СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc152768988)

[1 ЦЕЛЬ И ЗАДАЧИ УЧЕБНОЙ ПРАКТИКИ 4](#_Toc152768989)

[1.1 Основные требования по технике безопасности во время прохождения учебной практики 4](#_Toc152768990)

[1.2 Нормативно-правовые документы 5](#_Toc152768991)

[2 ПРЕДМЕТНАЯ ОБЛАСТЬ. АНАЛИЗ 7](#_Toc152768992)

[ПРЕДМЕТНОЙ ОБЛАСТИ. ПОСТАНОВКА ЗАДАЧИ 7](#_Toc152768993)

[2.1 Описание предметной области 7](#_Toc152768994)

[2.2 Группы пользователей информационной системы 8](#_Toc152768995)

[2.3 Основные требования, предъявляемые к информационной системе 8](#_Toc152768996)

[3 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «ТУРИСТИЧЕСКАЯ КОМПАНИЯ» 10](#_Toc152768997)

[3.1 Моделирование некоторых динамических аспектов системы 10](#_Toc152768998)

[3.1.1 Создание диаграммы вариантов использования 10](#_Toc152768999)

[3.1.2 Создание диаграммы деятельности 10](#_Toc152769000)

[3.2 Разработка функциональной структуры информационной системы. Карта навигации по информационной системе 11](#_Toc152769001)

[3.3 Разработка макета информационной системы «Туристическая компания» 11](#_Toc152769002)

[4 РАЗРАБОТКА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ «ТУРИСТИЧЕСКОЙ КОМПАНИИ» 12](#_Toc152769003)

[4.1 Моделирование и разработка базы данных 12](#_Toc152769004)

[4.2 Разработка программных модулей информационной системы «Туристической компании» 15](#_Toc152769005)

[5 РАЗРАБОТКА ТЕСТОВЫХ НАБОРОВ И ТЕСТОВЫХ СЦЕНАРИЕВ 42](#_Toc152769006)

[5.1 Назначение эксперимента. Выбор и обоснование методики проведения тестирования 42](#_Toc152769007)

[5.2 Технология тестирования 43](#_Toc152769008)

[5.3 Результаты проведения тестирования 44](#_Toc152769009)

[ЗАКЛЮЧЕНИЕ 47](#_Toc152769010)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 48](#_Toc152769011)

[ПРИЛОЖЕНИЕ 1 50](#_Toc152769012)

[ПРИЛОЖЕНИЕ 2 59](#_Toc152769013)

# ВВЕДЕНИЕ

Туризм является одной из ведущих и наиболее развивающихся отраслей экономики и за быстрые темпы он признан экономическим феноменом столетия. Быстрому его развитию способствует расширение политических, экономических, научных и культурных связей между государствами и народами мира. Массовое развитие туризма позволяет миллионам людей расширить свой кругозор, познакомиться с достопримечательностями, культурой, традициями той или иной страны.

Деятельность туристической компании включает в себя следующие основные направления:

1. Пляжный отдых: компания ООО «Вояж» представляет большое количество различных стран по всем направлениям:

* Европа: Греция, Кипр, Мальта, Черногория;
* Азия: Индонезия, Индия, Таиланд, Шри – Ланка;
* Латинская Америка: Гватемала, Доминикана, Куба;
* Африка: Египет, Камбоджа.

1. Горнолыжные курорты: компания ООО «Вояж» представляет туры в такие станы, как: Болгария, Италия, Турция, Франция, Финляндия, Швейцария. Из которых экономичными вариантами считаются туры – в Болгарию и Финляндию, а вариантами подороже считаются туры в Италию, Францию и Швейцарию.
2. Экскурсионные туры: ООО «Вояж» предлагает экскурсионные туры в Европу, Азию, Латинскую и Северную Америку, Африку, Австралию. Экскурсионные туры пользуются большим спросом и заинтересовывают клиентов своей разнообразностью.

# ЦЕЛЬ И ЗАДАЧИ УЧЕБНОЙ ПРАКТИКИ

Целью практической подготовки в виде учебной практики является закрепление общих и профессиональных компетенций, определяющих готовность обучающихся к виду профессиональной деятельности в области разработки, администрирования и защиты баз данных.

Основные задачи учебной практики:

* формирование у студентов знаний, умений и навыков, профессиональных компетенций, профессионально значимых личностных качеств;
* развитие профессионального интереса, формирование мотивационно-целостного отношения к профессиональной деятельности, готовности к выполнению профессиональных задач в соответствии с нормами морали, профессиональной этики и служебного этикета;
* адаптация студентов к профессиональной деятельности.

Для достижения поставленной цели и реализации задач практической подготовки в виде учебной практики необходимо разработать прикладное программное обеспечение деятельности агентства знакомств.

* 1. Основные требования по технике безопасности во время прохождения учебной практики

К работе в компьютерном классе допускаются лица, ознакомленные с данной инструкцией по технике безопасности и правилам поведения. Необходимо неукоснительно соблюдать правила по технике безопасности, т.к. нарушение этих правил может привести к поражению электрическим током, вызвать возгорание и навредить вашему здоровью.

* Работа обучающихся в компьютерном классе разрешается только в присутствии преподавателя (лаборанта).
* Во время занятий посторонние лица могут находиться в классе только с разрешения преподавателя.
* Во время перемен между уроками проводится обязательное проветривание компьютерного кабинета с обязательным выходом обучающихся из класса.

Каждый обучающийся в ответе за состояние своего рабочего места и сохранность размещенного на нем оборудования.

Обучающийся, допустивший нарушение по технике безопасности и правилам поведения в компьютерном классе, может быть привлечен к дисциплинарной ответственности. Если нарушение техники безопасности связано с причинением имущественного ущерба, обучающийся несет и материальную ответственность в установленном законом порядке.

* 1. Нормативно-правовые документы

Оформление и содержание технического задания должно соответствовать требованиям стандарта «ГОСТ 19.201-78. ЕСПД. Общие положения.

Техническое задание оформляют в соответствии с ГОСТ 19.106-78 на листах формата 11 и 12 по ГОСТ 2.301-68, как правило, без заполнения полей листа. Номера листов (страниц) проставляются в верхней части листа над текстом.

Лист утверждения и титульный лист оформляют в соответствии с ГОСТ 19.104-78.

Информационную часть (аннотацию и содержание), лист регистрации изменений допускается в документ не включать.

Для внесения изменений или дополнений в техническое задание на последующих стадиях разработки про1раммы или программного изделия выпускают дополнение к нему.

Согласование и утверждение дополнения к техническому заданию проводят в том же порядке, который установлен для технического задания.

Техническое задание должно содержать следующие разделы:

* введение;
* основания для разработки;
* назначение разработки;
* требования к программе или программному изделию;
* требования к программной документации;
* технико-экономические показатели;
* стадии и этапы разработки;
* порядок контроля и приемки;
* в техническое задание допускается включать приложения.

В зависимости от особенностей программы или программного изделия допускается уточнять содержание разделов, вводить новые разделы или объединять отдельные из них.

Вывод по главе: были описаны, техника безопасности во время прохождения учебной практики; нормативно-правовые документы, которые отвечают за стандарты разработки; цели и задачи учебной практики, включая вводную часть, где описана тематика отрасли и её тенденции развития.

# ПРЕДМЕТНАЯ ОБЛАСТЬ. АНАЛИЗ

# ПРЕДМЕТНОЙ ОБЛАСТИ. ПОСТАНОВКА ЗАДАЧИ

Предметная область информационной системы — это материальная система или система, характеризующая элементы материального мира, информация о которой хранится и обрабатывается. Предметная область рассматривается как некоторая совокупность реальных объектов и связей между ними.

* 1. Описание предметной области

Необходимо разработать и спроектировать прикладное программное обеспечение для эффективного управления продажами туров, скомплектованных туроператором.

Данная компания формирует туристические группы для заграничных поездок и обеспечивает им полную поддержку на маршруте. Количество туристов в группе заранее известно и ограничено. Маршрут группы может пролегать через несколько городов страны назначения. Экскурсии в несколько стран одновременно не проводятся. При обращении в «Вояж» группы из нескольких человек компания предоставляет скидку, которая зависит от количества туристов в группе. Вместе с группой следует представитель компании, который несет полную ответственность за качество услуг, предоставляемых компанией. При возникновении каких-либо неудобств на маршруте, возникших по вине компании, турист получает назад заранее оговоренную в контракте сумму.

* 1. Группы пользователей информационной системы

Потенциальные пользователи, которые будут взаимодействовать с информационной системой:

* 1. Случайный пользователь, взаимодействие которого с ИС не обусловлено служебными обязанностями.
  2. Оператор – планирует и координирует деятельность предприятия посредством просмотра таблиц. Разрешено редактирование, удаление, добавление и просмотр данных.
  3. Менеджер по работе с клиентами – работая со специально разработанными для них формами, поддерживают актуальность информации в информационной системе. Разрешено только чтение данных.
  4. Администратор – сопровождает информационную систему и обеспечивает её защиту, оптимизирует. По требованию предоставляет полномочия доступа пользователям. Разрешен полный доступ к базе данных.
  5. Основные требования, предъявляемые к информационной системе

Гибкость, способность к адаптации и дальнейшему развитию подразумевают возможность приспособления информационной системы к новым условиям, новым потребностям предприятия.

Выполнение этих условий возможно, если на этапе разработки информационной системы использовались общепринятые средства и методы документирования, так что по прошествии определенного времени сохранится возможность разобраться в структуре системы и внести в нее соответствующие изменения, даже если все разработчики или их часть по каким-либо причинам не смогут продолжить работу.

Надежность информационной системы подразумевает ее функционирование без искажения информации, потери данных по «техническим причинам».

Требование надежности обеспечивается созданием резервных копий хранимой информации, выполнения операций протоколирования, поддержанием качества каналов связи и физических носителей информации, использованием современных программных и аппаратных средств. Сюда же следует отнести защиту от случайных потерь информации в силу недостаточной квалификации персонала.

Система является эффективной, если с учетом выделенных ей ресурсов она позволяет решать возложенные на нее задачи в минимальные сроки.

В любом случае оценка эффективности будет производиться заказчиком, исходя из вложенных в разработку средств и соответствия представленной информационной системы его ожиданиям.

Под безопасностью, прежде всего, подразумевается свойство системы, в силу которого посторонние лица не имеют доступа к информационным ресурсам организации, кроме тех, которые для них предназначены.

Защита информации от постороннего доступа обеспечивается управлением доступом к ресурсам системы, использованием современных программных средств защиты информации. В крупных организациях целесообразно создавать подразделения, основным направлением деятельности которых было бы обеспечение информационной безопасности, в менее крупных организациях назначать сотрудника, ответственного за данный участок работы.

Вывод по главе: было описано, что такое предметная область и её описание; группы пользователей, которые будут пользоваться информационной системой «Туристическая компания» и основные требования к ней.

# ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «ТУРИСТИЧЕСКАЯ КОМПАНИЯ»

Проектированием информационных систем называется многоступенчатый процесс их создания и/или модернизации путём применения упорядоченной совокупности методологий и инструментария. Проектирование (в отличие от моделирования) предполагает работу с пока несуществующим объектом и направлено на создание информационной системы.

* 1. Моделирование некоторых динамических аспектов системы

При разработке программного обеспечения заблаговременное планирование и моделирование значительно упрощают программирование. На этапе создания концептуальной модели для описания функционального назначения используется диаграмма вариантов использования, а для описания логики процедур и бизнес-процессов используется диаграмма деятельности.

1. Создание диаграммы вариантов использования

В приложении 1 на рисунке 1 представлена диаграмма вариантов использования информационной системы «Туристической компании».

1. Создание диаграммы деятельности

В приложении 1 на рисунках 2 – 5 представлена диаграмма деятельности (по ролям) информационной системы «Туристическая компания».

* 1. Разработка функциональной структуры информационной системы. Карта навигации по информационной системе

Карта переходов по информационной системе «Туристическая компания» представлена в приложении 1 на рисунке 6.

* 1. Разработка макета информационной системы «Туристическая компания»

В приложении 1 на рисунке 7 представлена форма «Авторизация».

На этапе авторизации пользователь может ввести логин и пароль, чтобы продолжить работу в информационной системе.

В случае если пользователь еще не зарегистрирован в системе, он может сделать это, нажав на кнопку «Зарегистрироваться». В этом случае откроется форма «Регистрация», представленная в приложении 1 на рисунке 8.

На этапе регистрации пользователю необходимо придумать свой логин и пароль.

После того, как пользователь успешно прошел авторизацию, он может продолжить работу с таблицами на главной форме. В приложении 1 на рисунках 9 – 13 представлены главные формы, которые будут отображены, в зависимости от прав пользователя в системе. Например, для гостя будут скрыты возможности редактирования, добавления и удаления записи, так как данной группе пользователей доступен только просмотр данных.

Дизайн выполнен в простом стиле для того, чтобы пользователь с любым уровнем владения компьютером смог без препятствий работать с программным обеспечением.

Вывод по главе: была спроектирована диаграмма вариантов использования, диаграмма деятельности и макеты будущей информационной системы.

1. РАЗРАБОТКА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ «ТУРИСТИЧЕСКОЙ КОМПАНИИ»

Прототип — это один из этапов разработки, который заключается в продумывании содержания и расположения важных элементов интерфейса. Прототип — это моделирование конечного продукта. Это интерактивный макет, который может иметь любую степень точности. Основная цель создания прототипов – проверить, насколько последователен путь пользователя, и выявить препятствия, которые могут возникнуть в процессе его взаимодействия с продуктом.

Прототипы не только позволяют проверить удобство разрабатываемого продукта до начала написания кода, они также приводят к неожиданным открытиям и новым идеям, которые могут вывести проектируемый продукт на новый уровень.

* 1. Моделирование и разработка базы данных

Исходя из анализа предметной области, можно выделить семь сущностей: «Клиенты», «Документы», «Контракты», «Маршруты», «Пункты маршрута», «Гостиница» и «Экскурсии».

В приложении 1 на рисунке 14 представлена логическая модель данных базы данных «Туристическая компания».

