**Национальный исследовательский университет**

**Высшая школа экономики**

Факультет **«Бизнеса и менеджмента»**

Кафедра

**Управления информационными системами и цифровой инфраструктурой**

***Групповой проект***

***по дисциплине***

***«Программирование»***

Выполнили студенты 152 (1) группы:

Анчуткина Ярослава Григорьевна

Митракова Тамара Сергеевна

Заспенко Владислав Владимирович

Москва, 2016 год.

**Оглавление.**

1. Условие задачи.
2. Функции разрабатываемого приложения.

2.1. Функции, выполняемые приложением.

2.2. Описание интерфейса пользователя.

2.3. Описание базы данных.

1. Структура приложения.

3.1. Описание классов и методов.

1. Контрольный пример.
2. **Условие задачи**

Задача заключается в создании Windows Forms приложения для бронирования номеров в отеле. Пользователю будет предоставлена возможность выбрать такие параметры как:

* Город для временного проживания
* Категорию номера
* Ценовой диапазон
* Период проживания в отеле

Приложение включает в себя базу данных, из которой, после взаимодействия пользователя с интерфейсом приложения, будет извлекаться необходимая для работы информация.

1. **Функции разрабатываемого приложения.**

*2.1. Функции, выполняемые приложением*

Данное приложение “Hotel Booking” представляет собой Windows приложение, состоящие из трех окон, благодаря которому пользователь может найти и забронировать отель в России, существующий в базе данных.

Все исходные данные для поиска отеля пользователь выбирает с помощью экранных форм, содержащих следующие поля с выбором значения: “Выберете город”, “Выберете категорию”, “Дата заезда”, “Дата выезда”. А также вводит с помощью такого поля для текстового ввода: “Ценовой диапазон от/до”.

При корректном выборе и вводе всех значений пользователю предлагается список отелей, подходящих под выбранные и введенные пользователем данные.

Также предусмотрена возможность бронирования отеля.

Все исходные данные для бронирования отеля пользователь вводит с помощью экранных форм, содержащих следующие поля для текстового ввода: “Фамилия Имя Отчество”, “Телефон”, “Паспорт(Серия, Номер)”, “Пол”. А также имеются поля для выбора (по желанию): “Такси в отель”, “Такси из отеля”. Главная особенность приложения состоит в том, что выбранные пользователем данные с первой и второй страниц, автоматически сохраняются в памяти приложения, и пользователю не требуется тратить время на повторное их введение, то есть на третьем окне пользователю достаточно ввести только свои личные данные.

При корректном вводе личных данных пользователь может забронировать отель, нажав на кнопку забронировать. После чего происходит возврат на главную страницу, что позволит снова начать выбирать отель, если это необходимо.

*2.2 Описание интерфейса*

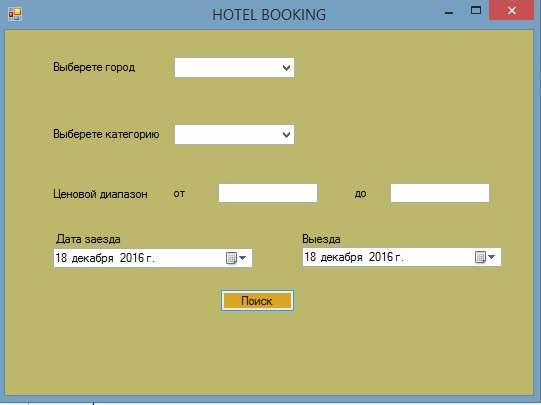
При запуске программы появляется главное окно. 

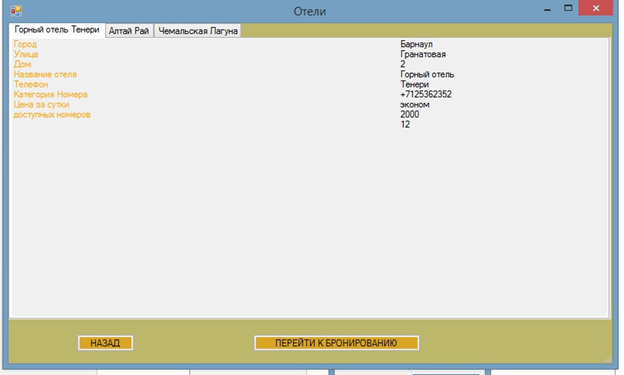
Рис. 2.1. Интерфейс главного окна программы

Поля “Выберете город” и “Выберете категорию” предназначены для выбора города и категории отеля из существующего списка, который хранится в базе данных.

