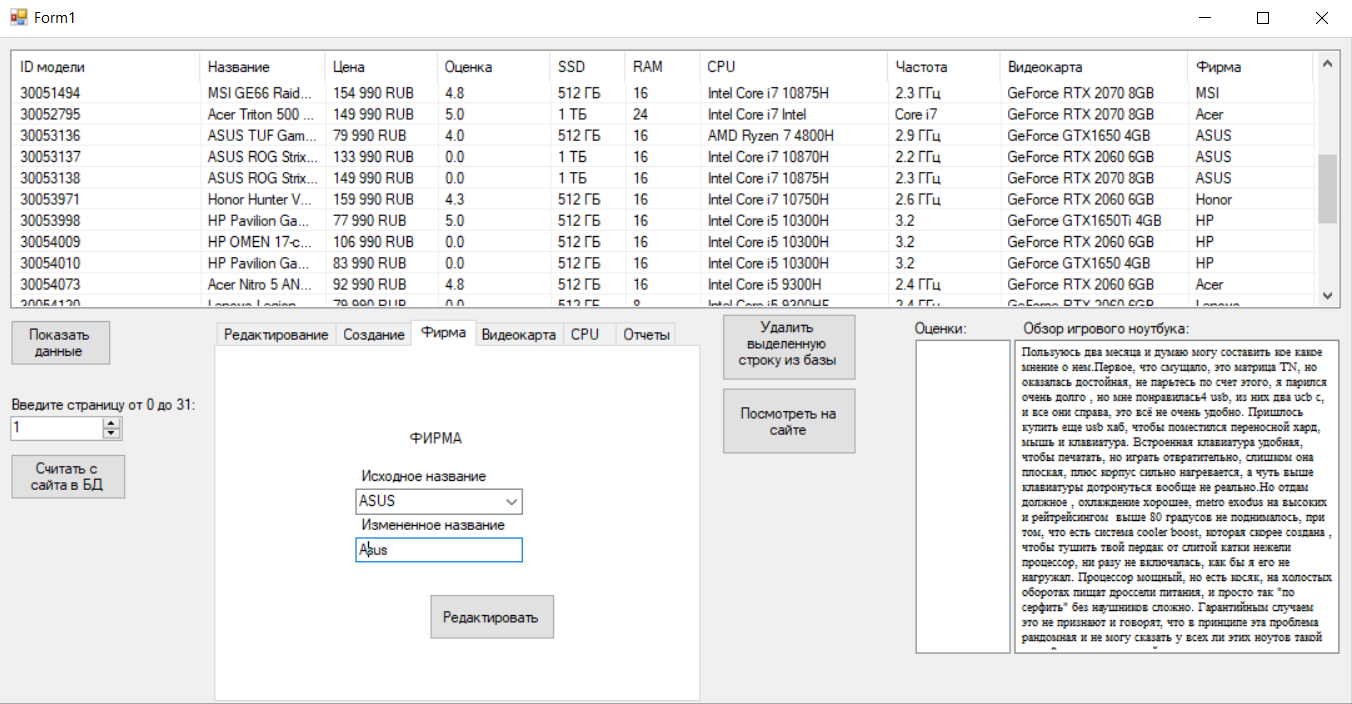


Рисунок 12 – Редактирование фирмы

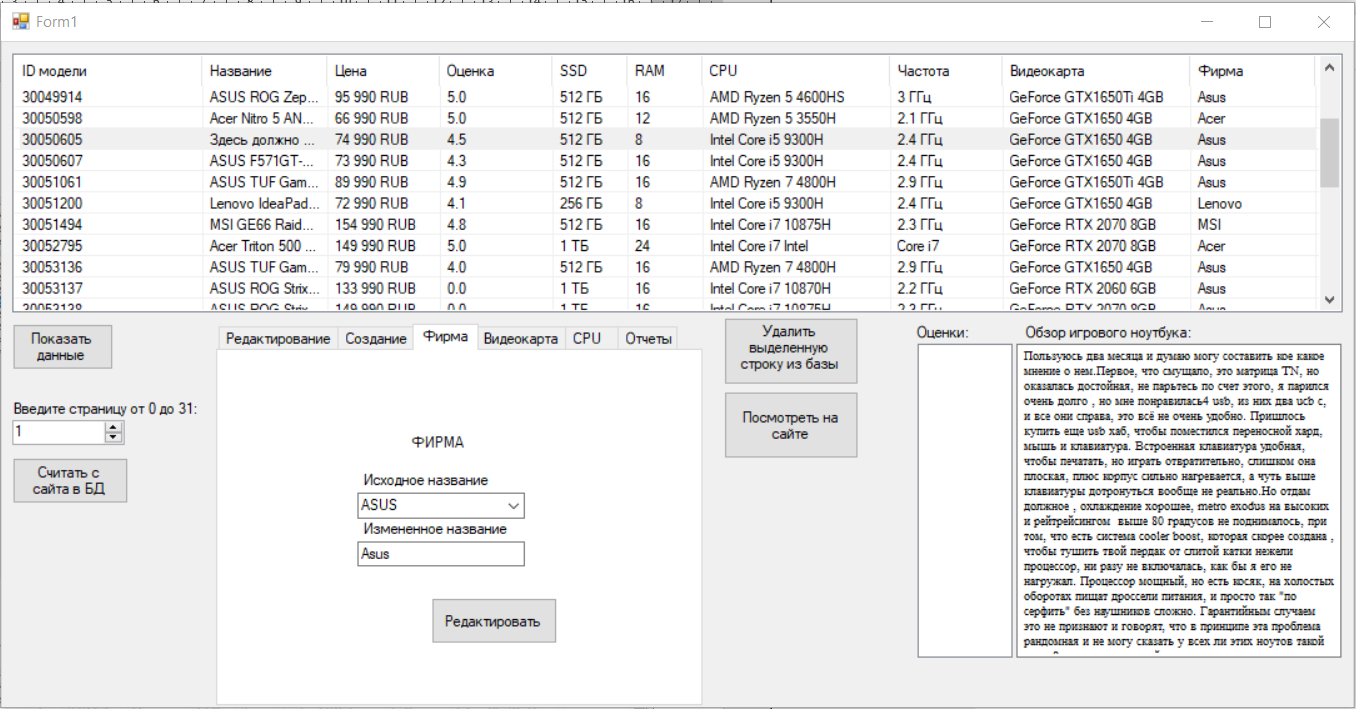


Рисунок 13 – Отредактированная фирма

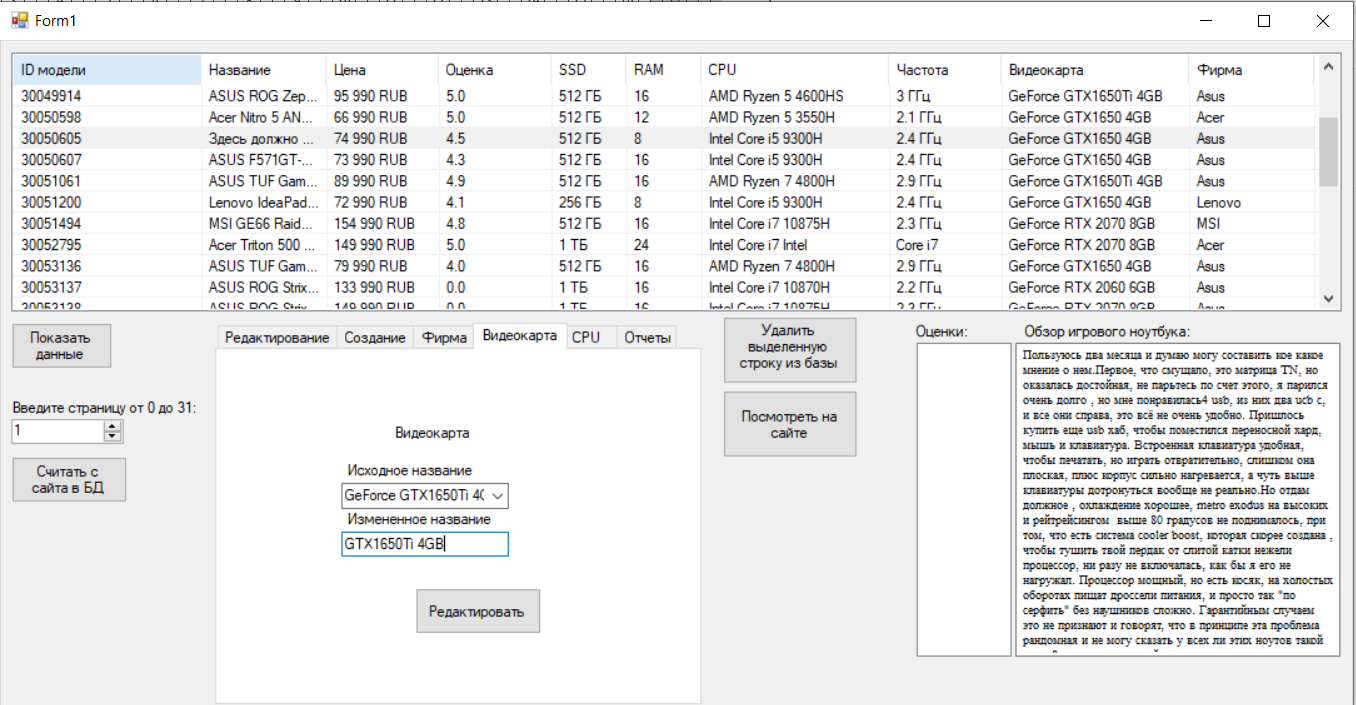


Рисунок 14 – Редактирование видеокарты



Рисунок 15 – Отредактированная видеокарта

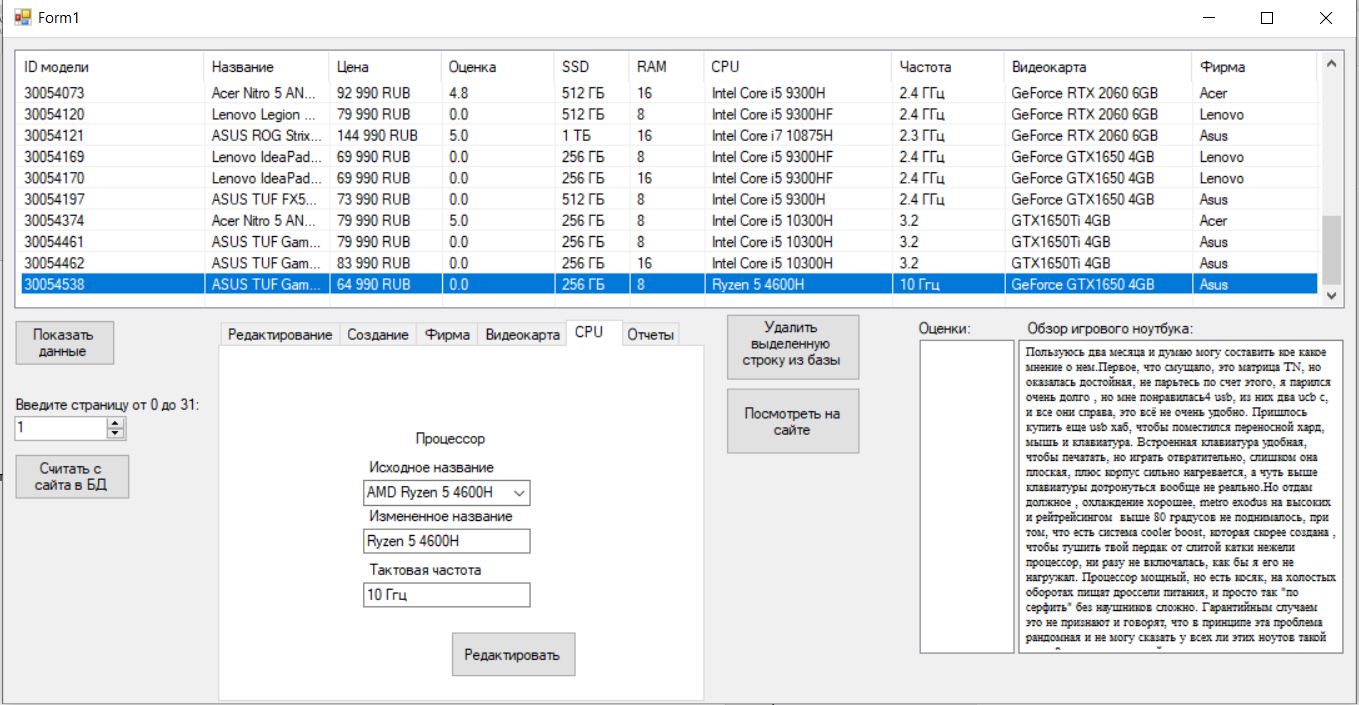


Рисунок 16 – Отредактированный процессор

## **2.5 Разработка шаблонов**

Было создано 2 шаблона для генерации отчетов.

Первый отчет будет показывать все записи о игровых ноутбуках, которые находятся в базе данных. Для этого создается таблица, в которой формируются 3 столбца: 1 столбец – id игрового ноутбука, согласно данным из сайта, 2 столбец – название игрового ноутбука, 3 столбец – цена игрового ноутбука.

Второй отчет будет показывать все отзывы о выбранном ноутбуке, который находится в базе данных. Для этого создается таблица, в которой формируются 3 столбца: 1 столбец – id отзыва, согласно данным из сайта, 2 столбец – оценка игрового ноутбука, 3 столбец – текст отзыва

Весь код реализованного компонента приведен в Приложении Б.