Создание базы данных происходит в SQL Server Management Studio, в примере представлено создание базы данных в MS SQL Server.

С помощью раздела «Создание» и конструктора таблиц MS SQL созданы все таблицы и поля, представленные на логической модели данных.

Названия полей и их типы данных представлены в таблицах 1 – 8.

Таблица 1 – Поля таблицы «Клиенты»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Код клиента | Числовой |
| Фамилия | Текстовый |
| Имя | Текстовый |
| Отчество | Текстовый |
| Телефон | Числовой |

Таблица 2 – Поля таблицы «Документы»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Код документа | Числовой |
| Код клиента | Числовой |
| Документ, удостоверяющий личность | Текстовый |
| Серия документа | Числовой |
| Номер документа | Числовой |
| Дата выдачи документа | Дата |
| Кем выдан | Текстовый |
| Дата рождения | Дата |
| Наличие загран. паспорта | Логический |

Таблица 3 – Поля таблицы «Контракты»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Код контракта | Числовой |
| Код клиента | Числовой |
| Код маршрута | Числовой |

Таблица 4 – Поля таблицы «Маршруты»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Код маршрута | Числовой |
| Название маршрута | Текстовый |
| Страна | Текстовый |
| Срок прибытия | Текстовый |
| Дата вылета | Дата |
| Стоимость путевки | Денежный |
| Скидка | Числовой |
| Неустойка | Денежный |
| Код пункта | Числовой |
| Код экскурсии | Числовой |

Таблица 5 – Поля таблицы «Пункты маршрута»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Код пункта | Числовой |
| Срок пребывания в пункте маршрута | Текстовый |
| Дата прибытия | Дата |
| Дата убытия | Дата |
| Название города | Текстовый |
| Код гостиницы | Числовой |

Таблица 6 – Поля таблицы «Гостиница»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Код гостиницы | Числовой |
| Название | Текстовый |
| Класс | Числовой |

Таблица 7 – Поля таблицы «Экскурсии»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Код экскурсии | Числовой |
| Название | Текстовый |
| Дата проведения | Дата |
| Стоимость | Денежный |

Таблица 8 – Поля таблицы «Пользователи»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Код пользователя | Числовой |
| Логин | Текстовый |
| Пароль | Числовой |
| Права доступа | Текстовый |

В приложении 1 на рисунке 15 представлены все созданные таблицы в обозревателе объектов базы данных.

После того, как таблицы созданы, необходимо создать между ними связи.

В приложении 1 на рисунке 16 представлена физическая модель данных, отражающая все таблицы и их поля, а также связи между таблицами.

* 1. Разработка программных модулей информационной системы «Туристической компании»

Разработанные в MS Visual Studio формы продемонстрированы в приложении 1 на рисунке 7–13.

Свойства элементов формы «Авторизация» представлены в таблице 9.

Таблица 9 – Свойства элементов формы «Авторизация»

|  |  |  |
| --- | --- | --- |
| Объект | Свойство | Значение |
| Авторизация | FormBorderStyle | None |
| Text | Авторизация |
| StartPosition | CenterScreen |
| BackColor | DarkSeaGreen |
| Size | 409; 272 |
| Авторизация\_label1 | Text | Авторизация |
| Font | Times New Roman; 30pt |
| Видимость\_пароля\_checkBox1 | Size | 122; 20 |
| Font | Times New Roman; 10pt |
| выйти\_label9 | Text | Выход |
| Font | Times New Roman; 11pt |
| Кнопка\_Войти\_как\_гость\_label8 | Size | 109; 17 |
| Text | Войти как гость |
| Font | Times New Roman; 11pt |
| Кнопка\_подтверждения\_label6 | Size | 86; 17 |
| Text | Продолжить |
| Font | Times New Roman; 11pt |
| BackColor | LightSteelBlue |
| Кнопка\_Рагистрация\_label7 | Size | 134; 17 |
| Text | Зарегистрироваться |
| Font | Times New Roman; 11pt |
| BackColor | LightSteelBlue |
| Логин\_label3 | Size | 71; 23 |
| Text | Логин: |
| Font | Times New Roman; 15pt |
| Пароль\_label4 | Size | 81; 23 |
| Text | Пароль: |
| Font | Times New Roman; 15pt |
| Логин\_textBox1 | Font | Times New Roman; 13pt |
| Size | 179; 27 |
| Пароль\_textBox2 | Font | Times New Roman; 13pt |
| Size | 179; 27 |
| Фон\_ label2 | BackColor | LightSteelBlue |
| Size | 354; 207 |
| BorderStyle | Fixed3D |

Событийно-управляемые процедуры в форме «Авторизация» представлены в таблице 10.

Таблица 10 – Событийно-управляемые процедуры в форме «Авторизация»

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| Видимость\_пароля\_checkBox1\_CheckedChanged | private void checkBox1\_CheckedChanged(object sender, EventArgs e)  {  //Видимость пароля  if (checkBox1.Checked)  {  textBox2.UseSystemPasswordChar = false;  }  else  {  textBox2.UseSystemPasswordChar = true;  }  } |
| Кнопка\_подтверждения\_ label6\_Click | private void label6\_Click(object sender, EventArgs e)  {  //Переходит на форму 'Администратор', 'Оператор', 'Менеджер', 'Клиент' и 'Гость'  if ((textBox1.Text == "") || (textBox2.Text == ""))  {  if ((textBox1.Text == "") && (textBox2.Text == "") && (textBox3.Text == ""))  {  MessageBox.Show("Заполните все поля!!!");  }  else  {  if ((textBox1.Text == "") && (textBox2.Text == ""))  {  MessageBox.Show("Введите Логин и Пароль!!!");  }  else  {  if (textBox1.Text == "")  {  MessageBox.Show("Введите Логин!!!");  }  else  {  if (textBox2.Text == "")  {  MessageBox.Show("Введите Пароль!!!");  }  }  }  }  }  else  {  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = УП01\_ИСПП5\_Шипицына\_ОИ; Integrated Security = True");  sqlConnect.Open();  SqlDataAdapter da = new SqlDataAdapter("select \* from Пользователи", sqlConnect);  DataTable dt = new DataTable();  da.Fill(dt);  Boolean log = false, password = false;  for (int i = 0; i < dt.Rows.Count; i++)  if (dt.Rows[i]["Логин"].ToString() == textBox1.Text)  {  if (dt.Rows[i]["Пароль"].ToString() == textBox2.Text)  {  password = true;  textBox3.Text = dt.Rows[i]["Права доступа"].ToString();  }  log = true;  }  if ((password == true) && (log == true))  {  if (textBox3.Text == "Гость")  {  this.Hide();  Клиент UsrFrm = new Клиент();  UsrFrm.Show();  UsrFrm.ii("Гость");  }  if (textBox3.Text == "Клиент")  {  this.Hide();  Клиент UsrFrm = new Клиент();  UsrFrm.Show();  UsrFrm.ii("Клиент");  }  if (textBox3.Text == "Менеджер ")  {  this.Hide();  Менеджер UsrFrm = new Менеджер();  UsrFrm.Show();  UsrFrm.ii("Менеджер ");  }  if (textBox3.Text == "Администратор")  {  this.Hide();  Администратор UsrFrm = new Администратор();  UsrFrm.Show();  UsrFrm.ii("Администратор");  }  if (textBox3.Text == "Оператор")  {  this.Hide();  Оператор UsrFrm = new Оператор();  UsrFrm.Show();  UsrFrm.ii("Оператор");  }  }  else  {  if ((password == false) && (log == false))  {  MessageBox.Show("Данный пользователь отсутвует!!!");  }  else  {  if (password == false)  {  MessageBox.Show("Неверно введен Пароль!!!");  popitki -= 1;  label4.Text = "Количество попыток для ввода Пароля осталось: " + popitki;  if (popitki == 0)  {  label5.Visible = true;  label7.Visible = false;  label8.Location = new Point(113, 125);  label9.Visible = false;  }  }  }  }  }  } |
| Кнопка\_Рагистрация\_label7\_Click | private void label7\_Click(object sender, EventArgs e)  {  //Переходит на форму 'Регистрация'  this.Hide();  Регистрация UsrFrm4 = new Регистрация();  UsrFrm4.Show();  } |
| Кнопка\_Войти\_как\_гость\_label8\_Click | private void label8\_Click(object sender, EventArgs e)  {  //Переходит на форму 'Гость'  this.Hide();  Гость UsrFrm3 = new Гость();  UsrFrm3.Show();  } |
| Кнопка\_выйти\_label9\_Click | private void label9\_Click(object sender, EventArgs e)  {  //Прекращает работу программы  Application.Exit();  } |
| Пароль\_textBox2\_KeyPress | private void textBox2\_KeyPress(object sender, KeyPressEventArgs e)  {  //Проверяет на отсутствие букв  if (!char.IsDigit(e.KeyChar) && (e.KeyChar != (char)Keys.Back))  {  e.Handled = true;  MessageBox.Show("Поле не может содержать буквы!");  }  } |

Свойства элементов формы «Регистрация» представлены в таблице 11.

Таблица 11 – Свойства элементов формы «Регистрация»

| Объект | Свойство | Значение |
| --- | --- | --- |
| Регистрация | FormBorderStyle | Fixed3D |
| Text | “ ” |
| StartPosition | CenterScreen |
| BackColor | 192; 192; 255 |
| Size | 377; 300 |
| ShowIcon | False |
| Регистрация\_label1 | Text | Регистрация |
| Font | Times New Roman; 30pt |
| Вернуться\_label6 | Size | 78; 19 |
| Text | Вернуться |
| Font | Times New Roman; 12pt |
| BackColor | LightSteelBlue |
| Логин\_textBox1 | Font | Times New Roman; 13pt |
| Size | 179; 27 |
| Пароль\_label4 | Text | Пароль: |
| Font | Times New Roman; 15,75pt |
| Пароль\_textBox2 | Font | Times New Roman; 13pt |
| Size | 179; 27 |
| Подтвердить\_label5 | Size | 33; 19 |
| Text | ОК |
| Font | Times New Roman; 12pt |
| BackColor | LightSteelBlue |
| Фон\_label2 | BackColor | LightSteelBlue |
| Size | 296; 156 |
| BorderStyle | Fixed3D |

Событийно-управляемые процедуры в форме «Регистрация» представлены в таблице 12.

Таблица 12 – Событийно-управляемые процедуры в форме «Регистрация»

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| class Регистрация | public partial class Регистрация : Form  {  public Регистрация()  {  InitializeComponent();  }  } |
| Кнопка\_подтверждения\_label5\_Click | private void label5\_Click(object sender, EventArgs e)  {  //Переходит на форму 'Авторизация'  if ((textBox2.Text == "") || (textBox1.Text == ""))  {  MessageBox.Show("Не все данные заполнены!", "Ошибка");  }  else  {  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП01\_ИСПП5\_Шипицына\_ОИ; Integrated Security = True");  sqlConnect.Open();  DataTable dt = new DataTable();  DataTable dta = new DataTable();  SqlDataAdapter daa = new SqlDataAdapter($"select \* from Пользователи", sqlConnect);  daa.Fill(dta);  string n = dta.Rows[dta.Rows.Count - 1]["Id"].ToString();  SqlDataAdapter da = new SqlDataAdapter($"INSERT INTO Пользователи (Id,Логин,Пароль,[Права доступа]) VALUES ('{int.Parse(n) + 1}','{textBox1.Text}','{textBox2.Text}','Гость');", sqlConnect);  da.Fill(dt);  пользователиTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пользователи);  MessageBox.Show($"Регистрация пользователя с логином {textBox1.Text} выполнена.\nВы зарегистрированы как гость");  this.Hide();  Авторизация UsrFrm = new Авторизация();  UsrFrm.Show();  }  } |
| Пароль\_textBox2\_KeyPress | private void textBox2\_KeyPress(object sender, KeyPressEventArgs e)  {  //Проверяет на отсутствие букв  if (!char.IsDigit(e.KeyChar) && (e.KeyChar != (char)Keys.Back))  {  e.Handled = true;  MessageBox.Show("Поле не может содержать буквы!");  }  } |
| Вернуться\_label6\_Click | private void label6\_Click(object sender, EventArgs e)  {  //Переход на форму авторизации  Авторизация AvtFrm = new Авторизация();  AvtFrm.Show();  this.Hide();  } |

Свойства элементов формы «Администратор» представлены в таблице 13.

Таблица 13 – Свойства элементов формы «Администратор»

| Объект | Свойство | Значение |
| --- | --- | --- |
| Администратор | FormBorderStyle | SizableToolWindow |
| Text | “ ” |
| StartPosition | CenterScreen |
| BackColor | 192; 192; 255 |
| Size | 797; 392 |
| Вернуться\_ toolStripMenuItem2 | BackColor | Silver |
| Text | Вернуться |
| ForeColor | ControlText |
| Выйти\_toolStripMenuItem12 | Size | 180; 28 |
| Text | Выход |
| Меню\_ toolStripMenuItem1 | BackColor | 128; 128; 255 |
| Text | Меню |
| ForeColor | ControlText |
| Обновить\_ pictureBox3 | BackColor | 128; 128; 255 |
| SizeMode | Zoom |
| Поиск\_Label1 | BackColor | 128; 128; 255 |
| Size | 100; 52 |
| BorderStyle | None |
| Поле\_для\_таблиц\_DataGridView1 | Size | 757; 288 |
| BackColor | Control |
| Строка\_поиска\_TextBox1 | Font | Segoe Print; 7pt |
| Size | 181; 24 |
| Таблица\_Контракты \_ToolStripMenuItem4 | BackColor | Silver |
| Text | Контракты |
| ForeColor | ControlText |
| Таблица\_Клиенты\_ToolStripMenuItem5 | BackColor | Silver |
| Text | Клиенты |
| ForeColor | ControlText |
| Таблица\_Документы\_ToolStripMenuItem6 | BackColor | Silver |
| Text | Документы |
| ForeColor | ControlText |
| Таблица\_Маршруты\_ToolStripMenuItem7 | BackColor | Silver |
| Text | Маршруты |
| ForeColor | ControlText |
| Таблица\_Пункты\_маршрута\_ToolStripMenuItem8 | BackColor | Silver |
| Text | Пункты маршрута |
| ForeColor | ControlText |
| Таблица\_Гостиница\_ToolStripMenuItem9 | BackColor | Silver |
| Text | Гостиница |
| ForeColor | ControlText |
| Таблица\_Экскурсии\_ToolStripMenuItem10 | BackColor | Silver |
| Text | Экскурсии |
| ForeColor | ControlText |
| Таблица\_Пользователи\_ToolStripMenuItem11 | BackColor | Silver |
| Text | Пользователи |
| ForeColor | ControlText |

Событийно-управляемые процедуры в форме «Администратор» представлены в таблице 14.

