федеральное государственное бюджетное образовательное учреждение высшего образования

**«Вологодский государственный университет»**

Институт математики, естественных и компьютерных наук

(наименование института)

Кафедра автоматики и вычислительной техники

(наименование кафедры)

**КУРСОВОЙ ПРОЕКТ**

Дисциплина: Объектно – ориентированное программирование

Наименование темы: Разработка приложения «Гостиница»

Код работы КП 09.03.02 001 2021

|  |  |
| --- | --- |
| Руководитель | канд. техн. наук, доцент Сазонова Г.А. |
| Выполнил студент | Иванов Денис Геннадьевич |
| Группа | 4Б09 ИТ-31 |
| Дата сдачи |  |
| Дата защиты  Оценка по защите | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(подпись преподавателя)* |

Вологда

2021 г.

Вариант 1

**Задание на курсовую работу**

**по дисциплине «Объектно-ориентированное программирование»**

студенту Иванову Денису

**1.Тема.**

**Разработка приложения «Гостиница».**

**2.Описание задачи.**

Создать приложение «Гостиница». База данных состоит из нескольких таблиц (не меньше четырех). Таблицы необходимо создать и хранить в формате MS Access. Приложение должно обеспечивать добавление, удаление и изменение данных.

В бизнесе участвуют**:** клиенты (фамилия, имя, отчество, паспортные данные, место регистрации), номера (тип, размещение на этаже, характеристика, стоимость одного дня и т.д.). Клиент может забронировать номер заранее. На каждый заказ оформляется квитанция, в которой указывается клиент, номер с характеристиками, количество дней проживания, общая сумма оплаты. У администратора имеется информация о занятости номеров, их бронировании, количестве проживающих в них людей и т.д.

**Функции приложения:**

1.Подсчет оплаты за конкретный номер.

2.Поиск клиента по различным характеристикам.

3.Проверка наличия свободного номера.

4.Бронирование конкретного номера.

5.Оформление квитанции.

6.Нахождение стоимости всех заказов за конкретный период.

7.Составление отчетов по типам номеров с указанием количества проживаний за указанный период, общей суммы и т.д.

**3.Исходные данные к работе.**

а) наличие пользовательского интерфейса приложения (меню, подсказки, диалоговые окна, оформленные кнопки, заставка приложения);

б) среда разработки ИЛМ – ER WIN.

в) состав и структуру объектов базы данных определить самостоятельно.

г) оформление курсовой работы выполнить в соответствии с методическими указаниями и правилами оформления курсовых проектов и работ.

**4.Содержание курсовой работы; перечень вопросов, подлежащих разработке.**

а) описать предметную область, разработать формы документов предметной области.

б) разработать информационно-логическую модель предметной области в среде ER WIN: построить диаграмму сущность-связь; модель данных, основанную на ключах; полную атрибутивную модель; физическую модель БД.

в) разработать удобный интерфейс пользователя.

г) разработать программное обеспечение.

д) разработать руководство пользователя и руководство программиста.

**5.Перечень иллюстративных материалов.**

а) пояснительная записка к курсовой работе с необходимыми таблицами и рисунками.

б) компьютерная реализация разработанного приложения.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc84771200)

[1 ПОСТАНОВКА ЗАДАЧИ 6](#_Toc84771201)

[1.1 Характеристика предметной области и описание документов 6](#_Toc84771202)

[1.2 Требования к приложению 7](#_Toc84771203)

[2 РАЗРАБОТКА ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ ПРИЛОЖЕНИЯ 9](#_Toc84771204)

[3 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 14](#_Toc84771205)

[3.1 Описание экранных форм 14](#_Toc84771206)

[3.2 Разработка приложения 16](#_Toc84771207)

[4 ДОКУМЕНТАЦИЯ ПО РАБОТЕ ПРИЛОЖЕНИЯ 19](#_Toc84771208)

[4.1 Руководство пользователя 19](#_Toc84771209)

[4.2 Руководство программиста 24](#_Toc84771210)

[ЗАКЛЮЧЕНИЕ 26](#_Toc84771211)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 27](#_Toc84771212)

[ПРИЛОЖЕНИЕ 28](#_Toc84771213)

# ВВЕДЕНИЕ

Информационные технологии содержатся практически во всех сферах человеческой жизни. Вообще, что представляют собой информационные технологии? ИТ – это различные методы, способы, которые предназначены для поиска, хранения, обработки информации. 21 век – век информационного общества. Именно информация даёт человеку неоспоримое преимущество перед другими существами.

Важным понятием, которое относится к ИТ является информационная система. ИС – совокупность техническим, программных средств для хранения и обработки информации. Как мы видим, эти два понятия очень похожи.

Плавно перетекаем к понятию баз данных. Далее БД. Что такое БД? БД – это совокупность данных, которые хранятся согласно какой-либо логической схеме. Для успешного ведения БД, как правило, необходимо наличие программного обеспечения, технического обеспечения, информационного обеспечения, математического обеспечения.

Начнём с технического обеспечения. К ТО относится в общем электроника, которая функционирует по законам электрических токов. Именно электрическая физика позволяет решать многое. На экране монитора мы видим данные, а внутри железа через множество уровней абстракции осуществляют движение электроны по заданным алгоритмам. Похожее творится и в человеческом мозге.

Программное обеспечение уже позволяет человеку взаимодействовать с внутренней физикой. Это программы системного и прикладного характера.

Царица всех наук – математика. Это инструмент, с помощью которого строятся алгоритмы обработки процессов, информации.

К информационному обеспечению можно отнести документацию, знания о предметной области.

Наша информационная система, база данных будет реализована с помощью программного продукта прикладного характера под названием Builder.

C++ Builder – это большая среда разработки, которая позволяет создавать программы различных уровней сложности. Разобраться с ней несложно, это говорит о удачно созданном интерфейсе взаимодействия. Можно выделить условно два фронта работы: компонентная и программная.

С помощью компонентов реализуются какие-либо объекты – графические либо же неграфические, но выполняющие различные цели.

Основная же работа программиста – написание программного кода.

На основе того, что было написано выше и создаётся база данных.

Курсовой проект содержит в себе 4 раздела.

В первом разделе описывается постановки задачи. Здесь формулируются конкретные цели и задачи для данной работы, дается характеристика предметной области и описание документов.

Во втором разделе идёт описание разработки информационного обеспечения приложения (т.е. анализируется состав информации, устанавливаются функциональные зависимости реквизитов).

Далее в третьем разделе разрабатывается программное обеспечение, описываются все экранные формы этого приложения.

В последнем разделе прописывается руководство пользователя и программиста.

# 1 ПОСТАНОВКА ЗАДАЧИ

В данном курсовом проекте необходимо разработать логическую и физическую модель реляционной БД в программной среде создания БД – СУБД MS Access и само приложение “Гостиница” в среде C++ Builder.