## **2.6 Разработка механизма генерации графиков на основе данных с ресурса**

Excel формирует график зависимости начисления бонусов от цены (рисунок 17). Для построения графика, берутся все записи игровых ноутбуков и берутся 2 атрибута: бонусы и цена. Весь реализованный функционал представлен в Приложении Б. Вспомогательная информация взята из 5 источника.

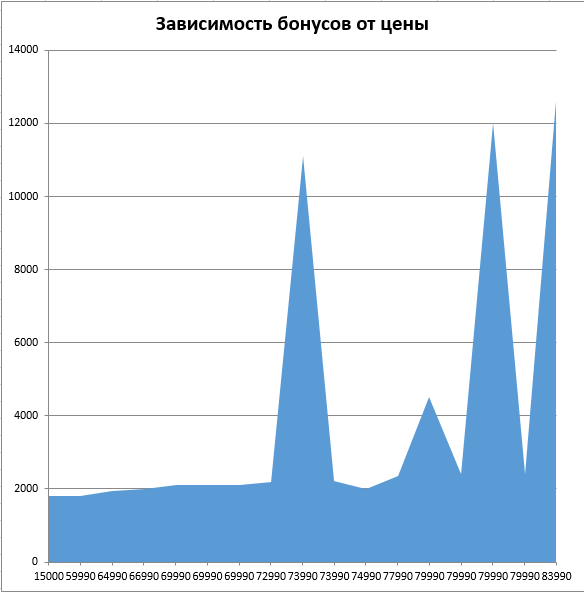


Рисунок 17 – График

## **2.7 Генерация отчетов на основе выбранных данных**

Далее будут показаны полученные отчеты (рисунки 18-19). Их генерация представлена в Приложении Б. Вспомогательная информация взята из 4 источника.