Таблица 14 – Событийно-управляемые процедуры в форме «Администратор»

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| class Администратор | public void tab(int i)  { //Добавляет таблицы  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП01\_ИСПП5\_Шипицына\_ОИ; Integrated Security = True");  sqlConnect.Open();  switch (i)  {  case 0: dataGridView1.DataSource = контрактыBindingSource; break;  case 1: dataGridView1.DataSource = клиентыBindingSource; break;  case 2: dataGridView1.DataSource = документыBindingSource; break;  case 3: dataGridView1.DataSource = маршрутыBindingSource; break;  case 4: dataGridView1.DataSource = пунктыМаршрутаBindingSource; break;  case 5: dataGridView1.DataSource = гостиницаBindingSource; break;  case 6: dataGridView1.DataSource = экскурсииBindingSource; break;  case 7: dataGridView1.DataSource = пользователиBindingSource; break;  }  } |
| Вернуться\_ toolStripMenuItem2\_Click | private void toolStripMenuItem2\_Click(object sender, EventArgs e)  { //Переходит на форму 'Авторизация'  Авторизация AvtFrm = new Авторизация();  AvtFrm.Show();  this.Hide();  } |
| Обновить\_ pictureBox3\_Click | private void pictureBox3\_Click(object sender, EventArgs e)  { //Обновляет и сохраняет изменениям в таблицах  switch (c)  {  case 0: контрактыTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Контракты); break;  case 1: клиентыTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Клиенты); break;  case 2: документыTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Документы); break;  case 3: маршрутыTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Маршруты); break;  case 4: пункты\_маршрутаTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пункты\_маршрута); break;  case 5: гостиницаTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Гостиница); break;  case 6: экскурсииTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Экскурсии); break;  case 7: пользователиTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пользователи); break;  }  } |
| Выйти\_toolStripMenuItem12\_Click | private void toolStripMenuItem12\_Click(object sender, EventArgs e)  { //Прекращает работу программы  Application.Exit();  } |
| Таблица\_Контракты\_ toolStripMenuItem4\_Click | private void toolStripMenuItem4\_Click(object sender, EventArgs e)  { //Открывает таблицу 'Контракты'  tab(0);  c = 0;  } |
| Таблица\_Клиенты\_ toolStripMenuItem5\_Click | private void toolStripMenuItem5\_Click(object sender, EventArgs e)  { //Открывает таблицу 'Клиенты'  tab(1);  c = 1;  } |
| Таблица\_Документы\_ toolStripMenuItem6\_Click | private void toolStripMenuItem6\_Click(object sender, EventArgs e)  { //Открывает таблицу 'Документы'  tab(2);  c = 2;  } |
| Таблица\_Маршруты\_ toolStripMenuItem7\_Click | private void toolStripMenuItem7\_Click(object sender, EventArgs e)  { //Открывает таблицу 'Маршруты'  tab(3);  c = 3;  } |
| Таблица\_Пункты\_маршрута\_ toolStripMenuItem8\_Click | private void toolStripMenuItem8\_Click(object sender, EventArgs e)  { //Открывает таблицу 'Пункты маршрута'  tab(4);  c = 4;  } |
| Таблица\_Гостиница\_ toolStripMenuItem9\_Click | private void toolStripMenuItem9\_Click(object sender, EventArgs e)  { //Открывает таблицу 'Гостиница'  tab(5);  c = 5;  } |
| Таблица\_Экскурсии\_ toolStripMenuItem10\_Click | private void toolStripMenuItem10\_Click(object sender, EventArgs e)  { //Открывает таблицу 'Экскурсии'  tab(6);  c = 6;  } |
| Таблица\_Пользователи\_ toolStripMenuItem11\_Click | private void toolStripMenuItem11\_Click(object sender, EventArgs e)  { //Открывает таблицу 'Пользователи'  tab(7);  c = 7;  } |
| Поисковая\_строка\_ textBox1\_TextChanged | private void textBox1\_TextChanged(object sender, EventArgs e)  { //Выделение строки, которая подходит под введённые данные в поисковую строку  for (int i = 0; i < dataGridView1.RowCount; i++)  {  dataGridView1.Rows[i].Selected = false;  for (int j = 0; j < dataGridView1.ColumnCount; j++)  if (dataGridView1.Rows[i].Cells[j].Value != null)  if (dataGridView1.Rows[i].Cells[j].Value.ToString().Contains(textBox1.Text))  {  dataGridView1.Rows[i].Selected = true;  break;  }  }  } |
| Администратор1\_Load | private void Администратор1\_Load(object sender, EventArgs e)  { //Загрузка данных в таблицы  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Экскурсии". При необходимости она может быть перемещена или удалена.  this.экскурсииTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Экскурсии);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пункты\_маршрута". При необходимости она может быть перемещена или удалена.  this.пункты\_маршрутаTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пункты\_маршрута);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пользователи". При необходимости она может быть перемещена или удалена.  this.пользователиTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пользователи);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Маршруты". При необходимости она может быть перемещена или удалена.  this.маршрутыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Маршруты);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Контракты". При необходимости она может быть перемещена или удалена.  this.контрактыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Контракты);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Клиенты". При необходимости она может быть перемещена или удалена.  this.клиентыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Клиенты);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Документы". При необходимости она может быть перемещена или удалена.  this.документыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Документы);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Гостиница". При необходимости она может быть перемещена или удалена.  this.гостиницаTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Гостиница);  } |

Свойства элементов формы «Оператор» представлены в таблице 15.

Таблица 15 – Свойства элементов формы «Оператор»

| Объект | Свойство | Значение |
| --- | --- | --- |
| Оператор | FormBorderStyle | SizableToolWindow |
| Text | Оператор |
| StartPosition | CenterScreen |
| Size | 798; 402 |
| ShowIcon | False |
| Вернуться\_ToolStripMenuItem2 | BackColor | Silver |
| Text | Вернуться |
| ForeColor | Black |
| Выйти\_ToolStripMenuItem11 | Size | 180; 28 |
| Text | Выход |
| Меню\_ToolStripMenuItem1 | BackColor | 128; 128; 255 |
| Text | Меню |
| ForeColor | ControlText |
| Обновить\_ pictureBox3 | BackColor | 128; 128; 255 |
| SizeMode | Zoom |
| Поиск\_Label1 | BackColor | 128; 128; 255 |
| Size | 100; 52 |
| BorderStyle | None |
| Поле\_для\_таблиц\_DataGridView1 | Size | 757; 288 |
| BackColor | Control |
| Строка\_поиска\_TextBox1 | Font | Segoe Print; 7pt |
| Size | 181; 24 |
| Таблица\_Контракты \_ToolStripMenuItem4 | BackColor | Silver |
| Text | Контракты |
| ForeColor | ControlText |
| Таблица\_Клиенты\_ToolStripMenuItem5 | BackColor | Silver |
| Text | Клиенты |
| ForeColor | ControlText |
| Таблица\_Документы\_ToolStripMenuItem6 | BackColor | Silver |
| Text | Документы |
| ForeColor | ControlText |
| Таблица\_Маршруты\_ToolStripMenuItem7 | BackColor | Silver |
| Text | Маршруты |
| ForeColor | ControlText |
| Таблица\_Пункты\_маршрута\_ToolStripMenuItem8 | BackColor | Silver |
| Text | Пункты маршрута |
| ForeColor | ControlText |
| Таблица\_Гостиница\_ToolStripMenuItem9 | BackColor | Silver |
| Text | Гостиница |
| ForeColor | ControlText |
| Таблица\_Экскурсии\_ToolStripMenuItem10 | BackColor | Silver |
| Text | Экскурсии |
| BackColor | Silver |

Событийно-управляемые процедуры в форме «Оператор» представлены в таблице 16.

Таблица 16 – Событийно-управляемые процедуры в форме «Оператор»

|  |  |
| --- | --- |
| Объект | Событийно-управляемые процедуры |
| class Оператор | public void tab(int i)  { //Добавляет таблицы  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП01\_ИСПП5\_Шипицына\_ОИ; Integrated Security = True");  sqlConnect.Open();  switch (i)  {  case 0: dataGridView1.DataSource = контрактыBindingSource; break;  case 1: dataGridView1.DataSource = клиентыBindingSource; break;  case 2: dataGridView1.DataSource = документыBindingSource; break;  case 3: dataGridView1.DataSource = маршрутыBindingSource; break;  case 4: dataGridView1.DataSource = пунктыМаршрутаBindingSource; break;  case 5: dataGridView1.DataSource = гостиницаBindingSource; break;  case 6: dataGridView1.DataSource = экскурсииBindingSource; break;  }  } |
| Вернуться\_ toolStripMenuItem2\_Click | private void toolStripMenuItem2\_Click(object sender, EventArgs e)  { //Переходит на форму 'Авторизация'  Авторизация AvtFrm = new Авторизация();  AvtFrm.Show();  this.Hide();  } |
| Обновить\_ pictureBox3\_Click | private void pictureBox3\_Click(object sender, EventArgs e)  { //Обновляет и сохраняет изменениям в таблицах  switch (c)  {  case 0: контрактыTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Контракты); break;  case 1: клиентыTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Клиенты); break;  case 2: документыTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Документы); break;  case 3: маршрутыTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Маршруты); break;  case 4: пункты\_маршрутаTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пункты\_маршрута); break;  case 5: гостиницаTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Гостиница); break;  case 6: экскурсииTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Экскурсии); break;  case 7: пользователиTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пользователи); break;  }  } |
| Выйти\_toolStripMenuItem12\_Click | private void toolStripMenuItem12\_Click(object sender, EventArgs e)  { //Прекращает работу программы  Application.Exit();  } |
| Таблица\_Контракты\_ toolStripMenuItem4\_Click | private void toolStripMenuItem4\_Click(object sender, EventArgs e)  { //Открывает таблицу 'Контракты'  tab(0);  c = 0;  } |
| Таблица\_Клиенты\_ toolStripMenuItem5\_Click | private void toolStripMenuItem5\_Click(object sender, EventArgs e)  { //Открывает таблицу 'Клиенты'  tab(1);  c = 1;  } |
| Таблица\_Документы\_ toolStripMenuItem6\_Click | private void toolStripMenuItem6\_Click(object sender, EventArgs e)  { //Открывает таблицу 'Документы'  tab(2);  c = 2;  } |
| Таблица\_Маршруты\_ toolStripMenuItem7\_Click | private void toolStripMenuItem7\_Click(object sender, EventArgs e)  { //Открывает таблицу 'Маршруты'  tab(3);  c = 3;  } |
| Таблица\_Пункты\_маршрута\_ toolStripMenuItem8\_Click | private void toolStripMenuItem8\_Click(object sender, EventArgs e)  { //Открывает таблицу 'Пункты маршрута'  tab(4);  c = 4;  } |
| Таблица\_Гостиница\_ toolStripMenuItem9\_Click | private void toolStripMenuItem9\_Click(object sender, EventArgs e)  { //Открывает таблицу 'Гостиница'  tab(5);  c = 5;  } |
| Таблица\_Экскурсии\_ toolStripMenuItem10\_Click | private void toolStripMenuItem10\_Click(object sender, EventArgs e)  { //Открывает таблицу 'Экскурсии'  tab(6);  c = 6;  } |
| Поисковая\_строка\_ textBox1\_TextChanged | private void textBox1\_TextChanged(object sender, EventArgs e)  { //Выделение строки, которая подходит под введённые данные в поисковую строку  for (int i = 0; i < dataGridView1.RowCount; i++)  {  dataGridView1.Rows[i].Selected = false;  for (int j = 0; j < dataGridView1.ColumnCount; j++)  if (dataGridView1.Rows[i].Cells[j].Value != null)  if (dataGridView1.Rows[i].Cells[j].Value.ToString().Contains(textBox1.Text))  {  dataGridView1.Rows[i].Selected = true;  break;  }  }  } |
| Оператор\_Load | private void Оператор\_Load(object sender, EventArgs e)  { //Загрузка данных в таблицы  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Экскурсии". При необходимости она может быть перемещена или удалена.  this.экскурсииTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Экскурсии);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пункты\_маршрута". При необходимости она может быть перемещена или удалена.  this.пункты\_маршрутаTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пункты\_маршрута);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Маршруты". При необходимости она может быть перемещена или удалена.  this.маршрутыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Маршруты);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Клиенты". При необходимости она может быть перемещена или удалена.  this.клиентыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Клиенты);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Контракты". При необходимости она может быть перемещена или удалена.  this.контрактыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Контракты);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Документы". При необходимости она может быть перемещена или удалена.  this.документыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Документы);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Гостиница". При необходимости она может быть перемещена или удалена.  this.гостиницаTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Гостиница);  } |

Свойства элементов формы «Менеджер» представлены в таблице 17.