Необходимо разработать, то приложение, которое детально отразит предметную область и поможет автоматизировать все рутинные процессы.

При этом нужно учитывать множество факторов данного направления. Имеются клиенты, значит приложение должно содержать подробную информацию о них.

Клиенты заказывают номера, значит нужно хранить и обрабатывать информацию о номерах, реализовать функции поиска.

Сам процесс бронирования должен быть автоматизирован. Необходимо сделать так, чтобы администратор не занимался теми делами, которые можно автоматизировать.

## 1.1 Характеристика предметной области и описание документов

Предметная область: гостиница.

Необходимо разработать приложение, которое автоматизирует предприятие по оказанию услуг в гостиничной сфере.

Можно выделить пять основных документов, которые и определяют структуру программного продукта.

Вот список документов: клиенты, номера, типы номеров, бронирование, квитанция.

Форма документа «Клиенты» представлена на рисунке 1.1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Код клиента** | **ФИО клиента** | **Серия паспорта** | **Номер паспорта** | **Место регистрации** |
|  |  |  |  |  |

Рисунок 1.1 – Форма документа «Клиенты»

Форма документа «Номера» представлена на рисунке 1.2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Код клиента | Код типа номера | Номер этажа | Характеристика | Стоимость 1 дня |
|  |  |  |  |  |

Рисунок 1.2 – Форма документа «Номера»

Форма документа «Типы номеров» представлена на рисунке 1.3.

|  |  |
| --- | --- |
| Код типа номера | Наименование типа |
|  |  |

Рисунок 1.3 – Форма документа «Типы номеров»

Форма документа «Бронирование» представлена на рисунке 1.4.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Код бронирования | Код клиента | Код номера | Дата начала | Дата окончания | Количество человек |
|  |  |  |  |  |  |

Рисунок 1.4 – Форма документа «Бронирование»

Форма документа «Квитанция» представлена на рисунке 1.5.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Код бронирования | Код клиента | Код номера | ФИО клиента | Стоимость 1 дня | Дата начала | Дата окончания | Кол-во дней | Итого |
|  |  |  |  |  |  |  |  |  |

Рисунок 1.5 – Форма документа «Квитанция»

## 1.2 Требования к приложению

Разрабатываемое программное обеспечение должно решать много функций. Необходимо содержание полной информации о клиенте, а именно фамилию, имя, отчество, различные паспортные данные: серия паспорта, номер паспорта, а также пункт выдачи этого важного документа.

Далее приложение должно содержать подробную характеристику о всех номерах предприятия. К характеристике можно отнести код номера, номер этажа, текстовую характеристику, а также тип самого номера от бюджетного до VIP.

Программа должна обладать возможностью добавления типа номера. Ведь предприятие не стоит на месте, а развивается. Этим и обуславливается увеличение количества типов номеров.

Далее разрабатываемая программа должна подробного отражать сам процесс бронирования, а также как уже было сказано выше производить простые и рутинные расчёты. Эта информация доступна администратору организации. Нужен поиск необходимой информации по сущности бронирования.

После того, как процесс бронирования номера завершился, необходимо, чтобы программа печатала квитанцию, в которой отражена необходимая клиенту информация. Также есть необходимость в организации отчётных процессов.

Всё то, что описано выше – это относится к самому программному продукту. Давайте теперь перейдём к техническому обеспечению.

В первую очередь необходим сам компьютер: системный блок со всем стандартным “железом”.

Нужно устройство вывода: монитор. Устройство ввода, периферия: клавиатура, мышь, динамики.

К системному программному обеспечению можем отнести наличие операционной системы Windows.

# 2 РАЗРАБОТКА ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ ПРИЛОЖЕНИЯ

Отразим структуру информационного обеспечения. Состав информационного обеспечения отражён на таблице 2.1 ниже.

Таблица 2.1 – Состав информационного обеспечения

|  |  |  |
| --- | --- | --- |
| Название информационного объекта | Обозначение ИО | Семантика ИО |
| Клиенты | Клиенты | Сведения о клиентах |
| Номера | Номера | Сведения о номерах |
| Типы номеров | Типы номеров | Сведения о типах номеров |
| Бронирование | Бронирование | Сведения о бронировании |
| Квитанция | Квитанция | Сведения о квитанции |

Далее необходимо отразить все функциональные зависимости между сущностями базы данных. Они показаны на таблице 2.2.

Таблица 2.2 – Функциональные зависимости реквизитов

|  |  |  |  |
| --- | --- | --- | --- |
| Информационный объект | Название реквизитов | Имя реквизитов | Функциональные зависимости |
| 1 | 2 | 3 | 4 |
| Клиенты | Код клиента  ФИО  Серия паспорта  Номер паспорта  Место регистрации | КК  ФИОК  СПАС  НОМПАС  МЕСРЕГ |  |
| Номера | Код номера  Код типа номера  Номер этажа  Характеристика  Стоимость 1 дня | КНОМ  КТНОМ  НЭТ  ХАР  СТ1Д |  |
| Типы номеров | Код типа номера  Наименование типа номера | КТНОМ  НТНОМ |  |
| Бронирование | Код бронирования  Код клиента  Код номера  Дата начала  Дата окончания  Количество человек | КБР  КК  КНОМ  ДНАЧ  ДОКОН  КОЛЧЕЛ |  |
|  |  |  |  |

Продолжение таблицы 2.2

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Квитанция | Код бронирования  Код клиента  Код номера  ФИО клиента  Стоимость одного дня  Дата начала  Дата окончания  Кол-во дней  Итого | КБР  КК  КНОМ  ФИОК  СТ1Д  ДНАЧ  ДОКОН  КОЛДН  ИТОГО |  |

Приведём соответствие описательных и ключевых реквизитов. Оно на таблице 2.3.