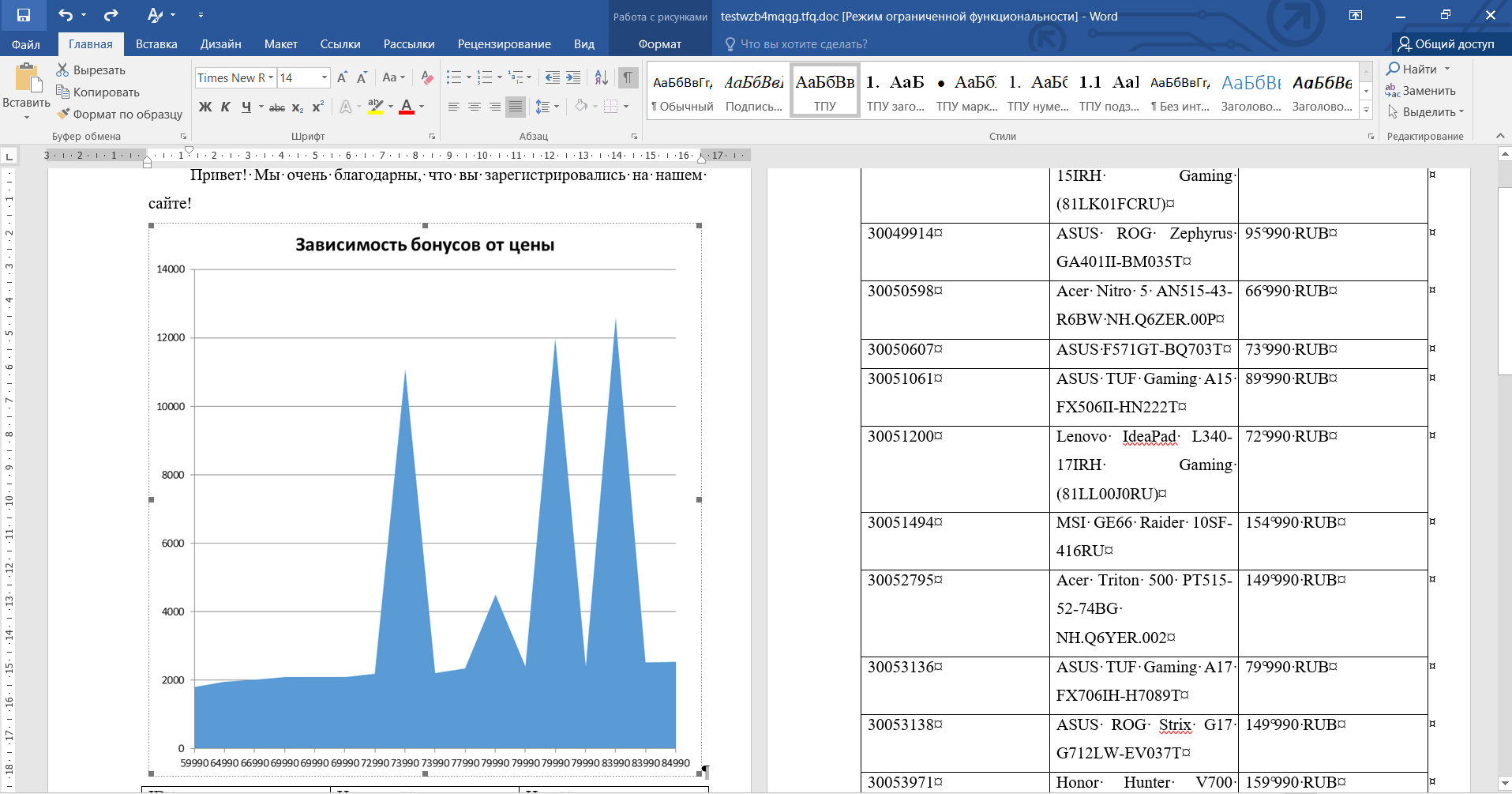


Рисунок 18 – Сгенерированный отчет №1

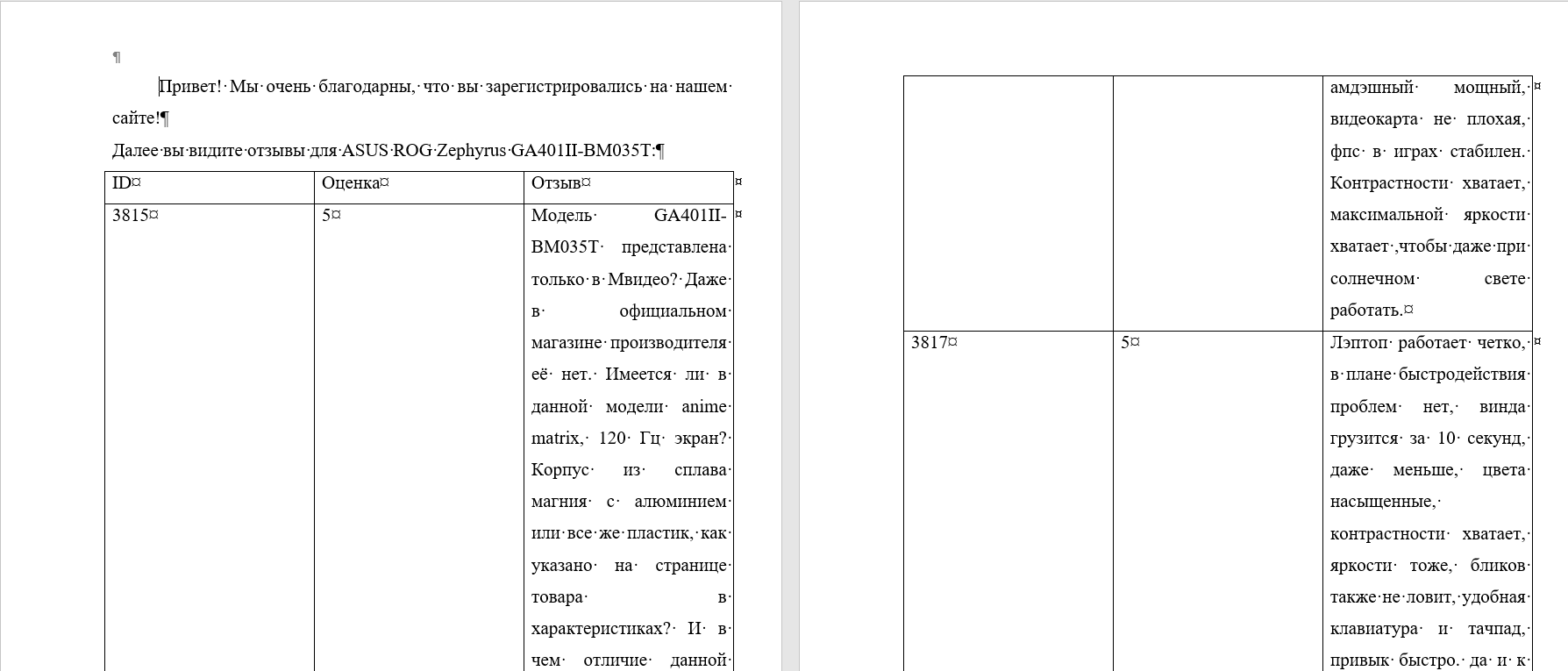


Рисунок 19 – Сгенерированный отчет №2

# 3 Заключение

В ходе выполнения были получены навыки выполнения полного цикла проектирования и создания базы данных, реализован функционал считывания информации и генерации отчетов и диаграммы с использование MS SQL Server, AngleSharp, Word.Interop, Excel.Interop на языке C#.