Таблица 17 – Свойства элементов формы «Менеджер»

| Объект | Свойство | Значение |
| --- | --- | --- |
| Менеджер | FormBorderStyle | SizableToolWindow |
| Text | Менеджер |
| StartPosition | CenterScreen |
| Size | 803; 409 |
| ShowIcon | False |
| Вернуться\_ToolStripMenuItem2 | BackColor | Silver |
| Text | Вернуться |
| ForeColor | Black |
| Выйти\_ToolStripMenuItem11 | Size | 180; 28 |
| Text | Выход |
| Меню\_ToolStripMenuItem1 | BackColor | 128; 128; 255 |
| Text | Меню |
| ForeColor | ControlText |
| Поиск\_Label1 | BackColor | 128; 128; 255 |
| Size | 100; 52 |
| BorderStyle | None |
| Поле\_для\_таблиц\_DataGridView1 | Size | 757; 288 |
| BackColor | Control |
| Строка\_поиска\_TextBox1 | Font | Segoe Print; 7pt |
| Size | 181; 24 |
| Таблица\_Контракты \_ToolStripMenuItem4 | BackColor | Silver |
| Text | Контракты |
| ForeColor | ControlText |
| Таблица\_Клиенты\_ToolStripMenuItem5 | BackColor | Silver |
| Text | Клиенты |
| ForeColor | ControlText |
| Таблица\_Документы\_ToolStripMenuItem6 | BackColor | Silver |
| Text | Документы |
| ForeColor | ControlText |
| Таблица\_Маршруты\_ToolStripMenuItem7 | BackColor | Silver |
| Text | Маршруты |
| ForeColor | ControlText |
| Таблица\_Пункты\_маршрута\_ToolStripMenuItem8 | BackColor | Silver |
| Text | Пункты маршрута |
| ForeColor | ControlText |
| Таблица\_Гостиница\_ToolStripMenuItem9 | BackColor | Silver |
| Text | Гостиница |
| ForeColor | ControlText |
| Таблица\_Экскурсии\_ToolStripMenuItem10 | BackColor | Silver |
| Text | Экскурсии |
| BackColor | Silver |

Событийно-управляемые процедуры в форме «Менеджер» представлены в таблице 18.

Таблица 18 – Событийно-управляемые процедуры в форме «Менеджер»

|  |  |
| --- | --- |
| Объект | Событийно-управляемые процедуры |
| Class Менеджер | public void Tub(int i)  { //Добавляет таблицы  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП01\_ИСПП5\_Шипицына\_ОИ; Integrated Security = True");  sqlConnect.Open();  SqlDataAdapter[] so = new SqlDataAdapter[7];  so[0] = new SqlDataAdapter("select \* from Контракты", sqlConnect);  so[1] = new SqlDataAdapter("select \* from Клиенты", sqlConnect);  so[2] = new SqlDataAdapter("select \* from Документы", sqlConnect);  so[3] = new SqlDataAdapter("select \* from Маршруты", sqlConnect);  so[4] = new SqlDataAdapter("select \* from [Пункты маршрута]", sqlConnect);  so[5] = new SqlDataAdapter("select \* from Гостиница", sqlConnect);  so[6] = new SqlDataAdapter("select \* from Экскурсии", sqlConnect);  DataSet ps = new DataSet();  so[i].Fill(ps);  dataGridView1.DataSource = ps.Tables[0];  } |
| Менеджер\_Load | private void Менеджер\_Load(object sender, EventArgs e)  { //Загрузка данных в таблицы  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Экскурсии". При необходимости она может быть перемещена или удалена.  this.экскурсииTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Экскурсии);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пункты\_маршрута". При необходимости она может быть перемещена или удалена.  this.пункты\_маршрутаTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пункты\_маршрута);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Маршруты". При необходимости она может быть перемещена или удалена.  this.маршрутыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Маршруты);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Контракты". При необходимости она может быть перемещена или удалена.  this.контрактыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Контракты);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Клиенты". При необходимости она может быть перемещена или удалена.  this.клиентыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Клиенты);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Документы". При необходимости она может быть перемещена или удалена.  this.документыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Документы);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Гостиница". При необходимости она может быть перемещена или удалена.  this.гостиницаTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Гостиница);  } |
| Вернуться\_toolStripMenuItem2\_Click1 | //Переходит на форму 'Авторизация'  private void toolStripMenuItem2\_Click(object sender, EventArgs e)  {  Авторизация AvtFrm = new Авторизация();  AvtFrm.Show();  this.Hide();  } |
| Выйти\_toolStripMenuItem12\_Click | private void toolStripMenuItem12\_Click(object sender, EventArgs e)  { //Прекращает работу программы  Application.Exit();  } |
| Таблица\_Маршруты\_toolStripMenuItem15\_Click | private void toolStripMenuItem15\_Click(object sender, EventArgs e)  {//Открывает таблицу 'Маршруты'  Tub(3);  t = 3;  a = true;  dataGridView1.ReadOnly = false;  } |
| Таблица\_Гостиница\_ toolStripMenuItem17\_Click | private void toolStripMenuItem17\_Click(object sender, EventArgs e)  {//Открывает таблицу 'Гостиница'  Tub(5);  t = 5;  a = true;  dataGridView1.ReadOnly = false;  } |
| Таблица\_Пункты\_маршрута\_ toolStripMenuItem16\_Click | private void toolStripMenuItem16\_Click(object sender, EventArgs e)  {//Открывает таблицу 'Пункты маршрута'  Tub(4);  t = 4;  a = true;  dataGridView1.ReadOnly = false;  } |
| Таблица\_Экскурсии \_ toolStripMenuItem18\_Click | private void toolStripMenuItem18\_Click(object sender, EventArgs e)  {//Открывает таблицу 'Экскурсии'  Tub(6);  t = 6;  a = true;  dataGridView1.ReadOnly = false;  } |
| Таблица\_Документы \_ toolStripMenuItem14\_Click | private void toolStripMenuItem14\_Click(object sender, EventArgs e)  { //Открывает таблицу 'Документы'  Tub(2);  t = 2;  a = true;  dataGridView1.ReadOnly = false;  } |
| Таблица\_Клиенты \_ toolStripMenuItem13\_Click | private void toolStripMenuItem13\_Click(object sender, EventArgs e)  { //Открывает таблицу 'Клиенты'  Tub(1);  t = 1;  a = true;  dataGridView1.ReadOnly = false;  } |
| Таблица\_Контракты \_ toolStripMenuItem11\_Click | private void toolStripMenuItem11\_Click\_1(object sender, EventArgs e)  { //Открывает таблицу 'Контракты'  Tub(0);  t = 0;  a = true;  dataGridView1.ReadOnly = false;  } |
| Поисковая\_строка\_ textBox1\_TextChanged | private void textBox1\_TextChanged(object sender, EventArgs e)  {  //Выделение строки, которая подходит под введённые данные в поисковую строку  for (int i = 0; i < dataGridView1.RowCount; i++)  {  dataGridView1.Rows[i].Selected = false;  for (int j = 0; j < dataGridView1.ColumnCount; j++)  if (dataGridView1.Rows[i].Cells[j].Value != null)  if (dataGridView1.Rows[i].Cells[j].Value.ToString().Contains(textBox1.Text))  {  dataGridView1.Rows[i].Selected = true;  break;  }  }  } |

Свойства элементов формы «Клиент» представлены в таблице 19.

Таблица 19 – Свойства элементов формы «Клиент»

| Объект | Свойство | Значение |
| --- | --- | --- |
| Клиент | FormBorderStyle | SizableToolWindow |
| Text | Клиент |
| StartPosition | CenterScreen |
| Size | 803; 409 |
| ShowIcon | False |
| Вернуться\_ToolStripMenuItem2 | BackColor | Silver |
| Text | Вернуться |
| ForeColor | Black |
| Выйти\_ToolStripMenuItem11 | Size | 180; 28 |
| Text | Выход |
| Меню\_ToolStripMenuItem1 | BackColor | 128; 128; 255 |
| Text | Меню |
| ForeColor | ControlText |
| Поиск\_Label1 | BackColor | 128; 128; 255 |
| Size | 100; 52 |
| BorderStyle | None |
| Поле\_для\_таблиц\_DataGridView1 | Size | 757; 288 |
| BackColor | Control |
| Строка\_поиска\_TextBox1 | Font | Segoe Print; 7pt |
| Size | 181; 24 |
| Таблица\_Маршруты\_ToolStripMenuItem7 | BackColor | Silver |
| Text | Маршруты |
| ForeColor | ControlText |
| Таблица\_Пункты\_маршрута\_ToolStripMenuItem8 | BackColor | Silver |
| Text | Пункты маршрута |
| ForeColor | ControlText |
| Таблица\_Гостиница\_ToolStripMenuItem9 | BackColor | Silver |
| Text | Гостиница |
| ForeColor | ControlText |
| Таблица\_Экскурсии\_ToolStripMenuItem10 | BackColor | Silver |
| Text | Экскурсии |
| BackColor | Silver |

Событийно-управляемые процедуры в форме «Клиент» представлены в таблице 20.

Таблица 20 – Событийно-управляемые процедуры в форме «Клиент»

|  |  |
| --- | --- |
| Объект | Событийно-управляемые процедуры |
| Class Клиент | //Добавляет таблицы  public void Tub(int i)  {  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП01\_ИСПП5\_Шипицына\_ОИ; Integrated Security = True");  sqlConnect.Open();  SqlDataAdapter[] so = new SqlDataAdapter[4];  so[0] = new SqlDataAdapter("select \* from Маршруты", sqlConnect);  so[1] = new SqlDataAdapter("select \* from [Пункты маршрута]", sqlConnect);  so[2] = new SqlDataAdapter("select \* from Гостиница", sqlConnect);  so[3] = new SqlDataAdapter("select \* from Экскурсии", sqlConnect);  DataSet ps = new DataSet();  so[i].Fill(ps);  dataGridView1.DataSource = ps.Tables[0];  } |
| Гость\_Load | //Загрузка данных в таблицы  private void Гость\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Гостиница". При необходимости она может быть перемещена или удалена.  this.гостиницаTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Гостиница);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Экскурсии". При необходимости она может быть перемещена или удалена.  this.экскурсииTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Экскурсии);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пункты\_маршрута". При необходимости она может быть перемещена или удалена.  this.пункты\_маршрутаTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пункты\_маршрута);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Маршруты". При необходимости она может быть перемещена или удалена.  this.маршрутыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Маршруты);  } |
| Вернуться\_toolStripMenuItem2\_Click | //Переходит на форму 'Авторизация'  private void toolStripMenuItem2\_Click(object sender, EventArgs e)  {  Авторизация AvtFrm = new Авторизация();  AvtFrm.Show();  this.Hide();  } |
| Выйти\_toolStripMenuItem4\_Click | private void toolStripMenuItem4\_Click(object sender, EventArgs e)  { //Прекращает работу программы  Application.Exit();  } |
| Таблица\_Маршруты\_toolStripMenuItem7\_Click | private void toolStripMenuItem7\_Click(object sender, EventArgs e)  {//Открывает таблицу 'Маршруты'  Tub(0);  t = 0;  a = true;  dataGridView1.ReadOnly = false;  } |
| Таблица\_Гостиница\_ toolStripMenuItem9\_Click | private void toolStripMenuItem9\_Click(object sender, EventArgs e)  {//Открывает таблицу 'Гостиница'  Tub(2);  t = 2;  a = true;  dataGridView1.ReadOnly = false;  } |
| Таблица\_Пункты\_маршрута\_ toolStripMenuItem8\_Click | private void toolStripMenuItem8\_Click(object sender, EventArgs e)  {//Открывает таблицу 'Пункты маршрута'  Tub(1);  t = 1;  a = true;  dataGridView1.ReadOnly = false;  } |
| Таблица\_Экскурсии \_ toolStripMenuItem10\_Click | private void toolStripMenuItem10\_Click(object sender, EventArgs e)  {//Открывает таблицу 'Экскурсии'  Tub(3);  t = 3;  a = true;  dataGridView1.ReadOnly = false;  } |
| Поисковая\_строка\_ textBox1\_TextChanged | private void textBox1\_TextChanged(object sender, EventArgs e)  {  //Выделение строки, которая подходит под введённые данные в поисковую строку  for (int i = 0; i < dataGridView1.RowCount; i++)  {  dataGridView1.Rows[i].Selected = false;  for (int j = 0; j < dataGridView1.ColumnCount; j++)  if (dataGridView1.Rows[i].Cells[j].Value != null)  if (dataGridView1.Rows[i].Cells[j].Value.ToString().Contains(textBox1.Text))  {  dataGridView1.Rows[i].Selected = true;  break;  }  }  } |

Свойства элементов формы «Гость» представлены в таблице 21.

Таблица 21 – Свойства элементов формы «Гость»

|  |  |  |
| --- | --- | --- |
| Объект | Свойство | Значение |
| Клиент | FormBorderStyle | SizableToolWindow |
| Text | Клиент |
| StartPosition | CenterScreen |
| Size | 803; 409 |
| ShowIcon | False |
| Вернуться\_ToolStripMenuItem2 | BackColor | Silver |
| Text | Вернуться |
| ForeColor | Black |
| Выйти\_ToolStripMenuItem11 | Size | 180; 28 |
| Text | Выход |
| Меню\_ToolStripMenuItem1 | BackColor | 128; 128; 255 |
| Text | Меню |
| ForeColor | ControlText |
| Поиск\_Label1 | BackColor | 128; 128; 255 |
| Size | 100; 52 |
| BorderStyle | None |
| Поле\_для\_таблиц\_DataGridView1 | Size | 757; 288 |
| BackColor | Control |
| Строка\_поиска\_TextBox1 | Font | Segoe Print; 7pt |
| Size | 181; 24 |
| Таблица\_Маршруты\_ToolStripMenuItem7 | BackColor | Silver |
| Text | Маршруты |
| ForeColor | ControlText |
| Таблица\_Пункты\_маршрута\_ToolStripMenuItem8 | BackColor | Silver |
| Text | Пункты маршрута |
| ForeColor | ControlText |
| Таблица\_Гостиница\_ToolStripMenuItem9 | BackColor | Silver |
| Text | Гостиница |
| ForeColor | ControlText |
| Таблица\_Экскурсии\_ToolStripMenuItem10 | BackColor | Silver |
| Text | Экскурсии |
| BackColor | Silver |

Событийно-управляемые процедуры в форме «Гость» представлены в таблице 22.