Таблица 2.3 – Соответствие описательных и ключевых реквизитов

|  |  |  |  |
| --- | --- | --- | --- |
| Описательные реквизиты | Ключевые реквизиты | Вид ключа | Имя ИО, включающего реквизит |
| Документ «Клиенты» | | | |
| ФИОК | КК | П., У. | Клиенты |
| СПАС | КК | П., У. | Клиенты |
| НОМПАС | КК | П., У. | Клиенты |
| МЕСРЕГ | КК | П., У. | Клиенты |
| Документ «Номера» | | | |
| КТНОМ | КНОМ | П., У. | Типы номеров |
| НЭТ | КНОМ | П., У. | Номера |
| ХАР | КНОМ | П., У. | Номера |
| СТ1Д | КНОМ | П., У. | Номера |
| Документ «Типы номеров» | | | |
| НТНОМ | КТНОМ | П., У. | Типы номеров |
| Документ «Бронирование» | | | |
| КК | КБР | П., У. | Клиенты |
| КНОМ | КБР | П., У. | Номера |
| ДНАЧ | КБР | П., У. | Бронирование |
| ДОКОН | КБР | П., У. | Бронирование |
| КОЛЧЕЛ | КБР | П., У. | Бронирование |
| Документ «Квитанция» | | | |
| КК | НЕТ | НЕТ | Квитанция |
| КНОМ | НЕТ | НЕТ | Квитанция |
| СТ1Д | НЕТ | НЕТ | Квитанция |
| ДНАЧ | НЕТ | НЕТ | Квитанция |
| ДОКОН | НЕТ | НЕТ | Квитанция |
| КОЛД | НЕТ | НЕТ | Квитанция |
| ИТОГО | НЕТ | НЕТ | Квитанция |
| КБР | НЕТ | НЕТ | Квитанция |

Разработаем и построим связи информационных объектов. Связи информационных объектов представлены в соответствии с таблицей 2.4.

Таблица 2.4 – Связи информационных объектов

|  |  |  |  |
| --- | --- | --- | --- |
| № связи | Главный ИО | Подчиненный ИО | Тип реального отношения |
| 1 | Бронирование | Клиенты | 1:М |
| 2 | Бронирование | Номера | 1:М |
| 3 | Номера | Типы номеров | 1:М |
| 4 | Квитанция | Бронирование | 1:М |

Разработаем и построим логическую модель данных в ER WIN. Для этого необходимо добавить на рабочую область 5 сущностей. Назвать их соответственно, добавить ключи и атрибуты. Полная атрибутивная модель, модель сущность – связь и модель данных, основанная на ключах представлены на рисунках 2.1, 2.2 и 2.3.

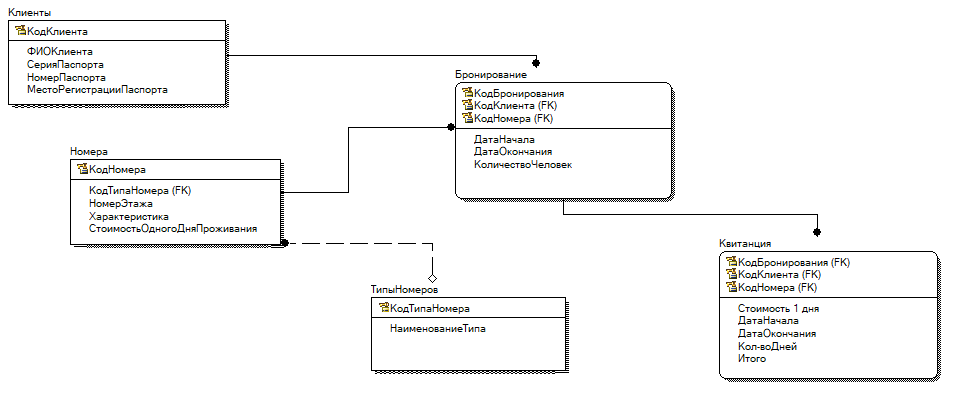


Рисунок 2.1 – Полная атрибутивная модель

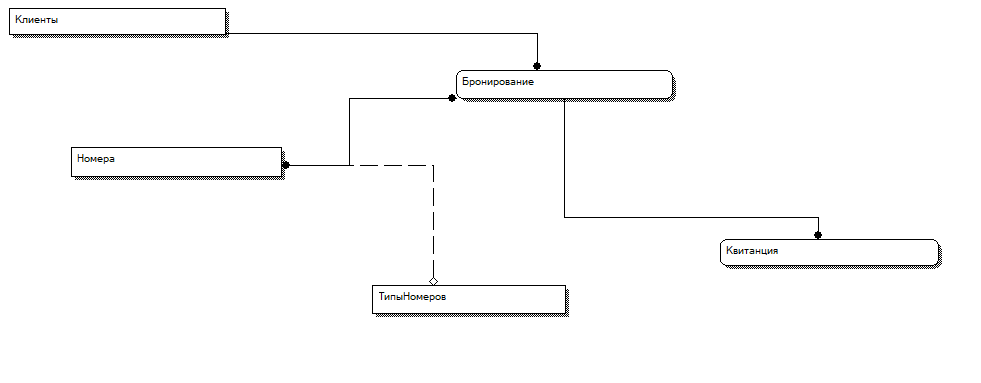


Рисунок 2.2 – Модель сущность – связь

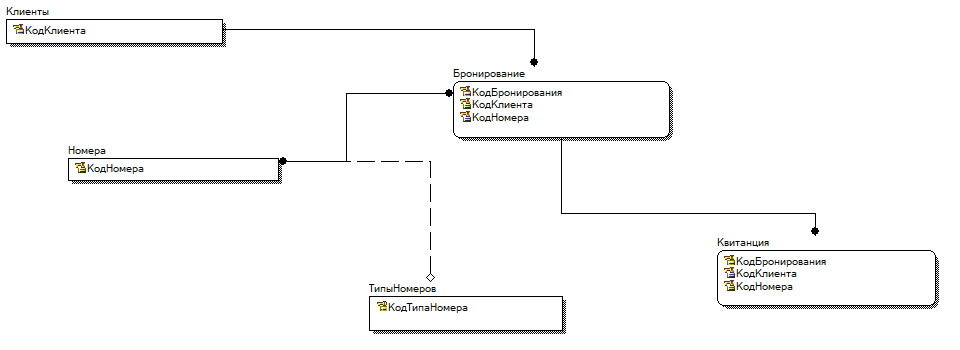


Рисунок 2.3 – Модель, основанная на ключах

Теперь необходимо привести полное описание всех таблиц реляционной базы данных. Сюда же относится и описание всех реквизитов и атрибутов.

Описание таблиц реляционной базы данных представлено в таблице 2.5.