**4 Список использованных источников**

1. AngleSharp [Электронный ресурс]. – Режим доступа: https://anglesharp.github.io/. – Дата доступа: 25.05.2013.

2. Metanit [Электронный ресурс]. – Режим доступа: https://metanit.com/sharp/entityframework. – Дата доступа: 31.10.2015.

3. Microsoft [Электронный ресурс]. – Режим доступа: https://docs.microsoft.com/ru-ru/aspnet/mvc/overview/getting-started/getting-started-with-ef-using-mvc/creating-an-entity-framework-data-model-for-an-asp-net-mvc-application. – Дата доступа: 22.01.2019.

4. Microsoft [Электронный ресурс]. – Режим доступа: https://docs.microsoft.com/ru-ru/dotnet/api/microsoft.office.interop.word?view=word-pia. – Дата доступа: 15.06.2001.

5. Razilov-code [Электронный ресурс]. – Режим доступа: https://razilov-code.ru/2017/12/13/microsoft-office-interop-excel/. – Дата доступа: 13.12.2017.

Приложение А. Компонент наполнения и взаимодействия с базой данных

**Приложение А**

(справочное)

Ниже представленный код отвечает за компонент наполнения базы данных.

public async void Parse(string url1, string rateing)

{

//Объект класса

using (KPgamenotebookContext db = new KPgamenotebookContext())

{

Model gameNotebook = new Model();

Firm firm = new Firm();

Graphiccard graphiccard = new Graphiccard();

CPU cPU = new CPU();

//Подключение к HTML-страничке

IConfiguration config = Configuration.Default.WithDefaultLoader();

IBrowsingContext context = BrowsingContext.New(config);

IDocument doc1 = await context.OpenAsync(url1);

int watch = Convert.ToInt32(url1.Substring(url1.Length - 8));

var obj = db.Model;

bool twice = false;

foreach (Model u in obj)

{

if (watch == u.ID\_model)

{

twice = true;

}

}

if (twice == false)

{

//Характеристики

IEnumerable<IElement> shortDescriptionParams = doc1.All.Where(block =>

block.LocalName == "span"

&& block.ClassList.Contains("c-specification\_\_value")

&& block.ParentElement.ParentElement.ParentElement.ParentElement.LocalName == "table"

&& block.ParentElement.ParentElement.ParentElement.ParentElement.ClassList.Contains("c-specification\_\_table"));

//

//Вспомогательная переменная

string text;

//Имя

IElement name = doc1.QuerySelector("h1.fl-h1");

text = name.TextContent;

//IElement rating = rate.ToList()[0];

string textttt = text;

// Добавление фирмы

string[] words = text.Split(new char[] { ' ' });

var obj2 = db.Firm;

bool third = false;

foreach (Firm t in obj2)

{

if (words[2] == t.Name\_firm)

{

gameNotebook.ID\_firm = t.ID\_firm;

third = true;

break;

}

}

if (third == false)

{

firm.Name\_firm = words[2];

db.Firm.Add(firm);

db.SaveChanges();

gameNotebook.ID\_firm = firm.ID\_firm;

}

////CPU

IElement cpu = shortDescriptionParams.ToList()[1];

text = cpu.TextContent;

text = DeleteNT(text);

string[] words1 = text.Split(new char[] { ' ' });

string namecpu = words1[0] + " " + words1[1] + " " + words1[2] + " " + words1[3];

string freauq = words1[4] + " " + words1[5];

var obj4 = db.CPU;

bool fifth = false;

foreach (CPU t in obj4)

{

if (namecpu == t.Name)

{

gameNotebook.ID\_CPU = t.ID\_CPU;

fifth = true;

}

}

if (fifth == false)

{

//cPU.ID\_CPU = count;

//count++;

cPU.Name = namecpu;

cPU.Frequency = freauq;

db.CPU.Add(cPU);

db.SaveChanges();

gameNotebook.ID\_CPU = cPU.ID\_CPU;

}

var obj3 = db.Graphiccard;

bool fourth = false;

IElement graphiccard1 = shortDescriptionParams.ToList()[3];

text = graphiccard1.TextContent;

text = DeleteNT(text);

foreach (Graphiccard t in obj3)

{

if (text == t.Name)