Таблица 22 – Событийно-управляемые процедуры в форме «Гость»

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| class Гость | //Добавляет таблицы  public void Tub(int i)  {  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП01\_ИСПП5\_Шипицына\_ОИ; Integrated Security = True");  sqlConnect.Open();  SqlDataAdapter[] so = new SqlDataAdapter[4];  so[0] = new SqlDataAdapter("select \* from Маршруты", sqlConnect);  so[1] = new SqlDataAdapter("select \* from [Пункты маршрута]", sqlConnect);  so[2] = new SqlDataAdapter("select \* from Гостиница", sqlConnect);  so[3] = new SqlDataAdapter("select \* from Экскурсии", sqlConnect);  DataSet ps = new DataSet();  so[i].Fill(ps);  dataGridView1.DataSource = ps.Tables[0];  } |
| Гость\_Load | //Загрузка данных в таблицы  private void Гость\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Гостиница". При необходимости она может быть перемещена или удалена.  this.гостиницаTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Гостиница);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Экскурсии". При необходимости она может быть перемещена или удалена.  this.экскурсииTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Экскурсии);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пункты\_маршрута". При необходимости она может быть перемещена или удалена.  this.пункты\_маршрутаTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пункты\_маршрута);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Маршруты". При необходимости она может быть перемещена или удалена.  this.маршрутыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Маршруты);  } |
| Вернуться\_toolStripMenuItem2\_Click | //Переходит на форму 'Авторизация'  private void toolStripMenuItem2\_Click(object sender, EventArgs e)  {  Авторизация AvtFrm = new Авторизация();  AvtFrm.Show();  this.Hide();  } |
| Выйти\_toolStripMenuItem4\_Click | private void toolStripMenuItem4\_Click(object sender, EventArgs e)  { //Прекращает работу программы  Application.Exit();  } |
| Таблица\_Маршруты\_toolStripMenuItem7\_Click | private void toolStripMenuItem7\_Click(object sender, EventArgs e)  {//Открывает таблицу 'Маршруты'  Tub(0);  t = 0;  a = true;  dataGridView1.ReadOnly = false;  } |
| Таблица\_Гостиница\_ toolStripMenuItem9\_Click | private void toolStripMenuItem9\_Click(object sender, EventArgs e)  {//Открывает таблицу 'Гостиница'  Tub(2);  t = 2;  a = true;  dataGridView1.ReadOnly = false;  } |
| Таблица\_Пункты\_маршрута\_ toolStripMenuItem8\_Click | private void toolStripMenuItem8\_Click(object sender, EventArgs e)  {//Открывает таблицу 'Пункты маршрута'  Tub(1);  t = 1;  a = true;  dataGridView1.ReadOnly = false;  } |
| Таблица\_Экскурсии \_ toolStripMenuItem10\_Click | private void toolStripMenuItem10\_Click(object sender, EventArgs e)  {//Открывает таблицу 'Экскурсии'  Tub(3);  t = 3;  a = true;  dataGridView1.ReadOnly = false;  } |
| Поисковая\_строка\_ textBox1\_TextChanged | private void textBox1\_TextChanged(object sender, EventArgs e)  {  //Выделение строки, которая подходит под введённые данные в поисковую строку  for (int i = 0; i < dataGridView1.RowCount; i++)  {  dataGridView1.Rows[i].Selected = false;  for (int j = 0; j < dataGridView1.ColumnCount; j++)  if (dataGridView1.Rows[i].Cells[j].Value != null)  if (dataGridView1.Rows[i].Cells[j].Value.ToString().Contains(textBox1.Text))  {  dataGridView1.Rows[i].Selected = true;  break;  }  }  } |

Вывод по главе: была полностью спроектирована и разработана система базы данных, были полностью спроектированы и разработаны все программные модули информационной системы «Туристическая компания».

1. РАЗРАБОТКА ТЕСТОВЫХ НАБОРОВ И ТЕСТОВЫХ СЦЕНАРИЕВ

Тестирование программного обеспечения – процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом.

Тестовый сценарий – это неавтоматизированный или автоматизированный сценарий, содержащий инструкции по реализации тестового набора. Тестовый сценарий может быть написан вручную (для выполнения человеком) либо полностью или частично автоматизирован.

* 1. Назначение эксперимента. Выбор и обоснование методики проведения тестирования

В данном разделе проведены два вида тестирования. Провести анализ тестирования и выбрать, как будет проходить тестирование ИС.

1. Модульное тестирование. Цель: проверить, что код работает именно так, как должен (при заданных входных параметрах выдает предполагаемый результат).
2. Анализ покрытия кода тестами. Цель: проверить, что весь код отрабатывает при модульном тестировании, что нет не участвующих в тестировании участков кода.
3. Анализ стилистики кода. Цель: проверка кода на соблюдение стандартов разработки (отраслевых, корпоративных, принятых на проекте).
4. Анализ производительности. Цель: проанализировать производительность ключевых операций, потребление памяти, утечку памяти.
   1. Технология тестирования

Рассмотрим, какие технологии можно применить для модульного тестирования и покрытия кода тестами.

Как в ходе выполнения проекта, так и при развитии системы, изменения в ее алгоритмы вносятся в соответствии с техническими заданиями, хотя данный документ может иметь различные названия. В технических заданиях, как правило, описываются критерии приемки: как мы и заказчик убедимся в том, что заказанный функционал работает верно.

В качестве примера можно привести такой упрощенный критерий:

* 1. Исходные данные: в систему внесен тур, который в дальнейшем приобретёт клиент.
  2. Действия пользователя: пользователь ищет в таблице маршрут тура, сколько он будет стоить и т п.
  3. В результате в результате отображена сумма тура, например 120000 руб.
  4. Вручную такое тестирование, как правило, выполняется единожды, поскольку выполнять каждый раз полный комплект тестов очень трудоемко.
  5. Нет гарантии того, что тесты, описанные в техническом задании, покрывают функциональность полностью. Т.е. даже если тесты описаны и проводятся регулярно, вероятно будет существовать наличие не протестированного функционала.

Автоматизированное тестирование лишено данных недостатков. Конечно, возникает отдельная задача кодирования данного теста на основании критериев приемки, но это скорее дисциплинирующий фактор…

Прежде чем начинать юзабилити-тестирование необходимо собрать группу пользователей, которые будут тестировать данную информационную систему. Количество привлеченных пользователей должно быть не менее пяти человек. Пользователям будет представлен список некоторых вопросов, относящихся к графическому интерфейсу.

В таблице 23 представлены сведения о пользователях.

Таблица 23 – Информация о пользователях

|  |  |  |
| --- | --- | --- |
| ФИО | Пол | Образование |
| Работягова Александра Андреевна | Женский | Общее образование |
| Попов Семен Евгеньевич | Мужской | Общее образование |
| Степурин Никита Евгеньевич | Мужской | Общее образование |
| Демешко Екатерина Олеговна | Женский | Общее образование |
| Дацюк Виктория Денисовна | Женский | Общее образование |

* 1. Результаты проведения тестирования

В ходе тестирования был применен метод нагрузочного тестирования и юзабилити-тестирования, которые проверяют работу при нагрузке системы и сделать простым и удобным в пользовании информационной системы.

Суть тестирования заключается в проверке работы всех обязательных функций.

Результаты проведения нагрузочного тестирования представлены в приложении 1 на рисунке 17.

Исходя из результатов нагрузочного тестирования, можно прийти к выводу, что нагрузка, оказываемая на центральный процессор и память, не имеет критичных значений. Это значит, что показатель производительности информационной системы на приемлемом уровне, то есть сбои и длительная загрузка не должны помешать пользователю при работе с данным программным продуктом.

Таблица 25 – Результаты проведения юзабилити-тестирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Вопрос | Работягова Александра Андреевна | Попов Семен Евгеньевич | Степурин Никита Евгеньевич | Демешко Екатерина Олеговна | Дацюк Виктория Денисовна |
| Присутствуют ли такие дизайн-элементы, которые неприятны для зрения? | нет | нет | нет | нет | нет |
| Присутствует ли отвлекающая информация? | нет | нет | нет | нет | нет |
| Понятна ли структура приложения на интуитивном уровне? | да | да | да | да | да |
| Смогли ли вы найти интересующую вас информацию без затруднений? | да | да | да | нет | да |
| Понятен ли смысл приложения? | да | да | да | да | да |
| Общая оценка | 8/10 | 7/10 | 9/10 | 8/10 | 7/10 |

Исходя из средней оценки информационной системы, рассчитанной на основе общих оценок всех пользователей с учетом ответов на все поставленные вопросы, можно сделать вывод о том, что внешние характеристики графического интерфейса находятся на довольно высоком уровне. Также в информационной системе нет помех для пользователя в виде отвлекающей информации, так как все отображаемые данные находятся в рамках предметной области.

Вывод по главе: были проведены тесты путём проведения модульного тестирования, анализа покрытия кода тестами, анализа стилистики кода и анализом производительности. Также было проведено нагрузочное тестирование, в результате чего нагрузка на процессор и память не имела критичных значений.

# ЗАКЛЮЧЕНИЕ

В результате учебной практики был проведен анализ предметной области, спроектирована и разработана информационная система «Туристическая компания», серверная часть которой была реализована в MS SQL, а клиентская часть в MS Visual Studio.

В MS SQL была создана база данных для хранения информации о клиентах, документах, контрактах, маршрутах, пунктах маршрута, гостиницах, экскурсиях.

Клиентская часть, реализованная в MS Visual Studio, отражает все хранящиеся в базе данных таблицы. Также разработаны некоторые запросы, позволяющие редактировать и удалять существующие данные, добавлять новые и осуществлять поиск по таблицам.

Цели учебной практики были достигнуты путем проведения анализа предметной области, проектирования информационной системы и разработки программных модулей.

Это позволяет сделать вывод, что поставленные цели и задачи выполнены в полном объеме.

В дальнейшем данная информационная система может быть модернизирована путем добавления формы личного кабинета, создания версии для слабовидящих людей или улучшения и изменения дизайна.

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

Основные источники

1. Фёдорова Г.Н. Основы проектирования баз данных (2-е изд., стер.) учебник \ Москва : Академия, 2018. - 219, [1] с. : ил. ; 22 см. - ISBN 978-5-4468-5800-2 (в пер.): Б. ц.
2. Гагарина, Л. Г. Технология разработки программного обеспечения: учеб. пособие / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Виснадул; Под ред. Л. Г. Гагариной. - М.: ФОРУМ: ИНФРА-М, 2017. -400 с.

Дополнительные источники

1. Федорова, Г.Н. Основы алгоритмизации и программирования: учебное пособие для студ. учреждений сред. проф. образования / Г.Н. Федорова. – М.: Издательский центр «Академия», 2014. – 224 с.
2. Подбельский В.В. Язык С#. Базовый курс. – М: Инфра, 2015.- 384 с.
3. Рудаков А. Технология разработки программных продуктов: учебник. Изд. Academia. Среднее профессиональное образование. 2013 г. 208 стр.
4. Федорова Г., Рудаков А. Технология разработки программных продуктов. Практикум: учебное пособие. Изд. Academia. Среднее профессиональное образование. 2012 г. 192 стр.
5. Орлов С.А., Цилькер Б.Я. Технологии разработки программного обеспечения: учебник. СПб: Питер. 2018, 609 стр.

Интернет – ресурсы

1. Разработка приложений для мобильных интеллектуальных систем на платформе Intel Atom [Электронный ресурс] : учебное пособие / К.C. Амелин [и др.]. — Электрон. дан. — Москва : , 2016. — 201 с. — Режим доступа:,https://e.lanbook.com/book/100461. — Заглавие с экрана. Яз. рус., англ.
2. Берлин, А.Н. Сотовые системы связи [Электронный ресурс] : учебное пособие / А.Н. Берлин. — Электрон. дан. — Москва : , 2016. — 430 с. — Режим доступа: https://e.lanbook.com/book/100494. — Заглавие с экрана. Яз. рус., англ.
3. Александров, Э.Э. Программирование на языке C в Microsoft Visual Studio 2010 [Электронный ресурс] : учебное пособие / Э.Э. Александров, В.В. Афонин. — Электрон. дан. — Москва : , 2016. — 570 с. — Режим доступа: https://e.lanbook.com/book/100410. — Заглавие. с экрана. Яз. рус., англ.
4. Галатенко, В.А. Мобильное программирование приложений реального времени в стандарте POSIX [Электронный ресурс] : учебное пособие / В.А. Галатенко. — Электрон. дан. — Москва : , 2016. — 479 с. — Режим доступа: https://e.lanbook.com/book/100656. — Заглавие. с экрана.
5. Снетков, В.М. Практикум прикладного программирования на C# в среде VS.NET 2008 [Электронный ресурс] : учебное пособие / В.М. Снетков. — Электрон. дан. — Москва : , 2016. — 1659 с. — Режим доступа: https://e.lanbook.com/book/100382. — Заглавие с экрана. Яз. рус., англ.
6. Мейер, Б. Основы объектно-ориентированного проектирования [Электронный ресурс] : учебник / Б. Мейер. — Электрон. дан. — Москва : , 2016. — 765 с. — Режим доступа: https://e.lanbook.com/book/100305. — Заглавие с экрана. Яз. рус., англ.
7. Тюкачев, Н.А. C#. Основы программирования [Электронный ресурс] : учебное пособие / Н.А. Тюкачев, В.Г. Хлебостроев. — Электрон. дан. — Санкт-Петербург : Лань, 2018. — 272 с. — Режим доступа: https://e.lanbook.com/book/104962. — Заглавие с экрана. Яз. рус., англ.