Таблица 2.5 – Описание таблиц реляционной базы данных

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Атрибут | | Признак ключа | Формат поля | | |
| Обозначение | Наименование | Тип | Длина | Точность |
| 1 | 2 | 3 | 4 | 5 | 6 |
| ИО <Клиенты> | | | | | |
| ФИОК | П., У. | Клиенты | Короткий текст | 30 |  |
| СПАС | Клиенты | Числовой | Длинное целое |
|  |  |  |  |  |  |

Продолжение таблицы 2.5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |
| НОМПАС |  | Клиенты | Числовой | Длинное целое |  |
| МЕСРЕГ | Клиенты | Короткий текст | 30 |  |
| ИО <Номера> | | | | | |
| КТНОМ | П., У. | Типы номеров | Числовой | Длинное целое |  |
| НЭТ | Номера | Числовой | Длинное целое |
| ХАР | Номера | Длинный текст |  |
| СТ1Д | Номера | Числовой | Длинное целое |  |
| ИО <Типы номеров> | | | | | |
| НТНОМ | П., У. | Типы номеров | Короткий текст | 30 |  |
| ИО <Бронирование> | | | | | |
| КК | П., У. | Клиенты | Числовой | Длинное целое |  |
| КНОМ | Номера | Числовой | Длинное целое |  |
| ДНАЧ | Бронирование | Дата и время |  |  |
| ДОКОН | Бронирование | Дата и время |  |  |
| КОЛЧЕЛ | Бронирование | Числовой | Длинное целое |  |
| ИО <Квитанция> | | | | | |
| КК | П., У. | Клиенты | Короткий текст | 30 |  |
| КНОМ | Номера | Числовой | Длинное целое |  |
| СТ1Д | Номера | Числовой | Длинное целое |  |
| ДНАЧ | Бронирование | Дата и время |  |  |
| ДОКОН | Бронирование | Дата и время |  |  |
| КОЛД | Квитанция | Числовой | Длинное целое |  |
| ИТОГО | Квитанция | Числовой | Длинное целое |  |
| КБР | Бронирование | Числовой | Длинное целое |  |

# 3 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

## 3.1 Описание экранных форм

Когда программа “Гостиница” включается пользователя приветствует форма симпатичная заставка приложения. Эта форма показана на рисунке 3.1.

image

button

Рисунок 3.1 – Заставка приложения

Далее будет открыта основная форма, на которой располагаются все элементы управления. Текущая вкладка – клиенты. Данная форма представлена на рисунке 3.2.

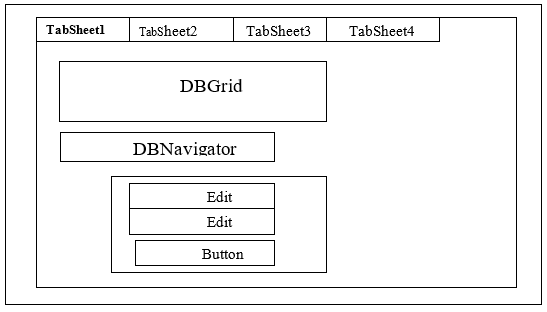


Рисунок 3.2 – Форма управление

На второй вкладке располагается информация о номерах. Форма клиенты представлена на рисунке 3.3.

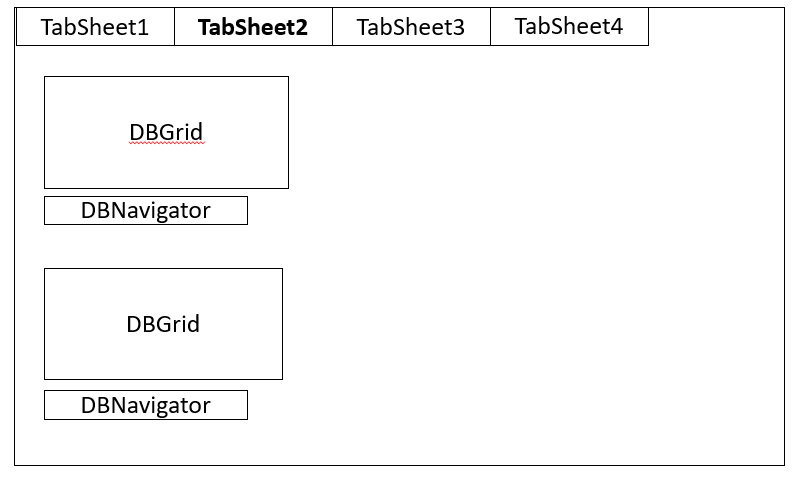


Рисунок 3.3 – Форма «Номера»

Перейдём к третьей вкладке. На ней расположился интерфейс по управлению процессом бронирования. Форма представлена рисунке 3.4.

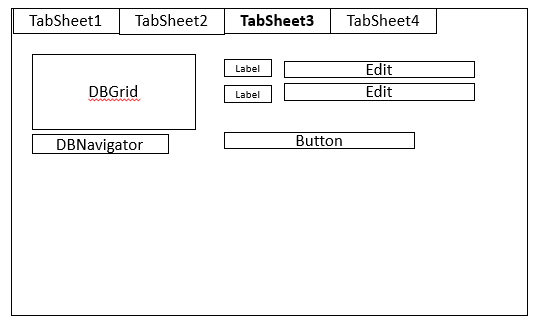


Рисунок 3.4 – Форма «Бронирование»

И последняя вкладка квитанции. На ней также содержатся дополнительные функции. Форма представлена рисунке 3.5.

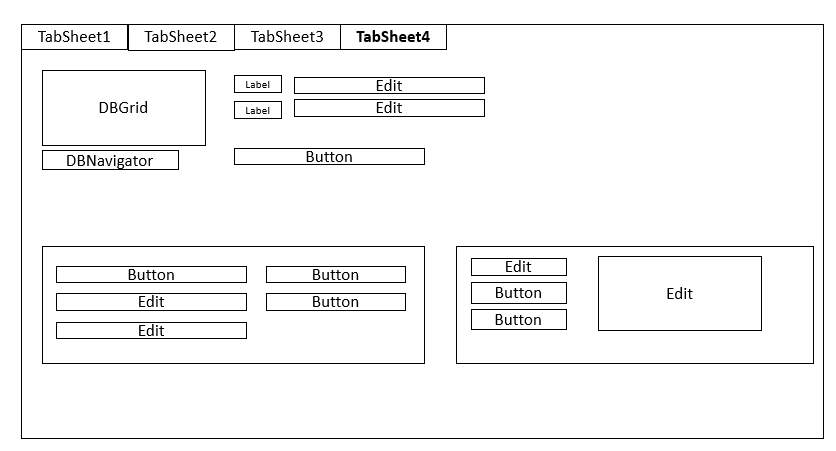


Рисунок 3.5 – Форма «Квитанция»

## 3.2 Разработка приложения

Какие же компоненты были использованы при создании БД? На данный вопрос можно получить ответ по таблице 3.1.