Поле “Ценовой диапазон от” позволяет пользователю ввести число, которое обозначает минимальную цену за сутки проживания в отеле, поле максимального диапазона “до” позволяет пользователю ввести число, которое обозначает максимальную цену отеля. Длина чисел не должна превышать 7 знаков.

Поля “Дата заезда”, “Дата выезда ” позволяют выбрать дату заезда в отель и выезда из отеля. При этом дата заезда не может быть меньше, чем следующий день, потому что заселение в отель происходит ровно в 00:00. Дата выезда, соответственно не может быть меньше, чем дата заезда. При неправильном вводе данных, пользователю будет выведено уведомление, о невозможности осуществить поиск, так как данные введены некорректно. После нажатия на кнопку “Поиск” главная страница бронирования отелей закрывается и открывается следующее окно “Отели” (см. рис 2.2).

При нажатии кнопки “Поиск” открывается второе окно “Отели”

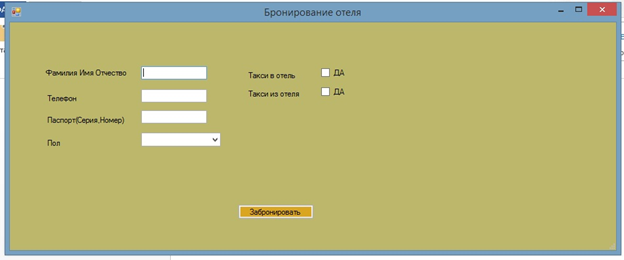
Рис. 2.2. Интерфейс второго окна - выбора отеля

Данное окно показывает результат поиска отелей по параметрам, заданным пользователем ранее в первом окне. Вверху окна располагаются вкладки с названиями отелей. В основной части окна располагается основная информация про каждый из отелей, в зависимости от открытой вкладки.

В нижней части окна располагаются две кнопки.

Кнопка “НАЗАД” возвращает пользователя в главное окно программы.

Если все данные введены корректно, то при нажатии кнопки “Перейти к бронированию” открывается третье окно программы “Бронирование отеля”. Если же количество свободных номеров в отеле, удовлетворяющих выбранной дате и категории, равно 0, то пользователю будет недоступна опция по бронированию отеля, в связи с отсутствие мест.

Рис. 2.3. Интерфейс третьего окна - бронирования отеля

Данное окно предназначено для бронирования выбранного пользователем отеля. В данном окне пользователю требуется только ввести контактные данные о себе указать (при желании) трансфер из/в отель, остальные данные вводить не требуется, так как приложение запоминает ранее выбранную информацию. Поля “Фамилия Имя Отчество”, “Телефон” и “Паспорт (Серия, Номер)” предназначены для ввода пользователем своих личных данных или того человека, на которого он хочет забронировать отель. Поле “Пол” предоставляет возможность выбрать пол пользователя или того человека, на которого бронируется отель. Поля “Такси в отель” и “Такси из отеля“ позволяют пользователю заказать данный вид услуги при выборе поля “ДА”. Если пользователь не отмечает поле “ДА”, то услуга не является заказанной.

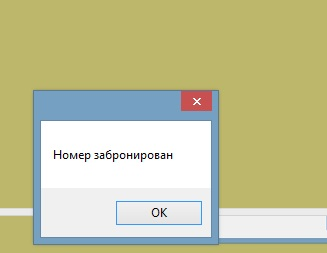
Если все данные введены корректно, то после нажатия кнопки “Забронировать” на экран выводится следующее окно:

Рис. 2.4. Окно подтверждения бронирования номера в отеле.