ПРИЛОЖЕНИЕ 1

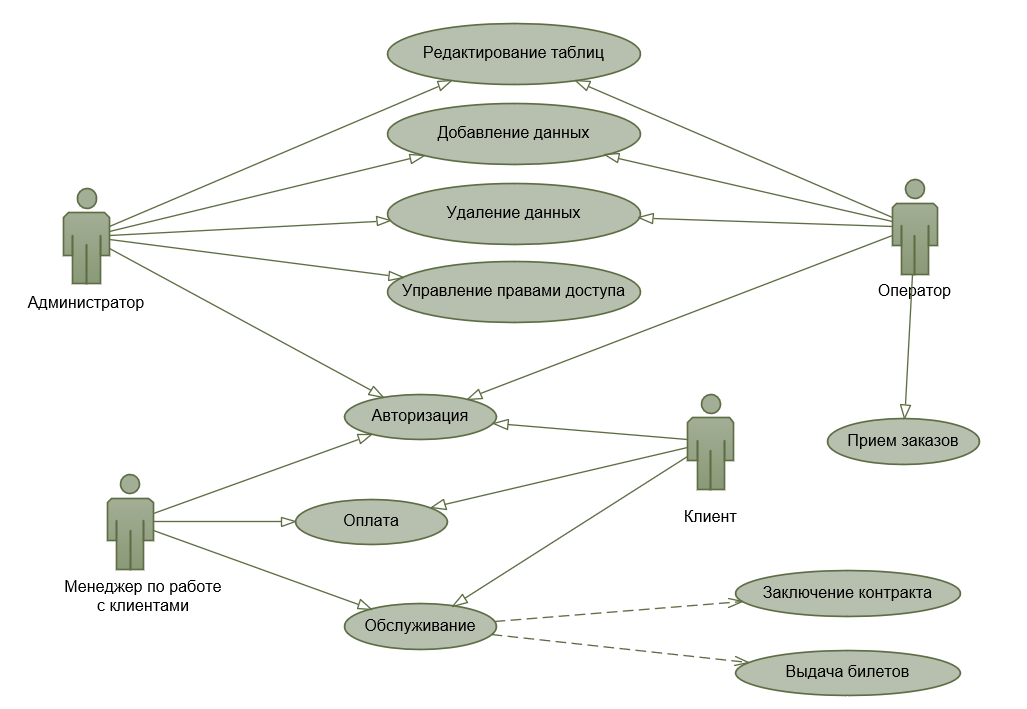


Рисунок 1 – Диаграмма вариантов использования

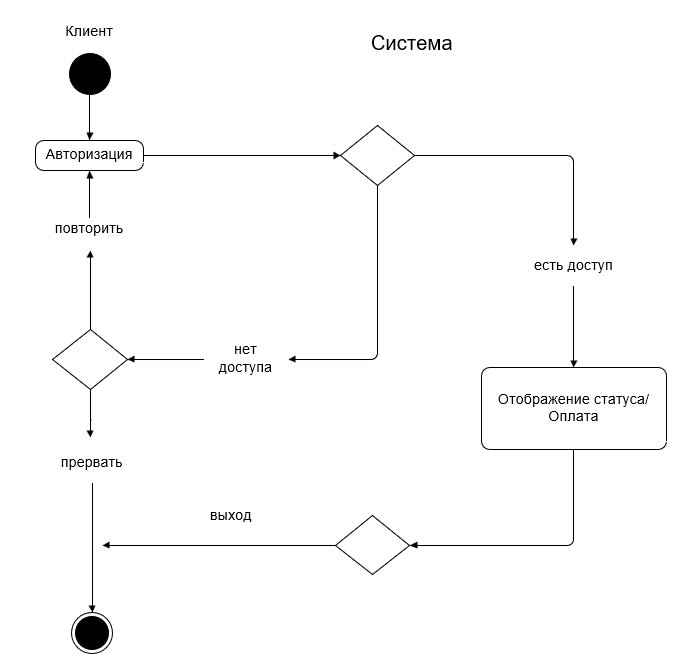


Рисунок 2 – Диаграмма деятельности «Клиент»

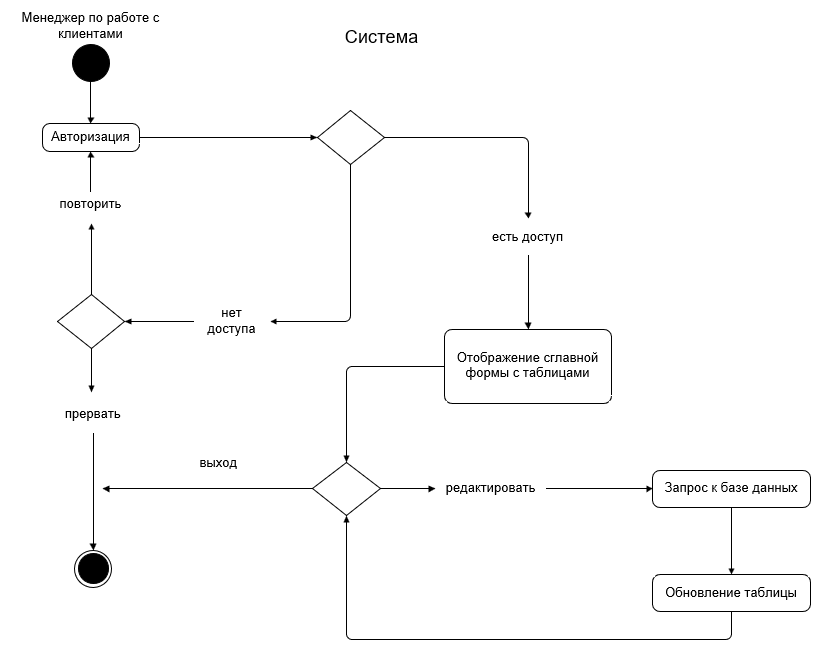


Рисунок 3 – Диаграмма деятельности «Менеджер по работе с клиентами»

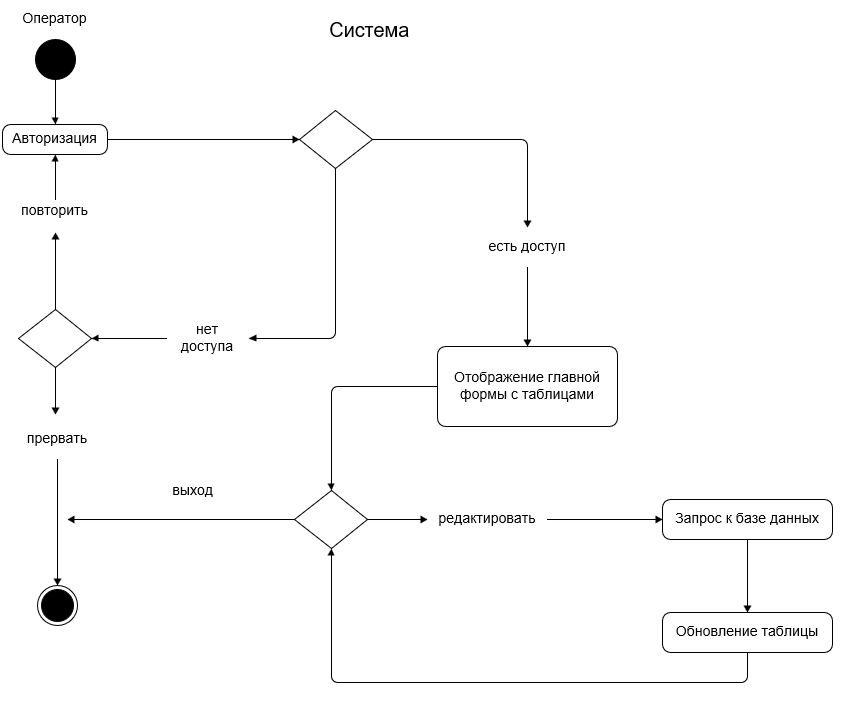
****

Рисунок 4 – Диаграмма деятельности «Оператор»

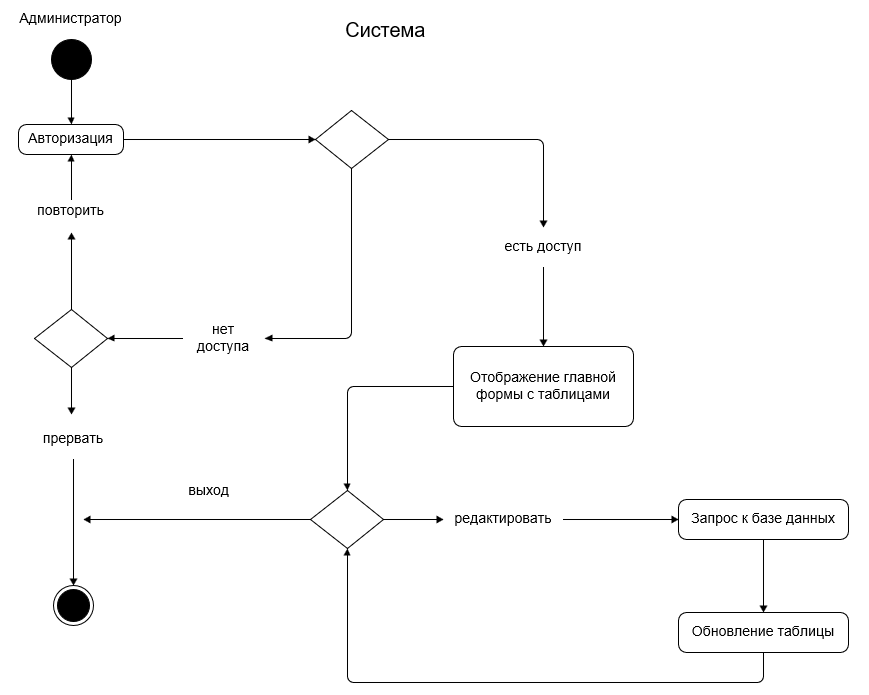


Рисунок 5 – Диаграмма деятельности «Администратор»

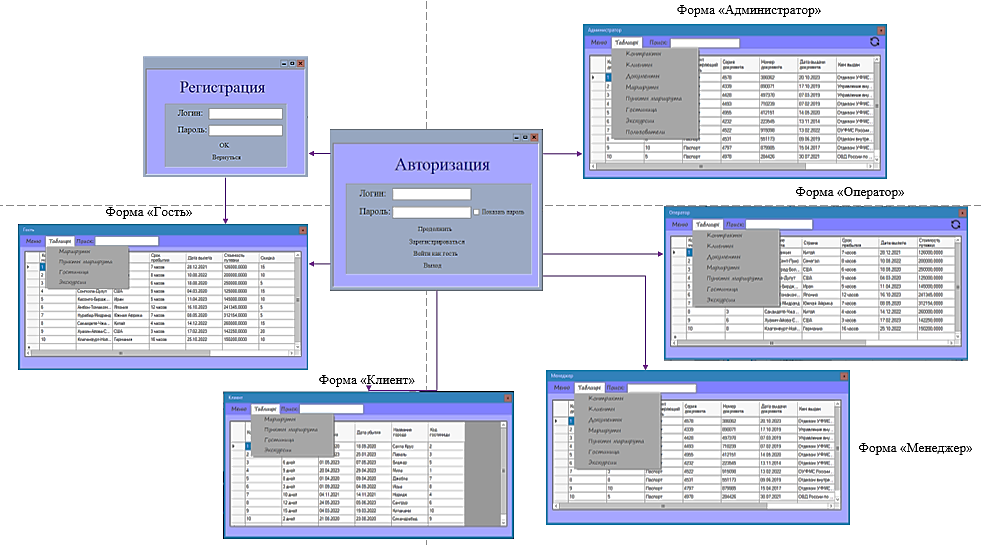


Рисунок 6 – Карта переходов

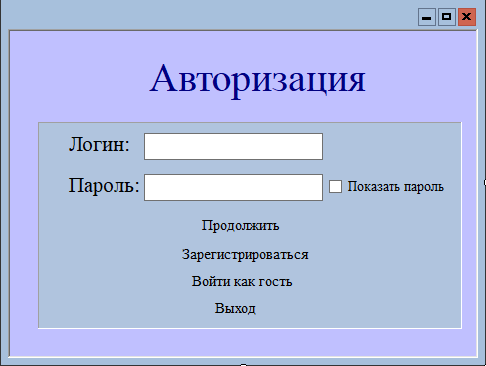


Рисунок 7 – Форма «Авторизация»

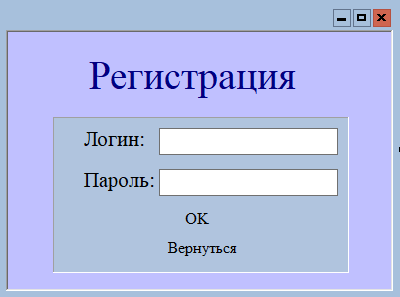


Рисунок 8 – Форма «Регистрация»

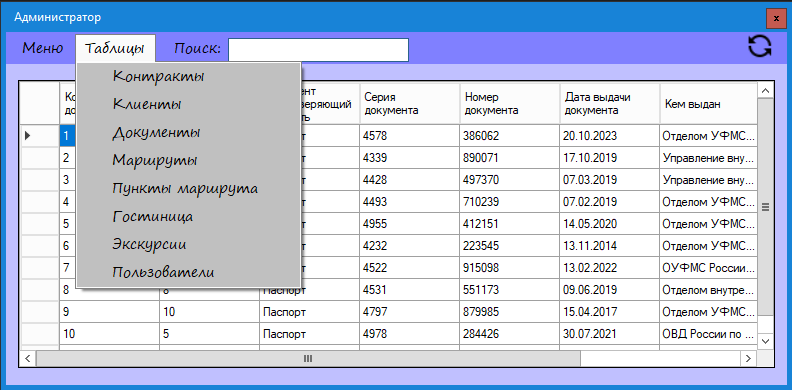


Рисунок 9 – Форма «Администратор»

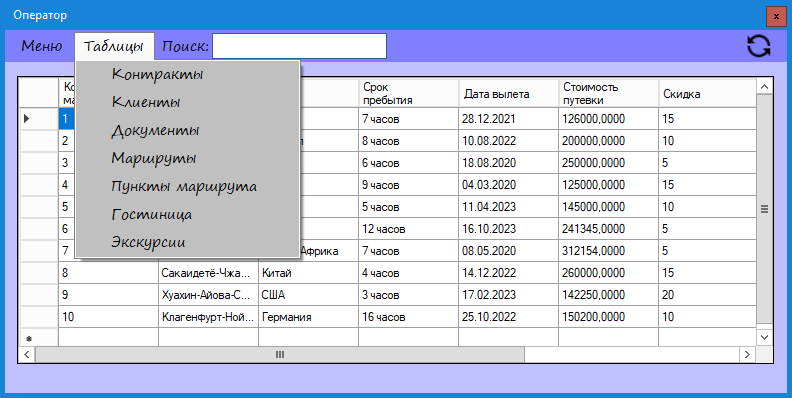


Рисунок 10 – Форма «Оператор»