Таблица 3.1 – Основные компоненты, используемые в приложении

|  |  |  |
| --- | --- | --- |
| Компонент | Свойство | Описание свойства |
| 1 | 2 | 3 |
| Button | Caption | Название кнопки |
| Font | Определяет шрифт для надписей внутри объекта |
| Hint | Текст подсказки, которая всплывает при наведении указателя мыши на объект. |
| Label | Caption | Название подписей |
| Font | Определяет шрифт для надписей внутри объекта |
| Hint | Текст подсказки, которая всплывает при наведении указателя мыши на объект. |
| Image | Width | Ширина картинки |
| Height | Высота картинки |
| Picture | Вставка изображения |
| DBGrid | DataSource | Служит для связи с компонентом DataSource |
| ADOConnection | ConnectionString | Служит для связи БД из Access |
|  |  |  |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
|  | LoginPrompt | Требуется ли ввод пароля |
| Connected | Подключена ли база данных |
| ADOTable | Connection | Служит для связи с компонентом ADOConnection |
| TableName | Служит для связи определённых таблиц |
| DataSource | DataSet | Служит для связи с компонентом ADOTable |
| DBNavigator | DataSource | Связывает с таблицей для дальнейшего изменения данных в таблице |
| ADOQuery | ConnectionString | Служит для связи БД из Access |
| QuickRep | DataSet | Служит для связи с компонентом ADOQuery |
| GroupBox | Caption | Название подписей |
| Font | Определяет шрифт для надписей внутри объекта |
| Hint | Текст подсказки, которая всплывает при наведении указателя мыши на объект. |
| Edit | Font | Определяет шрифт для надписей внутри объекта |
| DBEdit | DataSource | Служит для связи с компонентом DataSource |
| DataField | Связывает с определенным столбцом из таблицы |
| ComboBox | Items | Хранит все названия таблиц |

В программном продукте “Гостиница” были реализованы некоторые функции. Их описание приведено в таблице 3.2.

Таблица 3.2 – Реализация функций приложения

|  |  |  |
| --- | --- | --- |
| № п.п. | Функции приложения | Описание реализации функций приложения |
| 1 | 2 | 3 |
| 1 | Подсчет оплаты за конкретный номер. | Подсчёт оплаты происходит следующим образом: имеется таблица бронирование, номера, клиенты. Создаётся SQL запрос, который проходит по всем таблицам и берёт нужные данные, а также на основе полученных данных производит расчет оплаты за конкретный номер. |
| 2 | Поиск клиента по различным характеристикам. | Данная функция реализована на вкладке клиенты. Имеется два текстовых поля и одна кнопка. В первое поле вводится название столбца, а во второе требуемое значение атрибута. При нажатии кнопки происходит процесс поиска. |
|  |  |  |

Продолжение таблицы 3.2

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 3 | Проверка наличия свободного номера. | Это очень интересная функция. Она реализована на третьей вкладке бронирования. Алгоритм при добавлении новой брони анализирует информацию в таблице, а именно даты, чтобы не было пересечений. Если они будут программа выдаст ошибку. |
| 4 | Бронирование конкретного номера. | Тут всё просто. Реализация на 3 вкладке. |
| 5 | Оформление квитанции. | Квитанция оформляется на основе бронирования. Имеется возможность вывода на печать. Главное указать номер необходимой брони. |
| 6 | Нахождение стоимости всех заказов за конкретный период. | Кнопка нахождения стоимости заказов за конкретный период находится на последней вкладке. Там же имеется два текстовых поля типа дата. Алгоритм реализован на основе цикла. |
| 7 | Составление отчетов. | Для составления отчетов необходимо добавить компоненты QuickRep. Составляются отчеты в QuickRep, добавляем в него все необходимые надписи. Далее в свойстве SQL создаем вычисляемое поле, в котором извлекаем из столбца. |

# 4 ДОКУМЕНТАЦИЯ ПО РАБОТЕ ПРИЛОЖЕНИЯ

## 4.1 Руководство пользователя

И так для того, чтобы открыть приложение необходимо найти его исполняемый файл или просто ярлык на него, нажать на него правой кнопкой мыши и открыть, можно двойным щелчком.

После открытия программы появится заставка приложения. Вид её изображён на рисунке 4.1.

Нажимаем “Далее”.

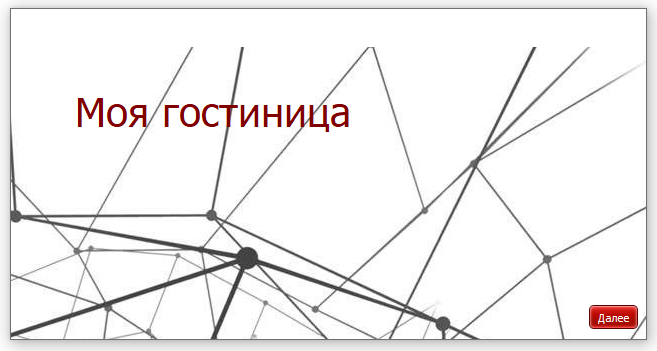


Рисунок 4.1 – Заставка приложения

После нажатия кнопки далее попадаем в само приложение. Мы видим, на рабочей области располагаются элементы управления. Можем перемещаться по различным навигационным вкладкам. Каждая вкладка содержит в себе определённую таблицу.

Как выглядит рабочая область можно посмотреть на рисунке 4.2.

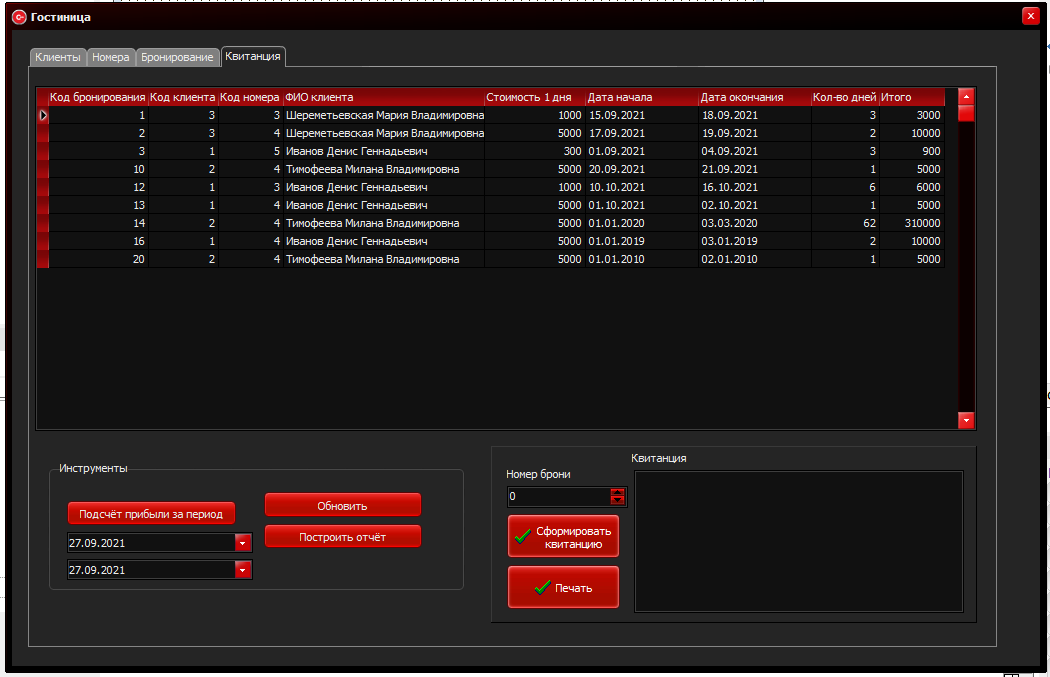


Рисунок 4.2 – Квитанция

На вкладке квитанция мы полную информацию о бронировании. Именно по этой информации формируется квитанция.