*2.3 Описание базы данных*

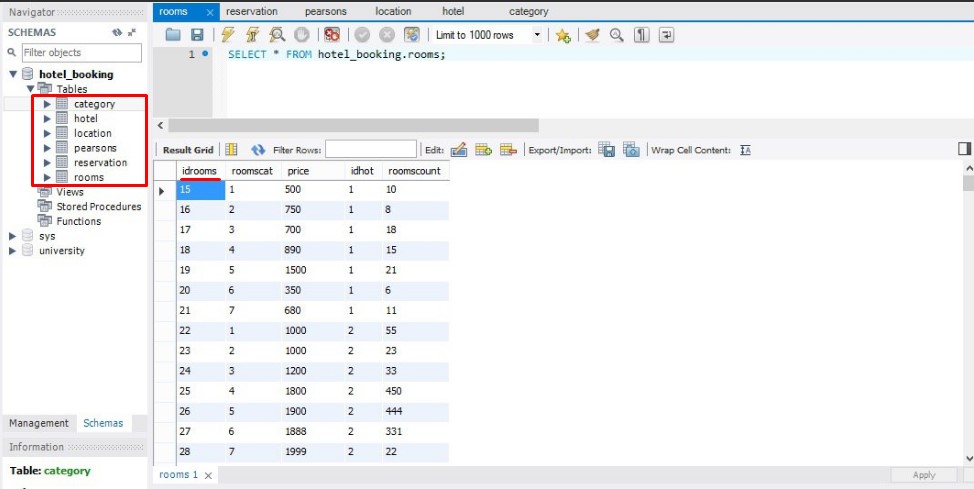
База данных, используемая данным приложением расположена на локальном сервере программы SQL WorkBench. В ней имеется 6 сущностей: категория,отель,местоположение,человек,резервация,комнаты.

Рис. 2.3.1. Список сущностей и пример одной из них.

Каждая сущность имеет свои поля, где для каждого поля прописаны параметры и тип данных, который может вводиться в него.

В базе данных имеются сущности:

* Category
* Hotel
* Location
* Persons
* Reservation
* Rooms

Category - Предоставление информации о категориях номеров отелей.

