Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

ОТЧЕТ

по лабораторной работе

«Анализ цен на номера в отелях»

Выполнил:

Студент: гр. 956341

Морозов Е.П.

Проверил:

Стержанов М. В.

Минск 2019

**Содержание**

[1 Постановка задачи 3](#_Toc26857285)

[2 Работа с данными 3](#_Toc26857286)

[2.1 Сбор данных 3](#_Toc26857287)

[2.2 Обработка данных 4](#_Toc26857288)

[2.3 Хранение данных 4](#_Toc26857289)

[3 Анализ 6](#_Toc26857290)

[3.1 Соотношение стоимости и рейтинга 7](#_Toc26857291)

[3.2 Соотношение отелей с разным рейтингом 7](#_Toc26857292)

[3.3 Соотношение стоимости ко времени 8](#_Toc26857293)

[4 Вывод 9](#_Toc26857294)

1 Постановка задачи

Целью данной лабораторной работы является написание проекта для сбора и анализа данных об отелях Минска за следующий год от текущего момента.

Ссылка на проект: https://github.com/EugeneMv/ScraperOnCSharp.

2 Работа с данными

2.1 Сбор данных

Получение данных осуществляется с ресурса Booking.com при помощи пакета ScrapySharp для языка C#. Для эмулирования бразуера используется класс ScrapingBrowser. Класс содержит методы по загрузке страницы.

Для получения списка отелей с сайта используются селекторы, которые идентифицируют нужные теги <div>, содержащие информацию о каждом отеле.

Для формирования URL ссылки для доступа к странице используется вспомогательный класс SearchManager. Класс формирует URL строку на основе следующих данных:

* локация;
* дата начала аренды;
* дата окончания аренды;
* строка отсчета для перехода по страницам.

Диаграмма классов для работы со ScrapySharp и извлечения данных из интернет ресурса представлена на рисунке 2.1.

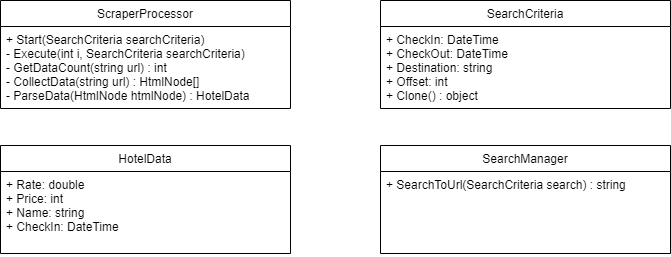


Рисунок 2.1 – диаграмма классов

Продолжительность аренды составляет 7 дней. Для каждый 7 дней следующего года собираются данные об отелях, их ценах и рейтингах.

Данные извлекаемые для каждого отеля:

* название отеля;
* рейтинг отеля;
* стоимость номера.

2.2 Обработка данных

Чтобы данные приняли корректный вид и формат используются сервис-класс DataExtractor, в котором содержатся методы с логикой регулярных выражений для каждого из полей.

На сайте формат записи стоимости: US$1,209. Для записи значения в базу данных используется метод HotelPrice класса DataExtractor. На выходе метода в данном примере выйдет 1209 типа данных int. Код метода HotelPrice представлен на листинге 2.1.

public static int HotelPrice(string s)

{

Regex regex = new Regex(@"([0-9]+\,?[0-9]\*)", RegexOptions.IgnoreCase);

var b = regex.Match(s);

var res = int.Parse(b.Value, NumberStyles.AllowThousands);

return res;

}

Листинг 2.1 – метод HotelPrice

2.3 Хранение данных

Для хранения данных используется Entity Framework. Entity Framework представляет специальную объектно-ориентированную технологию на базе фреймворка .NET для работы с данными.

Для работы с данными создано 2 модели:

* HotelPrice;
* HotelHeader.

HotelHeader хранит статичную информацию об отеле. Структура представлена в таблице 2.1.

Таблица 2.1 – структура модели «ApplicationUser»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Id | int | Идентификатор отеля |
| Name | string | Название отеля |
| Rate | double | Рейтинг отеля |

HotelPrice хранит стоимость 7 дней норма номера в отеле на определенное число. Структура представлена в таблице 2.1.

Таблица 2.1 – структура модели «ApplicationUser»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Id | int | Идентификатор записи о цене |
| HotelHeaderId | int | Идентификатор отеля |
| Price | int | Стоимость |
| CheckDate | DateTime | Дата заезда |

Таблицы HotelHeader и HotelPrice имеют отношение 1 ко многим, так как в разные моменты времени стоимость номера разная.

Центральной концепцией Entity Framework является понятие сущности. Сущность представляет набор данных, ассоциированных с определенным объектом. Поэтому данная технология предполагает работу не с таблицами, а с объектами и их наборами. Пример работы с таблицей HotelPrices на листинге 2.2.