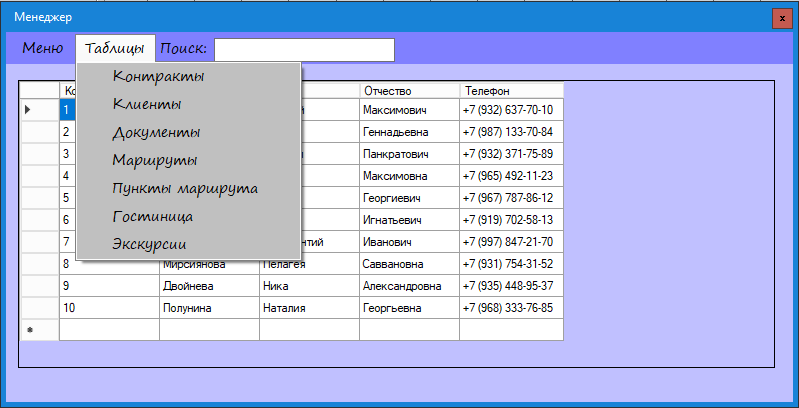


Рисунок 11 – Форма «Менеджера»

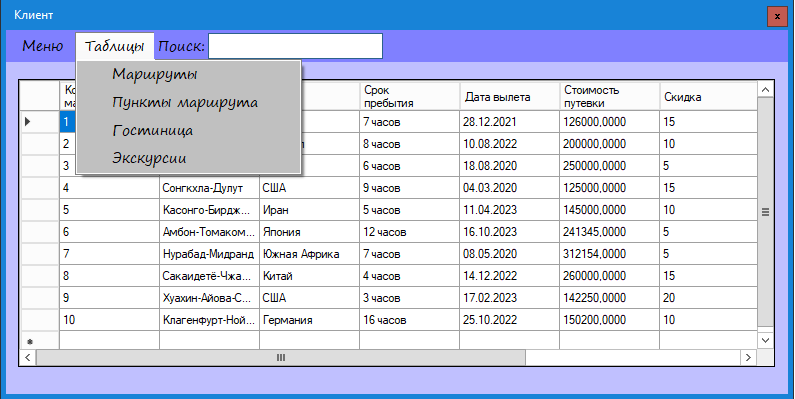


Рисунок 12 – Форма «Клиент»

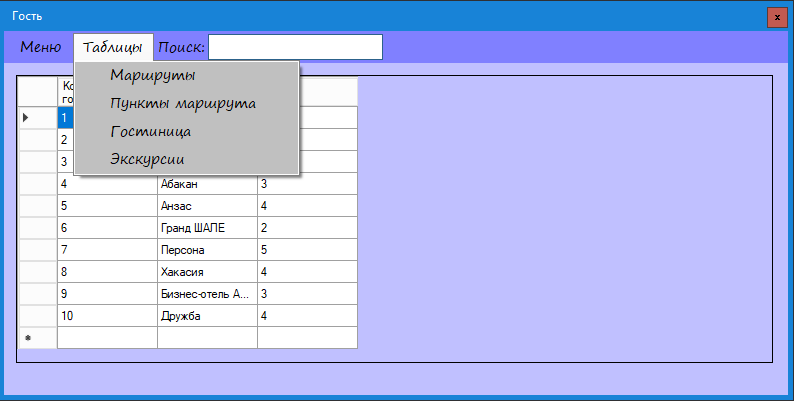
****

Рисунок 13 – Фома «Гость»

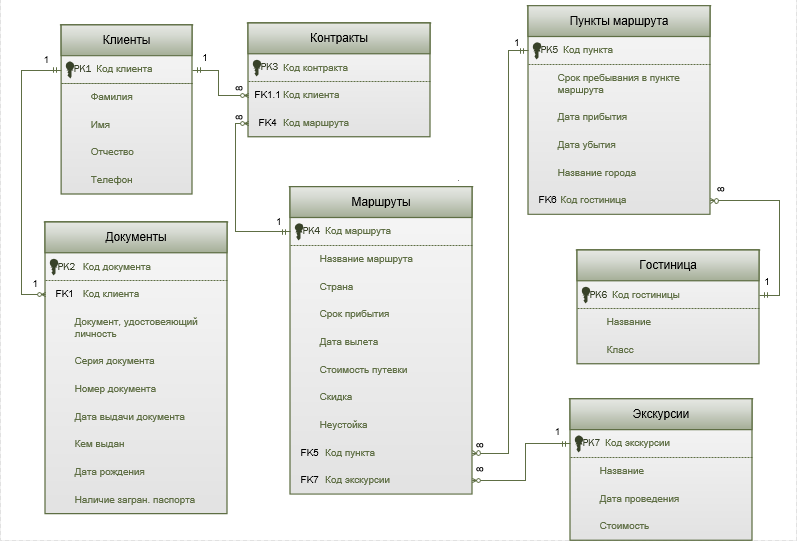


Рисунок 14 – Логическая модель данных

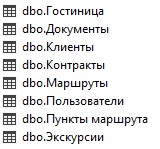


Рисунок 15 – Созданные таблицы

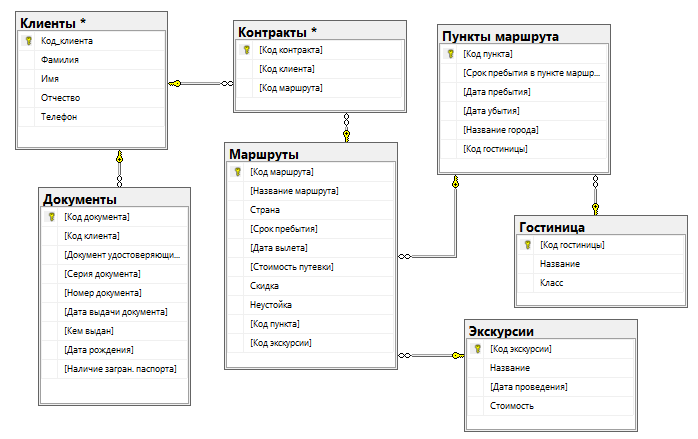


Рисунок 16 – Физическая модель данных

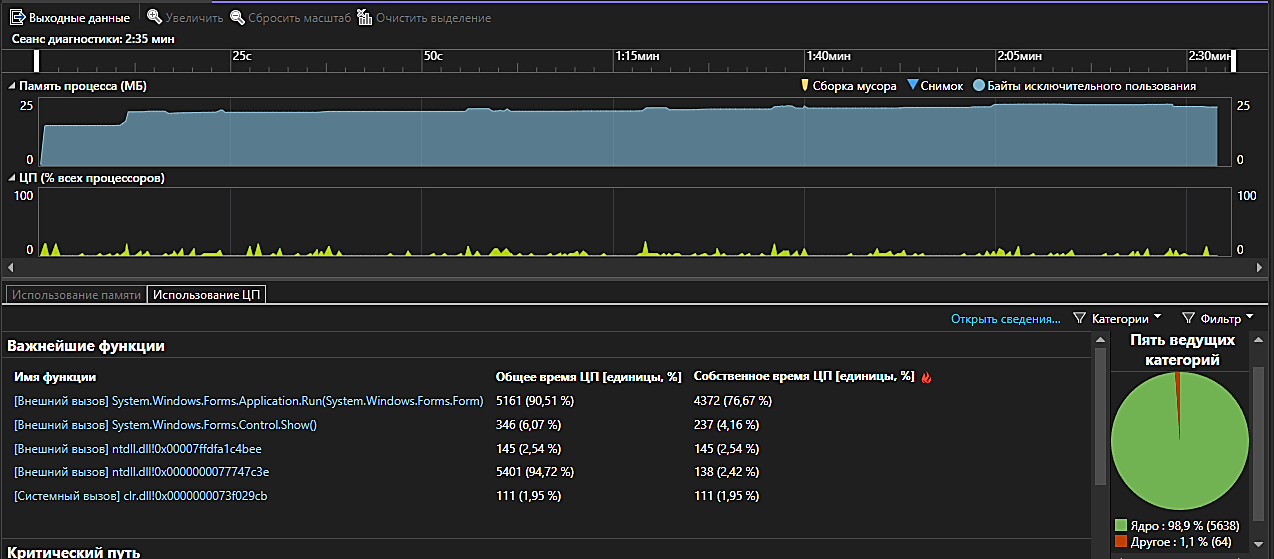


Рисунок 17 – Результаты нагрузочного тестирования

ПРИЛОЖЕНИЕ 2

Листинг кода формы «Авторизация»:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Туристическая\_компания1

{

public partial class Авторизация : Form

{

public Авторизация()

{

InitializeComponent();

}

int a = 0;

int popitki = 6;

private void textBox2\_KeyPress(object sender, KeyPressEventArgs e)

{

//Проверяет на отсутствие букв

if (!char.IsDigit(e.KeyChar) && (e.KeyChar != (char)Keys.Back))

{

e.Handled = true;

MessageBox.Show("Поле не может содержать буквы!");

}

}

private void label6\_Click(object sender, EventArgs e)

{

//Переходит на форму 'Администратор', 'Оператор', 'Менеджер', 'Клиент' и 'Гость'

if ((textBox1.Text == "") || (textBox2.Text == ""))

{

if ((textBox1.Text == "") && (textBox2.Text == "") && (textBox3.Text == ""))

{

MessageBox.Show("Заполните все поля!!!");

}

else

{

if ((textBox1.Text == "") && (textBox2.Text == ""))

{

MessageBox.Show("Введите Логин и Пароль!!!");

}

else

{

if (textBox1.Text == "")

{

MessageBox.Show("Введите Логин!!!");

}

else

{

if (textBox2.Text == "")

{

MessageBox.Show("Введите Пароль!!!");

}

}

}

}

}

else

{

SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = УП01\_ИСПП5\_Шипицына\_ОИ; Integrated Security = True");

sqlConnect.Open();

SqlDataAdapter da = new SqlDataAdapter("select \* from Пользователи", sqlConnect);

DataTable dt = new DataTable();

da.Fill(dt);

Boolean log = false, password = false;

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

if (dt.Rows[i]["Логин"].ToString() == textBox1.Text)

{

if (dt.Rows[i]["Пароль"].ToString() == textBox2.Text)

{

password = true;

textBox3.Text = dt.Rows[i]["Права доступа"].ToString();

}

log = true;

}

if ((password == true) && (log == true))

{

if (textBox3.Text == "Гость")

{

this.Hide();

Клиент UsrFrm = new Клиент();

UsrFrm.Show();

UsrFrm.ii("Гость");

}

if (textBox3.Text == "Клиент")

{

this.Hide();

Клиент UsrFrm = new Клиент();

UsrFrm.Show();

UsrFrm.ii("Клиент");

}

if (textBox3.Text == "Менеджер ")

{

this.Hide();

Менеджер UsrFrm = new Менеджер();

UsrFrm.Show();

UsrFrm.ii("Менеджер ");

}

if (textBox3.Text == "Администратор")

{

this.Hide();

Администратор UsrFrm = new Администратор();

UsrFrm.Show();

UsrFrm.ii("Администратор");

}

if (textBox3.Text == "Оператор")

{

this.Hide();

Оператор UsrFrm = new Оператор();

UsrFrm.Show();

UsrFrm.ii("Оператор");

}

}

else

{

if ((password == false) && (log == false))

{

MessageBox.Show("Данный пользователь отсутвует!!!");

}

else

{

if (password == false)

{

MessageBox.Show("Неверно введен Пароль!!!");

popitki -= 1;

label4.Text = "Количество попыток для ввода Пароля осталось: " + popitki;

if (popitki == 0)

{

label5.Visible = true;

label7.Visible = false;

label8.Location = new Point(113, 125);

label9.Visible = false;

}

}

}

}

}

}

private void label7\_Click(object sender, EventArgs e)

{

//Переходит на форму 'Регистрация'

this.Hide();

Регистрация UsrFrm4 = new Регистрация();

UsrFrm4.Show();

}

private void label8\_Click(object sender, EventArgs e)

{

//Переходит на форму 'Гость'

this.Hide();

Гость UsrFrm3 = new Гость();

UsrFrm3.Show();

}

private void label9\_Click(object sender, EventArgs e)

{

//Прекращает работу программы

Application.Exit();

}

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

//Видимость пароля

if (checkBox1.Checked)

{

textBox2.UseSystemPasswordChar = false;

}

else

{

textBox2.UseSystemPasswordChar = true;

}

}

}

}

Листинг кода формы «Регистрация»:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Туристическая\_компания1

{

public partial class Регистрация : Form

{

public Регистрация()

{

InitializeComponent();

}

private void textBox2\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!char.IsDigit(e.KeyChar) && (e.KeyChar != (char)Keys.Back))

{

e.Handled = true;

MessageBox.Show("Поле не может содержать буквы!");

}

}

private void Регистрация\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пользователи". При необходимости она может быть перемещена или удалена.

this.пользователиTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пользователи);

}

private void label5\_Click(object sender, EventArgs e)

{

if ((textBox2.Text == "") || (textBox1.Text == ""))

{

MessageBox.Show("Не все данные заполнены!", "Ошибка");

}

else

{

SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП01\_ИСПП5\_Шипицына\_ОИ; Integrated Security = True");

sqlConnect.Open();

DataTable dt = new DataTable();

DataTable dta = new DataTable();

SqlDataAdapter daa = new SqlDataAdapter($"select \* from Пользователи", sqlConnect);

daa.Fill(dta);

string n = dta.Rows[dta.Rows.Count - 1]["Id"].ToString();

SqlDataAdapter da = new SqlDataAdapter($"INSERT INTO Пользователи (Id,Логин,Пароль,[Права доступа]) VALUES ('{int.Parse(n) + 1}','{textBox1.Text}','{textBox2.Text}','Гость');", sqlConnect);

da.Fill(dt);

пользователиTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пользователи);

MessageBox.Show($"Регистрация пользователя с логином {textBox1.Text} выполнена.\nВы зарегистрированы как гость");

this.Hide();

Авторизация UsrFrm = new Авторизация();

UsrFrm.Show();

}

}

private void label6\_Click(object sender, EventArgs e)