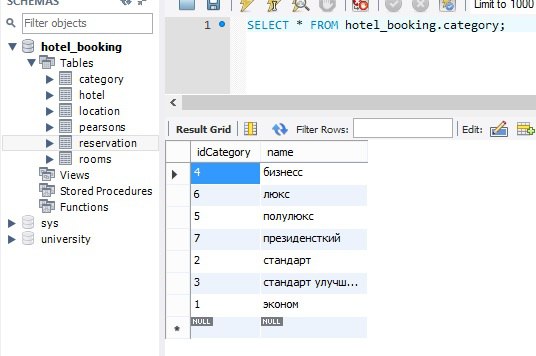


Рис. 2.3.2. Поля сущности - Category

Hotel - Информация об отелях

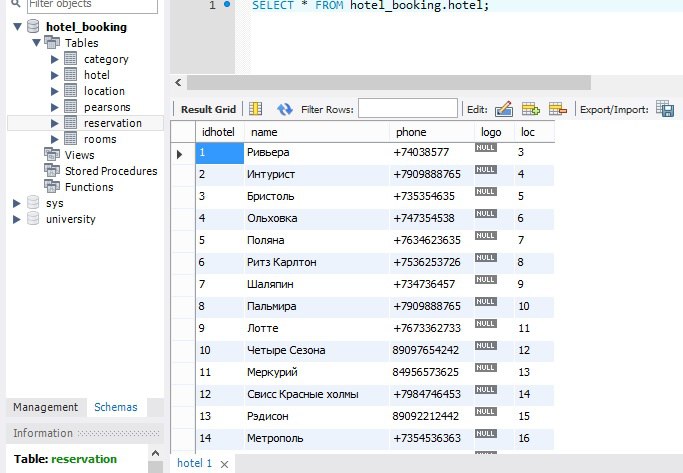


Рис. 2.3.3. Поля сущности - Hotel

Location - Информация о месте, где расположен отель

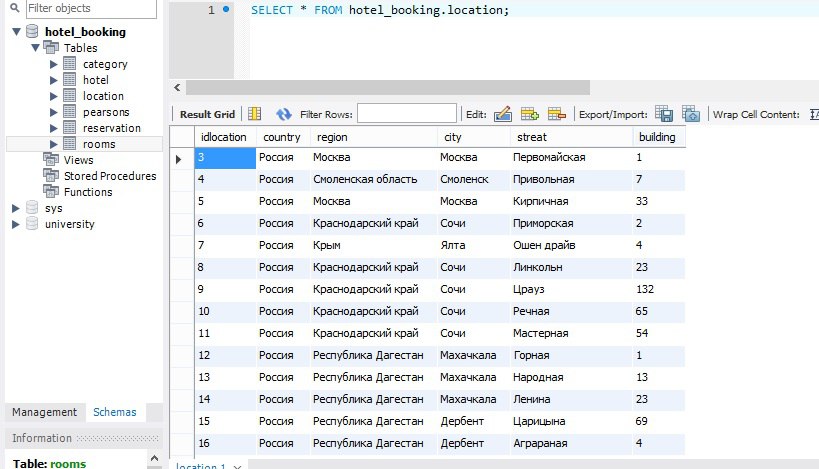


Рис. 2.3.4. Поля сущности - Location

Persons - Информация о клиентах, бронирующих номер отеля

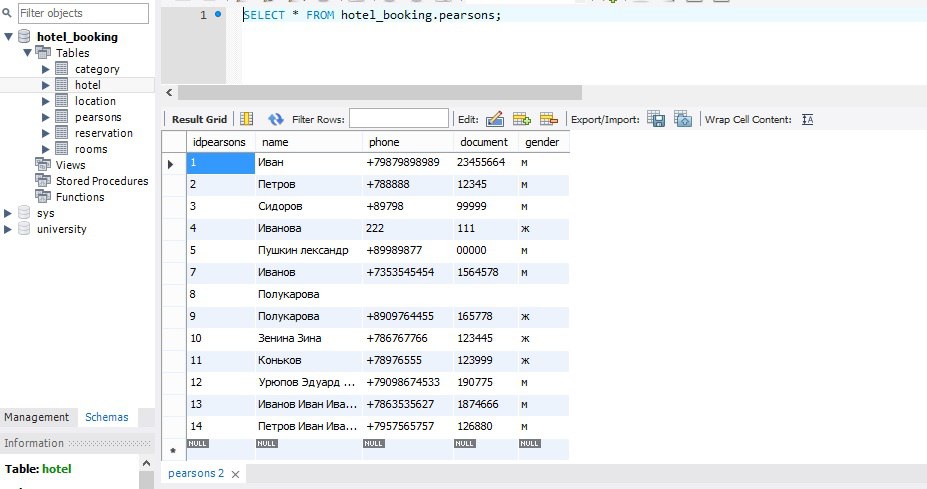


Рис. 2.3.5. Поля сущности - Persons

Reservation - информация о бронировании клиентом номера

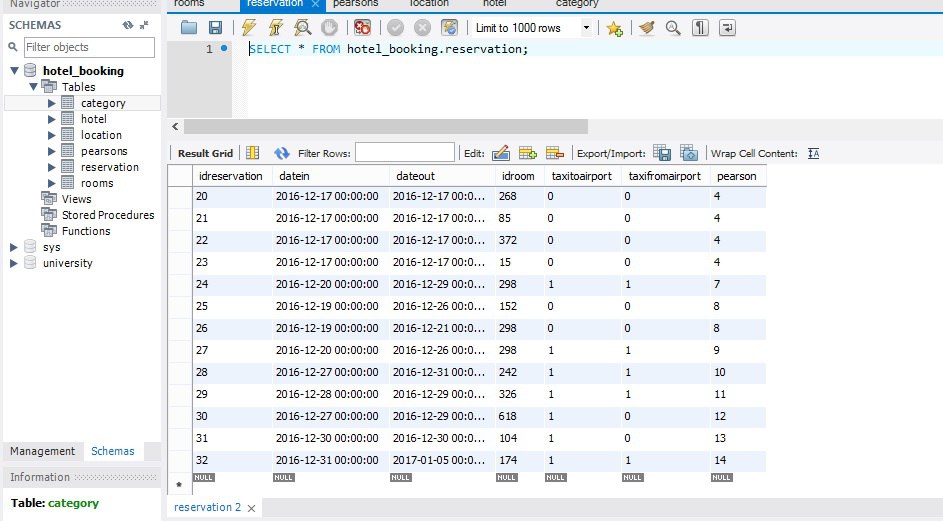


Рис. 2.3.6. Поля сущности - Reservation

Rooms - Информация о свободным номерах в отеле

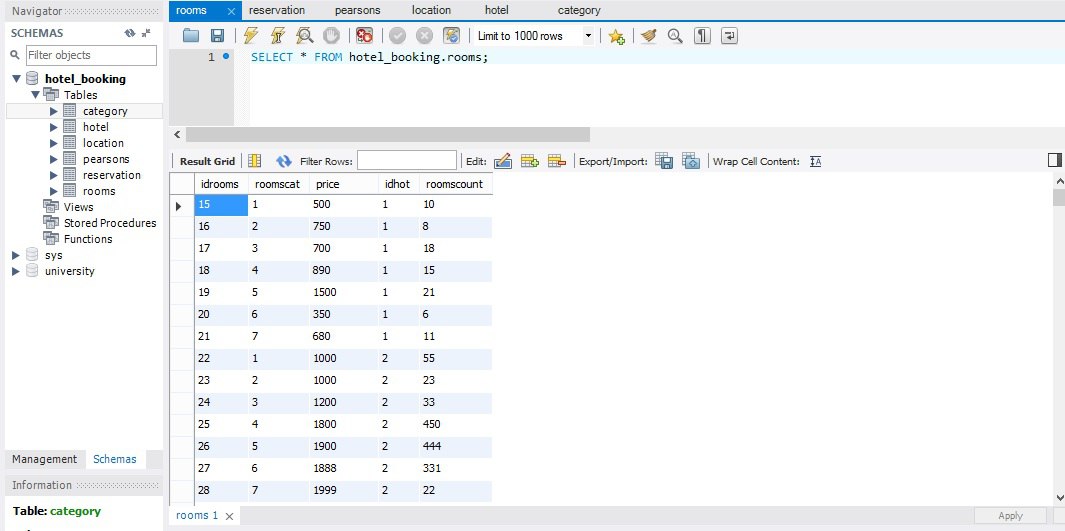


Рис. 2.3.7. Поля сущности - Rooms

Информация извлекается из базы данных при помощи запросов в коде программы, перед этим производя поиск по базе, а также группируя сущности исходя из вида запрашиваемой информации при помощи inner join по ID нужной сущности.

1. **Структура приложения**

*3.2 Описание классов*

Класс“datain.cs” является основным классом для подключения к базе данных SQL и получения всей нужной информации запрашиваемой пользователем. Он включает в себя два метода:

* “Connect” – Метод, использующийся для подключения к базе данных SQL.
* “Command” - Метод для обращения к базе данных SQL

Класс “Form1.cs” является основным классом программы, он включает в себя такие методы как:

* “AddComponents” - Подгружает все доступные элементы из базы данных, при загрузке приложения, в Combobox .
* “button1\_Click” - Считывает информацию, введенную пользователем, из ComboBox, TextBox и dateTimePicker.
* Textbox1, textbox2- поля для ввода ценового диапазона.
* Combobox1, Combobox2 - поля для выбора города и категории отеля.
* DatetimePicker1, DatetimePicker2 - поля для выбора дат выезда и заезда.