Model1 model1 = new Model1();

var prices = model1.HotelPrices.ToList();

var prc = prices.OrderBy(p => p.CheckDate).ToList();

var date = DateTime.Now;

for (int i = 1; i <= 12; i++, date = date.AddMonths(1))

{

chart5.Series[0].Points.AddXY(date.ToString("MMM"), prc.Where(p => p.CheckDate.Month == date.Month).Average(p => p.Price));

}

Листинг 2.2 – пример работы с таблицей

Данные извлекаются из таблицы при нажатии завершении загрузки данных и помещаются на UI.

3 Анализ

Пользовательский интерфейс, на котором можно просмотреть построенные графики, показан на рисунке 3.1.

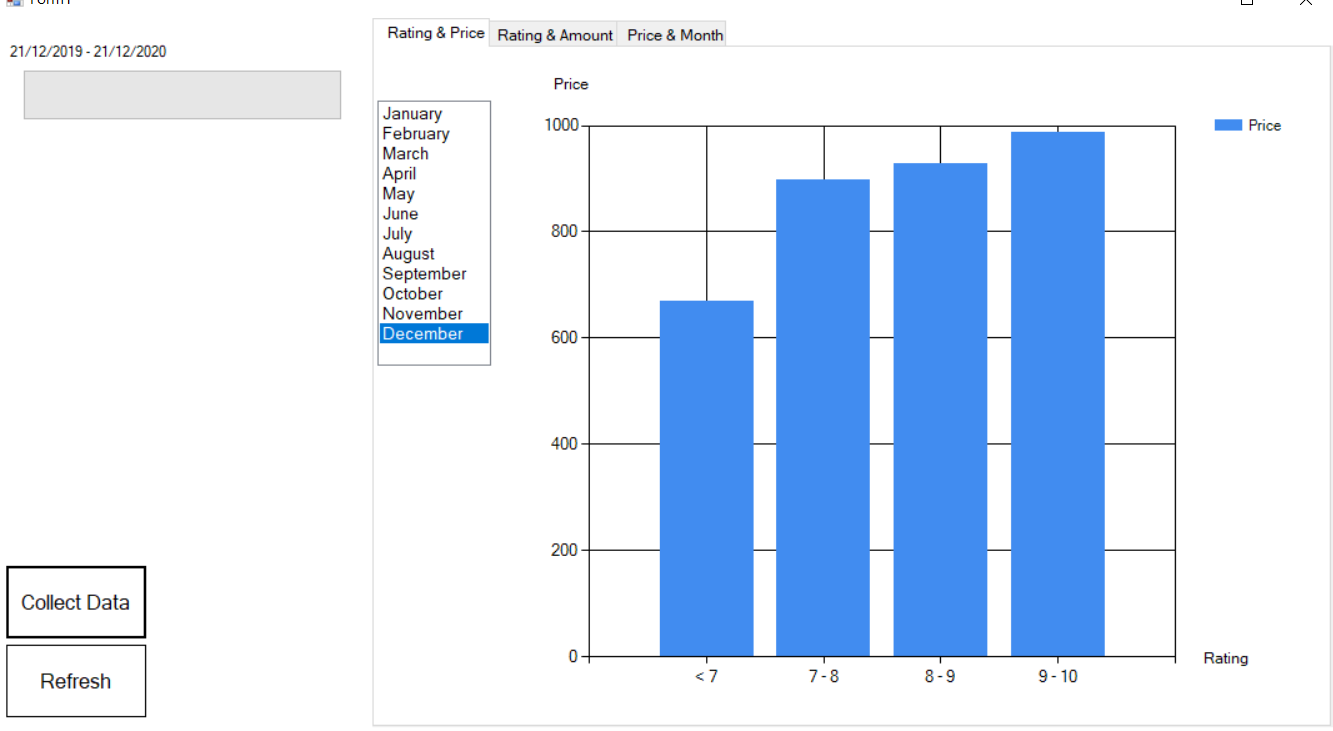


Рисунок 3.1 – пользовательский интерфейс

Для обработки событий приложения, таких как нажатие кнопок Collect Data (запускает сбор данных) и Refresh Data (перерисовка графиков) создан класс Form1.

В классе реализованы методы обращения к базе данных через класс-контекст Model1 и обработки данных для формирования массива пар значений X и Y для отображения на диаграммах.

Логика сбора и обработки данных для отображения расположена в следующих методах класса Form1:

* InitPages – вызывается при запуске программы;
* button2\_Click\_1 – вызывается при нажатии кнопки Refresh;
* listBox1\_SelectedIndexChanged – вызывается при смене месяца на первом графике.