{

gameNotebook.ID\_GC = t.ID\_GC;

fourth = true;

}

}

if (fourth == false)

{

graphiccard.Name = text;

db.Graphiccard.Add(graphiccard);

db.SaveChanges();

gameNotebook.ID\_GC = graphiccard.ID\_GC;

}

gameNotebook.Href = url1;

gameNotebook.ID\_model = watch;

string text1 = textttt.Remove(0, 23);

gameNotebook.Name = DeleteNT(text1);

//Оценка

gameNotebook.Rating = rateing;

//Цена

IElement price = doc1.QuerySelector("div.fl-pdp-price\_\_current");

string text2 = price.TextContent;

gameNotebook.Price = text2.Replace("₽", "RUB");

//Бонусы\*

IElement bonuses = doc1.QuerySelector("span.u-color-red.wrapper-text\_\_rouble");

if (bonuses == null)

{

gameNotebook.Bonuses = null;

}

else

{

string text3 = bonuses.TextContent;

gameNotebook.Bonuses = text3;

}

////Diagonal

//IElement diagonal = shortDescriptionParams.ToList()[0];

//text = diagonal.TextContent;

//gameNotebook. = DeleteNT(text);

//Ядра

//string url3 = url1 + "/specification";

//IDocument doc3 = await context.OpenAsync(url3);

//IEnumerable<IElement> elements = doc3.All.Where(block =>

// block.LocalName == "div"

// && block.ClassList.Contains("product-details-specification-column")

//);

//RAM

IElement ram = shortDescriptionParams.ToList()[2];

text = DeleteNT(text);

text = ram.TextContent;

string NoGB = DeleteNT(text);

gameNotebook.RAM = StringTOint(NoGB);

//SSD

IElement ssd = shortDescriptionParams.ToList()[4];

text = ssd.TextContent;

gameNotebook.SSD = DeleteNT(text);

//Отзывы

string url2 = url1 + "/reviews";

IDocument doc2 = await context.OpenAsync(url2);

//IElement rate = doc2.QuerySelector("span.text-cutter-wrapper");

//Текст

IEnumerable<IElement> opinion = doc2.All.Where(block =>

block.LocalName == "p"

&& block.OuterHtml.Contains("<p itemprop=\"description\">") == true

&& block.Children.All(u => u.LocalName == "span"));

//Оценка

IEnumerable<IElement> points = doc2.All.Where(block =>

block.LocalName == "span"

&& block.OuterHtml.Contains("<span itemprop=\"ratingValue\">") == true);

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

{

Reviews reviews = new Reviews();

bool sixth = false;

IElement element = opinion.ToList()[i];

IElement element1 = points.ToList()[i];

var obj5 = db.Reviews;

foreach (Reviews re in obj5)

{

if (element.TextContent == re.Review\_text)

{

sixth = true;

}

}

if (sixth == false)

{

reviews.Review\_text = element.TextContent;

reviews.Rating = element1.TextContent;

reviews.ID\_model = watch;

db.Reviews.Add(reviews);

}

reviews.Review\_text = element.TextContent;

}

db.Model.Add(gameNotebook);

db.SaveChanges();

}

else

{

db.SaveChanges();

}

}

}

public string DeleteNT(string text)

{

text = text.Replace("\n", "");

text = text.Replace("\t", "");

return text;

}

public int StringTOint(string text)

{

int ind = text.Length - 1;

text = text.Remove(text.Length - 3, 3);

return Convert.ToInt32(text);

}

public async Task<List<string>> GetLinks(string url)

{ //Подключение HTML главной странички

Console.WriteLine("Начанием загрузку главной страницы");

IConfiguration config = Configuration.Default.WithDefaultLoader();

IBrowsingContext context = BrowsingContext.New(config);

IDocument doc = await context.OpenAsync(url);

Console.WriteLine("Начинаем считывание главной страницы");

//Через LINQ запрашиваем блоки с тегом а

//Которые находятся в div

//C классом title

IEnumerable<IElement> aElements = doc.All.Where(block =>

block.LocalName == "a"

&& block.ParentElement.LocalName == "h4"

);

IEnumerable<IElement> rating = doc.QuerySelectorAll("span").Where(block =>

block.ClassName!=null && block.ClassName.Contains("fl-product-tile-rating\_\_stars-value")

);

string[] text=new string[rating.Count()];

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

IElement rateing = rating.ToList()[i];

text[i] = rateing.TextContent;

}

int b = 0;

List<string> output = new List<string>();

foreach (IElement a in aElements.ToList())

{

output.Add($"https://www.mvideo.ru{a.GetAttribute("href")}" + text[b]);

b++;