Чуть ниже на панели инструментов можно воспользоваться тремя функциями:

* Подсчёт прибыли за определённый период;
* Обновление;
* Построение отчёта;

Что касается подсчёта прибыли за определённый период, то тут всё просто: выбираем необходимый диапазон дат на полях и нажимает соответствующую кнопочку. Прибыль отобразится в текстовом сообщении.

Процедура обновления является очень важной процедурой. После проведения процесса бронирования клиента информация попадает в ядро приложения – базу данных Access. Чтобы была показана актуальная информация в данном разделе, её необходимо обновить соответствующей кнопкой. Произведутся все необходимые расчёты.

Важной частью деятельности любой организации является ведение отчётов. Это помогает в подведении итогов, определении дальнейших направлений развития предприятия.

Для создания отчёта нажимается кнопка построения отчёта. Вид самого отчёта предоставлен на рисунке 4.3.



Рисунок 4.3 – Пример отчёта

На рисунке 4.4. предоставлен результат работы функций квитанции и печати. Необходимо выбрать нужный код бронирования.

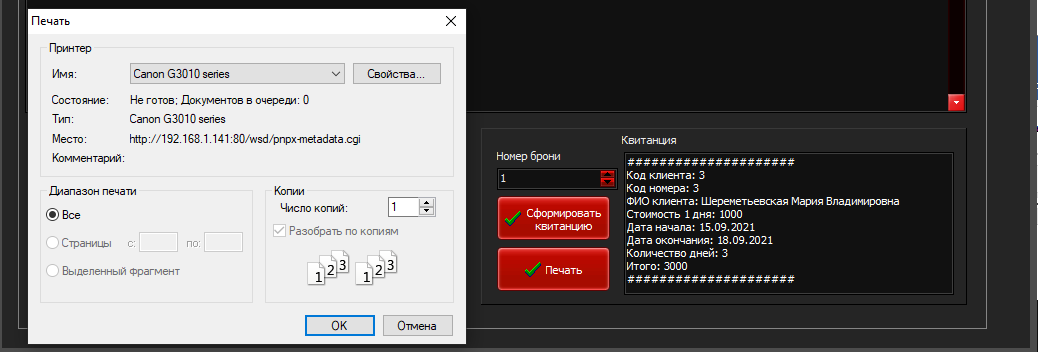


Рисунок 4.4 – Печать

Как мы видим, в текстовом окне показан вид квитанции, а немного левее открылось диалоговое окно печати.

Давайте перейдём к вкладке бронирование. Она показана на рисунке 4.5.

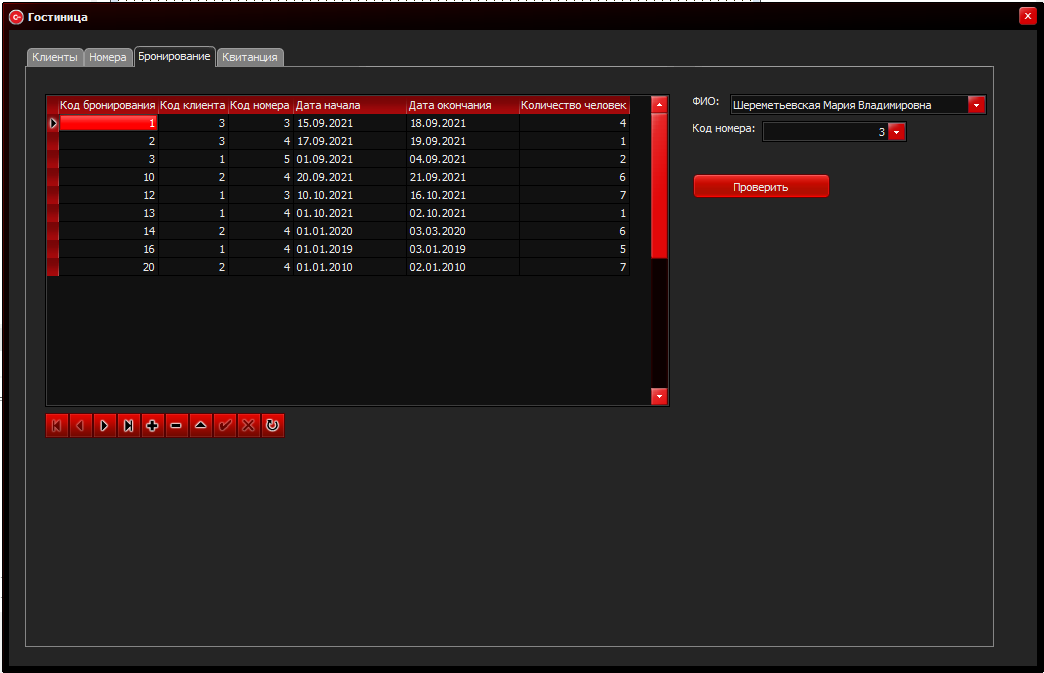


Рисунок 4.5 – Бронирование

Здесь отображается информация административного характера. Приходит клиент, у него имеется индивидуальный код. Далее клиент выбирает нужную комнату, у которой тоже имеется собственный код. По сути, в этом и заключается весь процесс бронирования.

Все кнопки находятся на специальном навигаторе таблиц баз данных. Там и производится добавление, удаление записей.

Для того, чтобы проверить нету ли совпадений по датам проживания необходимо нажать кнопку проверки. Информация о правильности отобразится в текстовом окне (рис. 4.6).

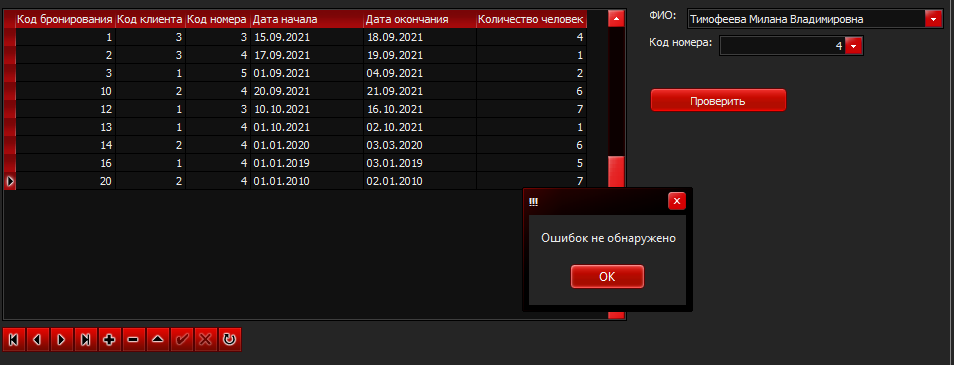


Рисунок 4.5 – Поиск ошибок

Осталось рассмотреть две вкладки номеров и клиентов. У номеров имеется две таблицы: таблица полной информации о номере и типы номеров. Как выглядит рабочая область показано на рисунке 4.6.