Для отображения графиков используется стандартный UI элемент Windows Forms Chart. Chart предоставляет возможность отображения различных видов диаграмм без использования сторонних библиотек.

На верхней части интерфейса расположены 3 вкладки, для переключения между окнами с графиками.

3.1 Соотношение стоимости и рейтинга

На вкладке Rating & Price показано соотношение средней стоимости за номер в отелях к рейтингу отелей. Поле Price рассчитывается как среднее арифметическое за каждый отдельный месяц.

График изображен на рисунке 3.2.

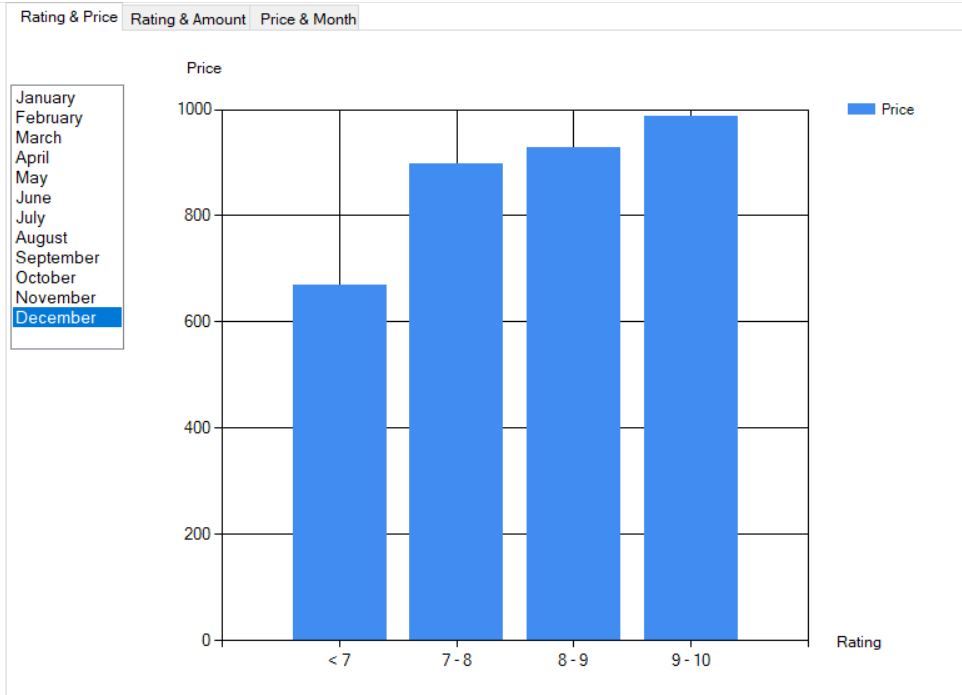


Рисунок 3.2 – соотношение цены и рейтинга

Тип диаграммы – столбчатая диаграмма. На графике видно, что для декабрь месяца значения поля Price наивысшее при значении поля Rating 9-10, что значит средняя цена за номер в отеле наивысшая при рейтинге отеля от 9 до 10 (10 – максимально возможный рейтинг).

3.2 Соотношение отелей с разным рейтингом

На вкладке Rating & Amount показана доля отелей с различными рейтингами.

Тип диаграммы – круговая диаграмма. Круговая диаграмма – это один из способов графического представления количественных данных. Представляет собой круг, разделенный на секторы, относительный размер которых пропорционален численным значениям.

На диаграмме для каждого диапазона рейтинга отелей посчитана доля отелей с определенным рейтингом от общего количества отелей (рисунок 3.3).