Класс “Form2.cs” представляет собой окно с найденной в базе данных информацией, что вводил пользователь в “Form1.cs”, в нем присутствуют методы:

* “button1\_Click” - Возвращает пользователя к первоначальному окну программы.
* “button2\_Click” - При нажатии на кнопку происходит наследование всей информации из окна класса “Form1.cs”, а также из окна “Form.2.cs” в окно “Form3.cs” для возможности пользователя забронировать нужный отель, без ввода ранее выбранной информации.
* “tabControl1\_Selected” - метод, не позволяющий пользователи делать дальнейшие действия с программой (бронировать номер) если не выбран отель, либо не найдено ничего подходящего в базе данных.

Класс “Form3.cs” является конечным для работы с пользователем окном, предоставляющий доступ к непосредственно бронированию номера, с некоторыми дополнительными опциями. Он содержит в себе метод:

* “button1\_Click” - Считывает информацию, введенную пользователем, из ComboBox, TextBox и dateTimePicker, также осуществляется загрузка ранее введенной информции.
* Textbox1- поле для ввода Ф.И.О.
* Textbox2-поле для ввода мобильного телефона.
* Textbox3- поле для ввода серии и номера паспорта.
* Combobox1- пол для выбора пола.
* Сheckbox1, checkbox2-поля для выбора трансфера в отель или из отеля.

1. **Контрольный пример и описание результатов**

Открыть программу Visual Studio. Запуск разработанного приложения производится путем однократного нажатия мыши на кнопку “Запуск”, расположенную в программе Visual Studio сверху. После запуска программы пользователю откроется первое окно, куда он сможет ввести информацию о городе, категории номера, о желаемых датах заезда, а также ценовой диапазон номера (см рис 4.1).

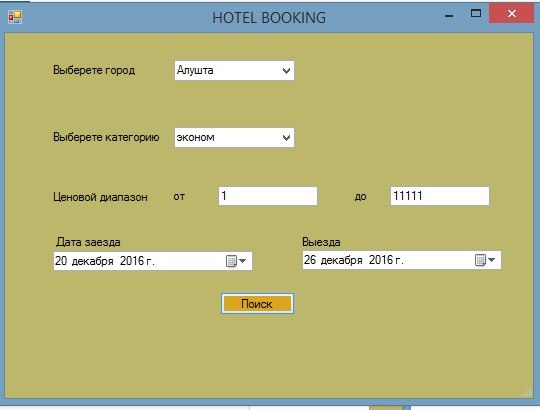


Рис. 4.1. Окно пользователя с введенными им данными.