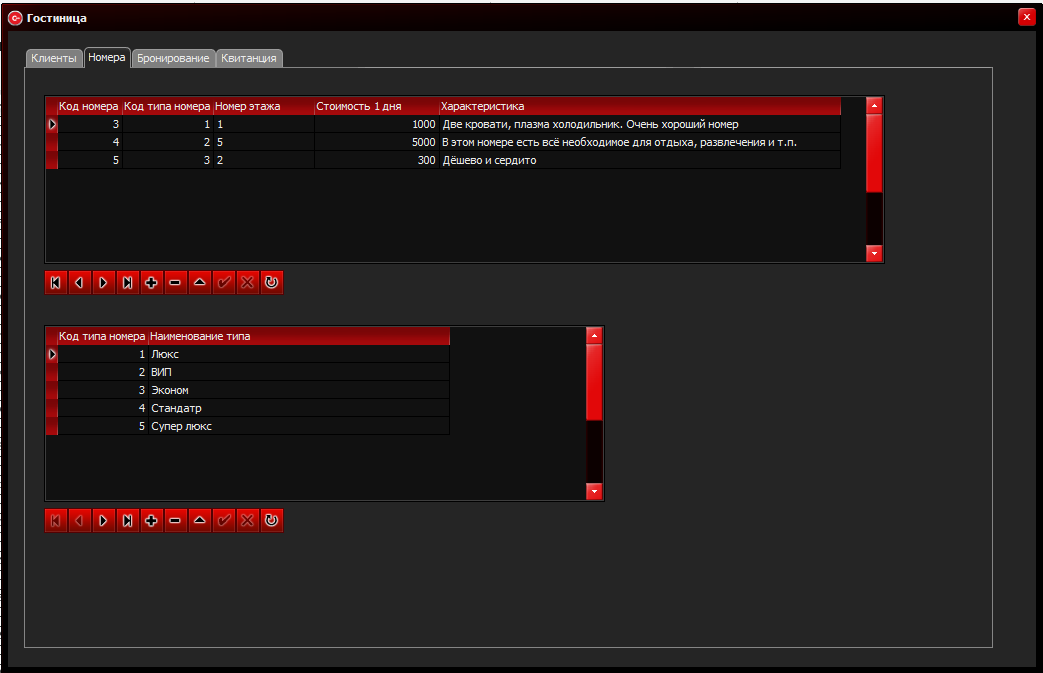


Рисунок 4.6 – Номера

Для редактирования записей таблиц необходимо воспользоваться кнопочными элементами под таблицами.

То же самое относится и к рабочей области клиентов на рисунке 4.7.

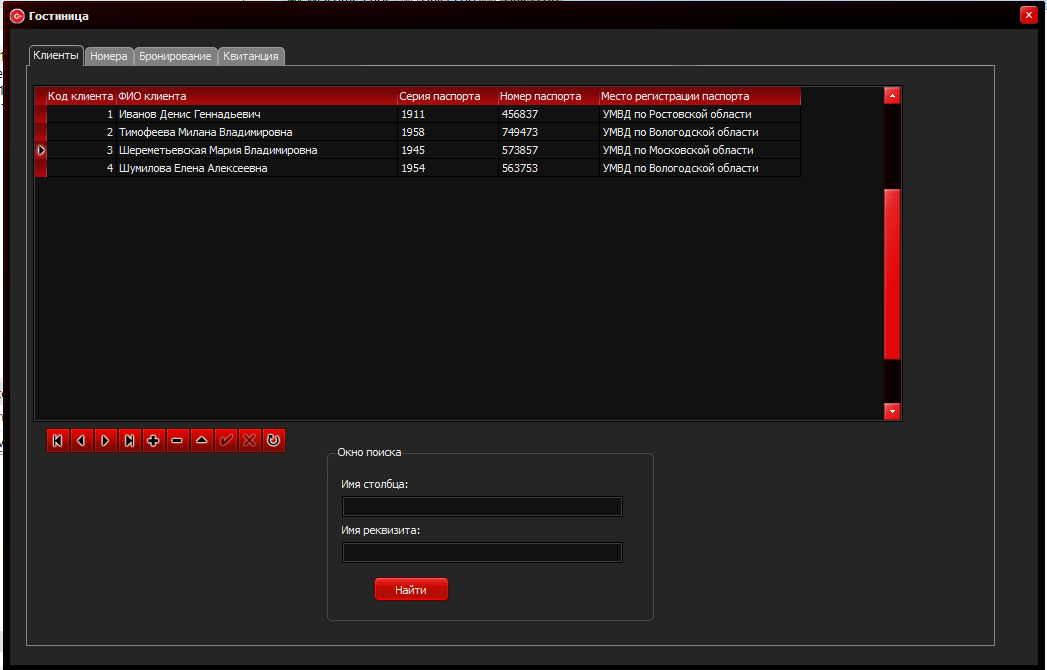


Рисунок 4.7 – Клиенты

Для того, чтобы найти в таблице определённые данные необходимо ввести имя столбца, по которому будет реализовываться поиск и имя атрибута. Информация будет обозначена специальной кареткой слева.

## 4.2 Руководство программиста

Для нормального функционирования приложения необходимо наличие базы данных Access в формате mdb.

После установки программы нужно, чтобы была обеспечена связь с таблицей.

Также необходимо наличие специальных библиотек Windows Visual c++ Redistributable.

Особых предписаний не имеется.

# ЗАКЛЮЧЕНИЕ

В ходе курсового проекта удалось проделать немалый объём работы, в ходе которого был получен ценный опыт по разработки информационной системы предприятия по ведению гостиничного бизнеса.

Были рассмотрены теоретические основы информационных процессов и систем, а также структура средств обеспечения информацией.

Даже были затронуты внутренние физические основы и законы, по которым функционируют информационные системы.

Удалось поработать со средой по созданию информационных систем, а именно C++ Builder.