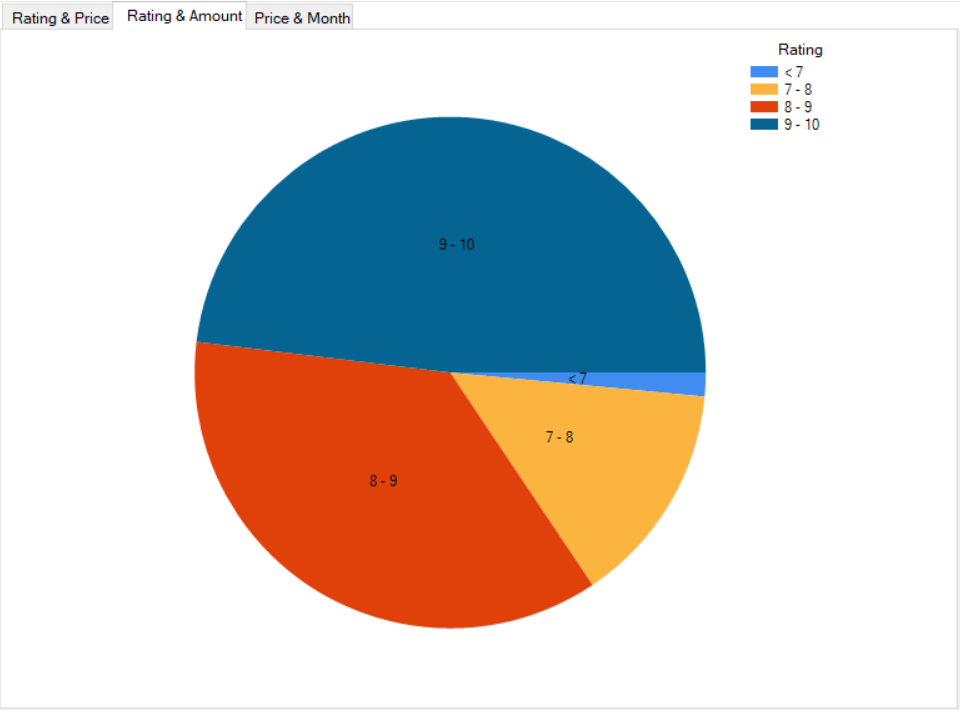


Рисунок 3.3 – соотношение отелей с разным рейтингом

Разбиение отелей на группы происходит по следующим признакам:

< 7 – рейтинг отеля ниже 7;

7-8 – рейтинг отеля находится в диапазоне от 7 до 8;

8-9 – рейтинг отеля находится в диапазоне от 8 до 9;

9-10 – рейтинг отеля выше 9.

На графике видно, что почти половина отелей имеет рейтинг от 9.

3.3 Соотношение стоимости ко времени

На вкладке Price & Month изображена динамика изменения средней стоимости номера в течение следующего года. Поле Price показывает величину средней стоимости.

Стоимость рассчитывается как среднее арифметическое стоимости номера в отелях в рамках конкретного месяца.

Для отображения данный используется линейный график зависимости стоимости от месяца (рисунок 3.4). Каждый месяц имеет свой номер от 1 до 12, который означает его порядковый номер в календаре.

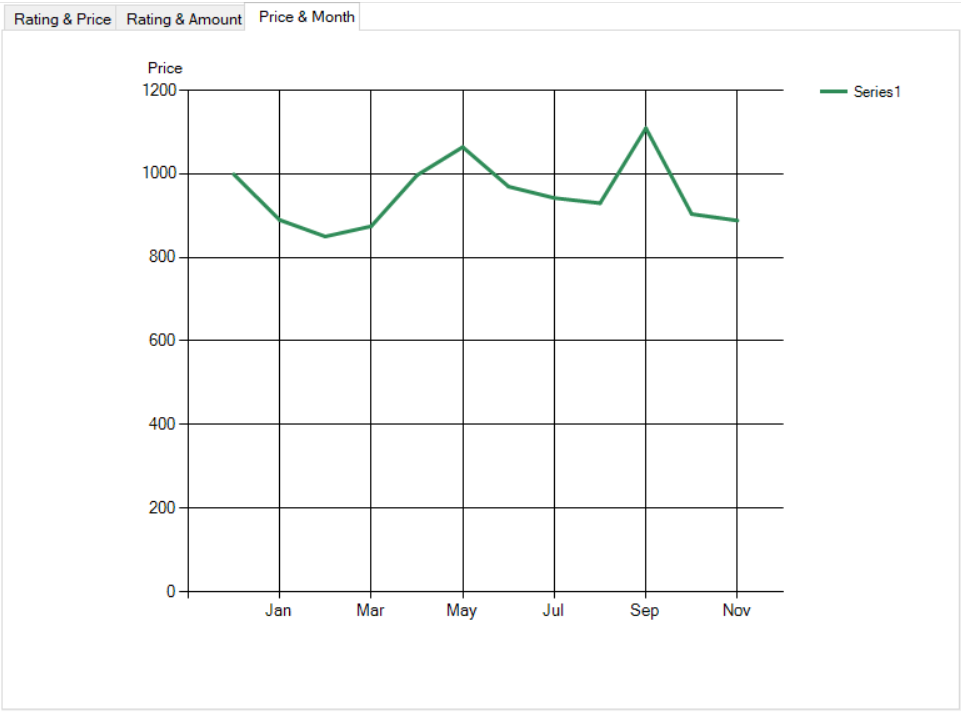


Рисунок 3.4 – соотношение средней цены и времени

4 Вывод

В данной работе рассмотрен веб-скрапинг при помощи пакета ScrapySharp для языка C#.

Для создания пользовательского интерфейса и отображения данных использовался шаблон Windows Forms.

Для создания реляционной базы данных и работы с ней использовался Entity Framework.

Изучены способы доступа к элементам html страницы при помощи XPath и css selector, а также способы извлечения полезных данных из этих элементов.

На основе данных произведен анализ и построены графики.