После того как пользователь ввел все необходимые данные, происходит поиск по базе данных SQL всех возможных вариантов бронирования номером отелей, исходя из выбранных пользователем параметров. Приложение открывает вторую страницу, куда выводятся отели, удовлетворяющие запросу (см. Рис. 4.2.)

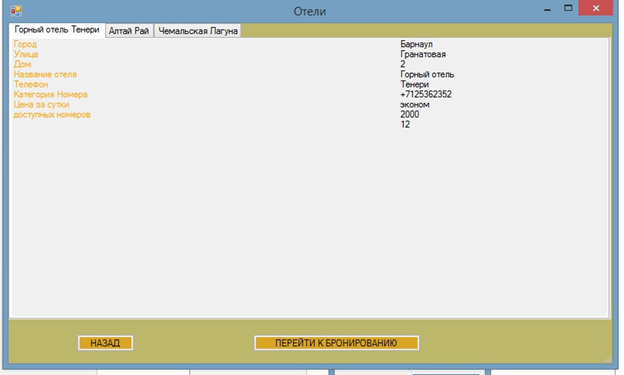


Рис.4.2. Окно выбора отеля для бронирования отеля.

Как только пользователь выбрал отель из всех возможных вариантов и нажал кнопку бронирования, вплывает последнее окно программы. Пользователь вводит свои личные данные, для того чтобы они были занесены в базу данных, если этот пользователь ранее производил процесс бронирования, то данные о нем не будут записываться в базу, если же наоборот, то всплывет окно, говорящее о том, что он успешно был добавлен в базу и его номер забронирован (см. рис. 4.3, 4.4, 4.5)

Когда пользователь введет некорректные данные, программа высветит сообщение об ошибке. После чего пользователь сможет внести данные правильно без повторного запуска программы. ( см. рис. 4.6).

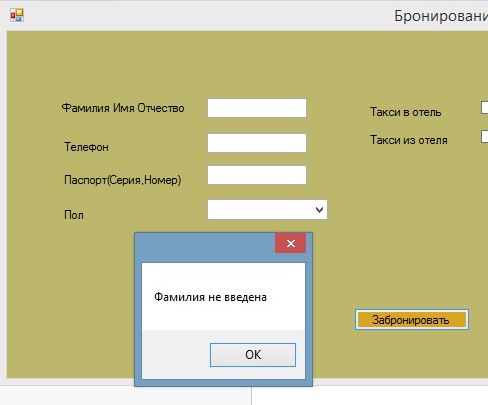


Рис. 4.6. Проверка на корректность введенных данных.

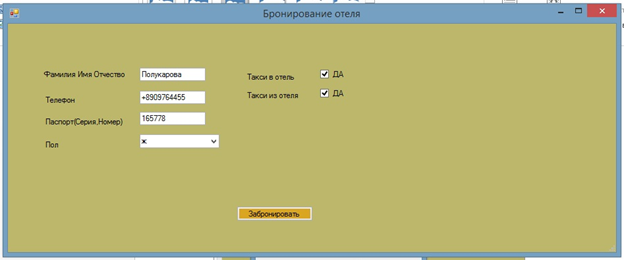


Рис. 4.3. Окно с введенными личными данными пользователя.

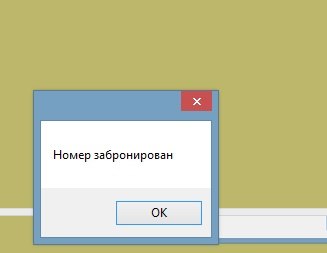
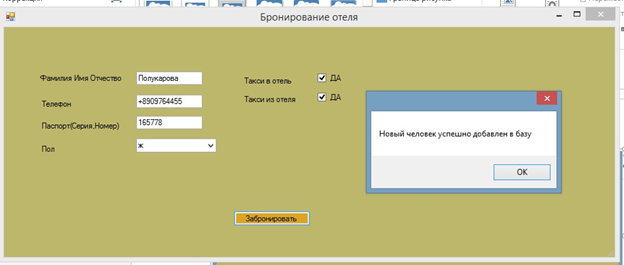


Рис. 4.4, 4.5 Успешно завершенное бронирование отеля.