}

Console.WriteLine("Считывание главной страницы завершено");

return output;

}

Приложение Б. Компонент генерации отчетов и построения графика

**Приложение Б**

(справочное)

private void button6\_Click(object sender, EventArgs e)

{

string outputPath1 = @"C:\Users\ReaLBERG\Desktop\3 курс\АИС\Отчеты\Ценабонусы" + Path.GetRandomFileName() + ".xlsx";

KPgamenotebookContext db2 = new KPgamenotebookContext();

int num2 = db2.Model.Count();

var models = db2.Model;

Model[] gameNotebooks2 = new Model[num2];

int p = 1;

Excel.Application excelApp = new Excel.Application();

Excel.Workbook workBook;

Excel.Worksheet workSheet;

workBook = excelApp.Workbooks.Add();

workSheet = (Excel.Worksheet)workBook.Worksheets.get\_Item(1);

int[] vs = new int[num2];

int[] vs2 = new int[num2];

int k = 0;

foreach (var t in models)

{

int value;

int value1;

int.TryParse(string.Join("", t.Price.Where(c => char.IsDigit(c))), out value);

int.TryParse(string.Join("", t.Bonuses.Where(c => char.IsDigit(c))), out value1);

vs[k] = value;

vs2[k] = value1;

k++;

}

int temp;

for (int i = 0; i < vs.Length - 1; i++)

{

for (int j = i + 1; j < vs.Length; j++)

{

if (vs[i] > vs[j])

{

temp = vs[i];

vs[i] = vs[j];

vs[j] = temp;

temp = vs2[i];

vs2[i] = vs2[j];

vs2[j] = temp;

}

}

}

for (p = 1; p < 24; p++)

{

workSheet.Cells[1, p] = (vs[p - 1]);

workSheet.Cells[2, p] = (vs2[p - 1]);

}

Excel.ChartObjects chartObjs = (Excel.ChartObjects)workSheet.ChartObjects();

Excel.ChartObject chartObj = chartObjs.Add(10, 50, 500, 500);

Excel.Chart xlChart = chartObj.Chart;

xlChart.ChartType = Excel.XlChartType.xlAreaStacked;

Excel.SeriesCollection seriesCollection = (Excel.SeriesCollection)xlChart.SeriesCollection(Type.Missing);

Excel.Series series = seriesCollection.NewSeries();

series.XValues = workSheet.get\_Range("A1", "Q1");

series.Values = workSheet.get\_Range("A2", "Q2");

xlChart.HasTitle = true;

xlChart.ChartTitle.Text = "Зависимость бонусов от цены";

xlChart.HasLegend = false;

excelApp.Visible = true;

excelApp.UserControl = true;

workSheet.SaveAs(outputPath1);

object misValue = System.Reflection.Missing.Value;

xlChart.Export("C:\\Users\\ReaLBERG\\Desktop\\3 курс\\АИС\\Graf.bmp", "BMP", misValue);

excelApp.Quit();

Word cOMFormatter = new Word(@"C:\Users\ReaLBERG\Desktop\3 курс\АИС\Отчеты\lb9tt.doc");

KPgamenotebookContext db = new KPgamenotebookContext();

Console.WriteLine("Пожалуйста, введите имя: ");

string name = Console.ReadLine();

cOMFormatter.Replace("{Имя}", name);

int num = db.Model.Count();

string[] start = new string[3] { "ID", "Название", "Цена" };

var gamenotebook = db.Model;

Model[] gameNotebooks = new Model[num];

int y = 0;

foreach (var t in gamenotebook)

{

gameNotebooks[y] = t;

y++;

}

cOMFormatter.TableCreate(num + 1, 3, start, gameNotebooks);

cOMFormatter.Close();

}

private void button8\_Click(object sender, EventArgs e)

{

using (KPgamenotebookContext db = new KPgamenotebookContext())

{

Word cOMFormatter = new Word(@"C:\Users\ReaLBERG\Desktop\3 курс\АИС\Отчеты\lb9ttотзывы.doc");

int num = Convert.ToInt32(textBox14.Text);

string[] start = new string[3] { "ID","Оценка", "Отзыв" };

var review = db.Reviews;

Reviews[] reviews = new Reviews[num];

int ID = Convert.ToInt32(textBox13.Text);

int y = 0;

foreach (var t in review)

{

if (t.ID\_model == ID)

{

reviews[y] = t;

y++;

}

}

cOMFormatter.TableCreateReview(num + 1, start.Length, start, reviews);

cOMFormatter.Replace("{Имя}", textBox12.Text);

}

}