Также пригодилась и среда Access, case-средство Erwin.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. **Дейл, Н. Программирование на С++: учебник / Н, Дейл. – Спб.: ДМК, 2000. – 673 с.**
2. **Хомоненко, А. Д. Начало С++: учебник / А. Д. Хомоненко. – Санкт-Петербург: БХВ-Петербург, 2006. – 464 с.**
3. **Ренди, Д. С. С++ для чайников: учебник / Д. С. Ренди. –** [Москва](https://www.ozon.ru/publisher/vilyams-856490/) **:** [Вильямс](https://www.ozon.ru/publisher/vilyams-856490/)**, 2018. – 1152 с.**
4. Гарнаев, А. Micrоsоft Office 2000. Разработка приложений /А. Гарнаев, А. Матросов, Ф. Новиков и др.; под ред.Ф. Новикова. - СПб: ВНУ, 2000. – 570 с.
5. Горбаченко В. И. Проектирование информационных систем с CA ERwin Modeling Suite 7.3: учебное пособие / В. И. Горбаченко, Г. Ф. Убиенных, Г. В. Бобрышева – Пенза: Изд-во ПГУ, 2012. – 154 с.

# ПРИЛОЖЕНИЕ

/---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit1.h"

#include "Unit3.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma link "QuickRpt"

#pragma link "QRCtrls"

#pragma resource "\*.dfm"

TForm1 \*Form1;

//---------------------------------------------------------------------------

\_\_fastcall TForm1::TForm1(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::BitBtn1Click(TObject \*Sender)

{

DBGrid4->DataSource->DataSet->Last();

int kol = DBGrid4->DataSource->DataSet->RecordCount;

int number = 0;

TDateTime datetimeBeg = 0;

TDateTime datetimeEnd = 0;

bool status = 1;

int numMis = 0;

for (int i1 = 1; i1 < kol + 1; i1++)

{

DBGrid4->DataSource->DataSet->RecNo = i1;

number = DBGrid4->Columns->Items[2]->Field->AsInteger;

datetimeBeg = DBGrid4->Columns->Items[3]->Field->AsDateTime;

datetimeEnd = DBGrid4->Columns->Items[4]->Field->AsDateTime;

for (int i2 = i1 + 1; i2 < kol + 1; i2++)

{

DBGrid4->DataSource->DataSet->RecNo = i2;

if (number == DBGrid4->Columns->Items[2]->Field->AsInteger)

{

if (

((datetimeBeg >= DBGrid4->Columns->Items[3]->Field->AsDateTime) && (datetimeBeg <= DBGrid4->Columns->Items[4]->Field->AsDateTime))

||

((DBGrid4->Columns->Items[3]->Field->AsDateTime >= datetimeBeg) && (DBGrid4->Columns->Items[3]->Field->AsDateTime <= datetimeEnd))

)

{

status = 0;

numMis = number;

}

if (

((datetimeEnd >= DBGrid4->Columns->Items[3]->Field->AsDateTime) && (datetimeEnd <= DBGrid4->Columns->Items[4]->Field->AsDateTime))

||

((DBGrid4->Columns->Items[4]->Field->AsDateTime >= datetimeBeg) && (DBGrid4->Columns->Items[4]->Field->AsDateTime <= datetimeEnd))

)

{

status = 0;

numMis = number;

}

}

}

}

if (status == 1) {

ShowMessage("Ошибок не обнаружено");

}

else

{

ShowMessage("Есть ошибки");

ShowMessage("Код проблемной комнаты: " + IntToStr(numMis));

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::BitBtn2Click(TObject \*Sender)

{

ADOQuery1->Refresh();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::BitBtn3Click(TObject \*Sender)

{

AnsiString str1,str2;

str1 = Edit1->Text;

str2 = Edit2->Text;

(ADOTable1->Locate(str1,str2,TLocateOptions() << loCaseInsensitive) );

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::BitBtn4Click(TObject \*Sender)

{

DBGrid5->DataSource->DataSet->Last();

int kol = DBGrid5->DataSource->DataSet->RecordCount;

TDateTime datetimeBeg = DateTimePicker1->Date;

TDateTime datetimeEnd = DateTimePicker2->Date;

int Sum = 0;

for (int i1 = 1; i1 < kol + 1; i1++)

{

DBGrid5->DataSource->DataSet->RecNo = i1;

if (

((DBGrid5->Columns->Items[5]->Field->AsDateTime >= datetimeBeg) && (DBGrid5->Columns->Items[5]->Field->AsDateTime <= datetimeEnd))

&&

((DBGrid5->Columns->Items[6]->Field->AsDateTime >= datetimeBeg) && (DBGrid5->Columns->Items[6]->Field->AsDateTime <= datetimeEnd))

)

{

Sum += DBGrid5->Columns->Items[8]->Field->AsInteger;

}

}

ShowMessage("Прибыль: " + IntToStr(Sum));

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::BitBtn5Click(TObject \*Sender)

{

QuickRep1->Preview();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::BitBtn6Click(TObject \*Sender)

{

DBGrid5->DataSource->DataSet->Last();

int kol = DBGrid5->DataSource->DataSet->RecordCount;

int number = 0;

for (int i1 = 1; i1 < kol + 1; i1++)

{

DBGrid5->DataSource->DataSet->RecNo = i1;

if (DBGrid5->Columns->Items[0]->Field->AsInteger == SpinEdit1->Value)

{

number = i1;

}

}

DBGrid5->DataSource->DataSet->RecNo = number;

AnsiString str = DBGrid5->Columns->Items[1]->Field->AsString;

RichEdit1->Text = "#####################\n";

RichEdit1->Text = RichEdit1->Text + "Код клиента: " + str + '\n';

str = DBGrid5->Columns->Items[2]->Field->AsString;

RichEdit1->Text = RichEdit1->Text + "Код номера: " + str + '\n';

str = DBGrid5->Columns->Items[3]->Field->AsString;

RichEdit1->Text = RichEdit1->Text + "ФИО клиента: " + str + '\n';

str = DBGrid5->Columns->Items[4]->Field->AsString;

RichEdit1->Text = RichEdit1->Text + "Стоимость 1 дня: " + str + '\n';

str = DBGrid5->Columns->Items[5]->Field->AsString;

RichEdit1->Text = RichEdit1->Text + "Дата начала: " + str + '\n';

str = DBGrid5->Columns->Items[6]->Field->AsString;

RichEdit1->Text = RichEdit1->Text + "Дата окончания: " + str + '\n';

str = DBGrid5->Columns->Items[7]->Field->AsString;

RichEdit1->Text = RichEdit1->Text + "Количество дней: " + str + '\n';

str = DBGrid5->Columns->Items[8]->Field->AsString;

RichEdit1->Text = RichEdit1->Text + "Итого: " + str + '\n';

RichEdit1->Text = RichEdit1->Text + "#####################";

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::BitBtn7Click(TObject \*Sender)

{

if(PrintDialog1->Execute())

{

RichEdit1->Print("Печать!");

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::FormCreate(TObject \*Sender)

{

Form3 = new TForm3(this);

Form3->ShowModal();

}

//---------------------------------------------------------------------------