{

//Переход на форму авторизации

Авторизация AvtFrm = new Авторизация();

AvtFrm.Show();

this.Hide();

}

}

}

Листинг кода формы «Администратор»:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Туристическая\_компания1

{

public partial class Администратор : Form

{

public Администратор()

{

InitializeComponent();

}

MaskedTextBox maskedTextBox;

string r;

public void ii(string b)

{

r = b;

i1();

}

public void i1()

{

if (r == "Администратор")

{

Text = "Администратор";

}

}

int c;

public void tab(int i)

{ //Добавляет таблицы

SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП01\_ИСПП5\_Шипицына\_ОИ; Integrated Security = True");

sqlConnect.Open();

switch (i)

{

case 0: dataGridView1.DataSource = контрактыBindingSource; break;

case 1: dataGridView1.DataSource = клиентыBindingSource; break;

case 2: dataGridView1.DataSource = документыBindingSource; break;

case 3: dataGridView1.DataSource = маршрутыBindingSource; break;

case 4: dataGridView1.DataSource = пунктыМаршрутаBindingSource; break;

case 5: dataGridView1.DataSource = гостиницаBindingSource; break;

case 6: dataGridView1.DataSource = экскурсииBindingSource; break;

case 7: dataGridView1.DataSource = пользователиBindingSource; break;

}

}

private void toolStripMenuItem2\_Click(object sender, EventArgs e)

{ //Переходит на форму 'Авторизация'

Авторизация AvtFrm = new Авторизация();

AvtFrm.Show();

this.Hide();

}

private void toolStripMenuItem4\_Click(object sender, EventArgs e)

{ //Открывает таблицу 'Контракты'

tab(0);

c = 0;

}

private void Администратор1\_Load(object sender, EventArgs e)

{ //Загрузка данных в таблицы

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Экскурсии". При необходимости она может быть перемещена или удалена.

this.экскурсииTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Экскурсии);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пункты\_маршрута". При необходимости она может быть перемещена или удалена.

this.пункты\_маршрутаTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пункты\_маршрута);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пользователи". При необходимости она может быть перемещена или удалена.

this.пользователиTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пользователи);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Маршруты". При необходимости она может быть перемещена или удалена.

this.маршрутыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Маршруты);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Контракты". При необходимости она может быть перемещена или удалена.

this.контрактыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Контракты);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Клиенты". При необходимости она может быть перемещена или удалена.

this.клиентыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Клиенты);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Документы". При необходимости она может быть перемещена или удалена.

this.документыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Документы);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Гостиница". При необходимости она может быть перемещена или удалена.

this.гостиницаTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Гостиница);

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{ //Выделение строки, которая подходит под введённые данные в поисковую строку

for (int i = 0; i < dataGridView1.RowCount; i++)

{

dataGridView1.Rows[i].Selected = false;

for (int j = 0; j < dataGridView1.ColumnCount; j++)

if (dataGridView1.Rows[i].Cells[j].Value != null)

if (dataGridView1.Rows[i].Cells[j].Value.ToString().Contains(textBox1.Text))

{

dataGridView1.Rows[i].Selected = true;

break;

}

}

}

private void toolStripMenuItem5\_Click(object sender, EventArgs e)

{ //Открывает таблицу 'Клиенты'

tab(1);

c = 1;

}

private void toolStripMenuItem6\_Click(object sender, EventArgs e)

{ //Открывает таблицу 'Документы'

tab(2);

c = 2;

}

private void toolStripMenuItem7\_Click(object sender, EventArgs e)

{ //Открывает таблицу 'Маршруты'

tab(3);

c = 3;

}

private void toolStripMenuItem8\_Click(object sender, EventArgs e)

{ //Открывает таблицу 'Пункты маршрута'

tab(4);

c = 4;

}

private void toolStripMenuItem9\_Click(object sender, EventArgs e)

{ //Открывает таблицу 'Гостиница'

tab(5);

c = 5;

}

private void toolStripMenuItem10\_Click(object sender, EventArgs e)

{ //Открывает таблицу 'Экскурсии'

tab(6);

c = 6;

}

private void toolStripMenuItem11\_Click(object sender, EventArgs e)

{ //Открывает таблицу 'Пользователи'

tab(7);

c = 7;

}

private void pictureBox3\_Click(object sender, EventArgs e)

{ //Обновляет и сохраняет изменениям в таблицах

switch (c)

{

case 0: контрактыTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Контракты); break;

case 1: клиентыTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Клиенты); break;

case 2: документыTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Документы); break;

case 3: маршрутыTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Маршруты); break;

case 4: пункты\_маршрутаTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пункты\_маршрута); break;

case 5: гостиницаTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Гостиница); break;

case 6: экскурсииTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Экскурсии); break;

case 7: пользователиTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пользователи); break;

}

}

private void toolStripMenuItem12\_Click(object sender, EventArgs e)

{ //Прекращает работу программы

Application.Exit();

}

}

}

Листинг кода формы «Оператор»:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Туристическая\_компания1

{

public partial class Оператор : Form

{

public Оператор()

{

InitializeComponent();

}

string r;

public void ii(string b)

{

r = b;

i1();

}

public void i1()

{

if (r == "Оператор")

{

Text = "Оператор";

}

}

int c;

public void tab(int i)

{ //Добавляет таблицы

SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП01\_ИСПП5\_Шипицына\_ОИ; Integrated Security = True");

sqlConnect.Open();

switch (i)

{

case 0: dataGridView1.DataSource = контрактыBindingSource; break;

case 1: dataGridView1.DataSource = клиентыBindingSource; break;

case 2: dataGridView1.DataSource = документыBindingSource; break;

case 3: dataGridView1.DataSource = маршрутыBindingSource; break;

case 4: dataGridView1.DataSource = пунктыМаршрутаBindingSource; break;

case 5: dataGridView1.DataSource = гостиницаBindingSource; break;

case 6: dataGridView1.DataSource = экскурсииBindingSource; break;

}

}

private void toolStripMenuItem2\_Click(object sender, EventArgs e)

{ //Переходит на форму 'Авторизация'

Авторизация AvtFrm = new Авторизация();

AvtFrm.Show();

this.Hide();

}

private void Оператор\_Load(object sender, EventArgs e)

{ //Загрузка данных в таблицы

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Экскурсии". При необходимости она может быть перемещена или удалена.

this.экскурсииTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Экскурсии);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пункты\_маршрута". При необходимости она может быть перемещена или удалена.

this.пункты\_маршрутаTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пункты\_маршрута);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Маршруты". При необходимости она может быть перемещена или удалена.

this.маршрутыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Маршруты);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Клиенты". При необходимости она может быть перемещена или удалена.

this.клиентыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Клиенты);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Контракты". При необходимости она может быть перемещена или удалена.

this.контрактыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Контракты);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Документы". При необходимости она может быть перемещена или удалена.

this.документыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Документы);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Гостиница". При необходимости она может быть перемещена или удалена.

this.гостиницаTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Гостиница);

}

private void pictureBox1\_Click(object sender, EventArgs e)

{ //Обновляет и сохраняет изменениям в таблицах

switch (c)

{

case 0: контрактыTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Контракты); break;

case 1: клиентыTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Клиенты); break;

case 2: документыTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Документы); break;

case 3: маршрутыTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Маршруты); break;

case 4: пункты\_маршрутаTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пункты\_маршрута); break;

case 5: гостиницаTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Гостиница); break;

case 6: экскурсииTableAdapter.Update(уП01\_ИСПП5\_Шипицына\_ОИDataSet.Экскурсии); break;

}

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{ //Выделение строки, которая подходит под введённые данные в поисковую строку

for (int i = 0; i < dataGridView1.RowCount; i++)

{

dataGridView1.Rows[i].Selected = false;

for (int j = 0; j < dataGridView1.ColumnCount; j++)

if (dataGridView1.Rows[i].Cells[j].Value != null)

if (dataGridView1.Rows[i].Cells[j].Value.ToString().Contains(textBox1.Text))

{

dataGridView1.Rows[i].Selected = true;

break;

}

}

}

private void toolStripMenuItem4\_Click(object sender, EventArgs e)

{ //Открывает таблицу 'Контракты'

tab(0);

c = 0;

}

private void toolStripMenuItem5\_Click(object sender, EventArgs e)

{ //Открывает таблицу 'Клиенты'

tab(1);

c = 1;

}

private void toolStripMenuItem6\_Click(object sender, EventArgs e)

{ //Открывает таблицу 'Документы'

tab(2);

c = 2;

}

private void toolStripMenuItem7\_Click(object sender, EventArgs e)

{ //Открывает таблицу 'Маршруты'

tab(3);

c = 3;

}

private void toolStripMenuItem8\_Click(object sender, EventArgs e)

{ //Открывает таблицу 'Пункты маршрута'

tab(4);

c = 4;

}

private void toolStripMenuItem9\_Click(object sender, EventArgs e)

{ //Открывает таблицу 'Гостиница'

tab(5);

c = 5;

}

private void toolStripMenuItem10\_Click(object sender, EventArgs e)

{ //Открывает таблицу 'Экскурсии'

tab(6);

c = 6;

}

private void toolStripMenuItem11\_Click(object sender, EventArgs e)

{ //Прекращает работу программы

Application.Exit();

}

}

}

Листинг кода формы «Менеджер»:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Туристическая\_компания1

{

public partial class Менеджер : Form

{

public Менеджер()

{

InitializeComponent();

}

string r;

public void ii(string b)

{

r = b;

i1();

}

public void i1()

{

if (r == "Менеджер")

{

Text = "Менеджер";

}

}

public void Tub(int i)

{ //Добавляет таблицы

SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП01\_ИСПП5\_Шипицына\_ОИ; Integrated Security = True");

sqlConnect.Open();

SqlDataAdapter[] so = new SqlDataAdapter[7];

so[0] = new SqlDataAdapter("select \* from Контракты", sqlConnect);

so[1] = new SqlDataAdapter("select \* from Клиенты", sqlConnect);

so[2] = new SqlDataAdapter("select \* from Документы", sqlConnect);

so[3] = new SqlDataAdapter("select \* from Маршруты", sqlConnect);

so[4] = new SqlDataAdapter("select \* from [Пункты маршрута]", sqlConnect);

so[5] = new SqlDataAdapter("select \* from Гостиница", sqlConnect);

so[6] = new SqlDataAdapter("select \* from Экскурсии", sqlConnect);

DataSet ps = new DataSet();

so[i].Fill(ps);

dataGridView1.DataSource = ps.Tables[0];

}

public Boolean a = false;

public Boolean b = false;

int t;

private void Менеджер\_Load(object sender, EventArgs e)

{ //Загрузка данных в таблицы

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Экскурсии". При необходимости она может быть перемещена или удалена.

this.экскурсииTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Экскурсии);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пункты\_маршрута". При необходимости она может быть перемещена или удалена.

this.пункты\_маршрутаTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пункты\_маршрута);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Маршруты". При необходимости она может быть перемещена или удалена.

this.маршрутыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Маршруты);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Контракты". При необходимости она может быть перемещена или удалена.

this.контрактыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Контракты);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Клиенты". При необходимости она может быть перемещена или удалена.

this.клиентыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Клиенты);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Документы". При необходимости она может быть перемещена или удалена.

this.документыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Документы);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Гостиница". При необходимости она может быть перемещена или удалена.

this.гостиницаTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Гостиница);

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{ //Выделение строки, которая подходит под введённые данные в поисковую строку

for (int i = 0; i < dataGridView1.RowCount; i++)

{

dataGridView1.Rows[i].Selected = false;

for (int j = 0; j < dataGridView1.ColumnCount; j++)

if (dataGridView1.Rows[i].Cells[j].Value != null)

if (dataGridView1.Rows[i].Cells[j].Value.ToString().Contains(textBox1.Text))

{

dataGridView1.Rows[i].Selected = true;

break;

}

}

}

private void toolStripMenuItem11\_Click\_1(object sender, EventArgs e)

{ //Открывает таблицу 'Контракты'

Tub(0);

t = 0;

a = true;

dataGridView1.AllowUserToAddRows = false;

dataGridView1.AllowUserToDeleteRows = false;

dataGridView1.ReadOnly = false;

}

private void toolStripMenuItem2\_Click\_1(object sender, EventArgs e)

{ //Переходит на форму 'Авторизация'

Авторизация AvtFrm = new Авторизация();

AvtFrm.Show();

this.Hide();

}

private void toolStripMenuItem12\_Click(object sender, EventArgs e)

{ //Прекращает работу программы

Application.Exit();

}

private void toolStripMenuItem13\_Click(object sender, EventArgs e)

{ //Открывает таблицу 'Клиенты'

Tub(1);

t = 1;

a = true;

dataGridView1.AllowUserToAddRows = false;

dataGridView1.AllowUserToDeleteRows = false;

dataGridView1.ReadOnly = false;

}

private void toolStripMenuItem14\_Click(object sender, EventArgs e)

{ //Открывает таблицу 'Документы'

Tub(2);

t = 2;

a = true;

dataGridView1.AllowUserToAddRows = false;

dataGridView1.AllowUserToDeleteRows = false;

dataGridView1.ReadOnly = false;

}

private void toolStripMenuItem15\_Click(object sender, EventArgs e)

{ //Открывает таблицу 'Маршруты'

Tub(3);

t = 3;

a = true;

dataGridView1.AllowUserToAddRows = false;

dataGridView1.AllowUserToDeleteRows = false;

dataGridView1.ReadOnly = false;

}

private void toolStripMenuItem16\_Click(object sender, EventArgs e)

{ //Открывает таблицу 'Пункты маршрута'

Tub(4);

t = 4;

a = true;

dataGridView1.AllowUserToAddRows = false;

dataGridView1.AllowUserToDeleteRows = false;

dataGridView1.ReadOnly = false;

}

private void toolStripMenuItem17\_Click(object sender, EventArgs e)

{ //Открывает таблицу 'Гостиница'

Tub(5);

t = 5;

a = true;

dataGridView1.AllowUserToAddRows = false;

dataGridView1.AllowUserToDeleteRows = false;

dataGridView1.ReadOnly = false;

}

private void toolStripMenuItem18\_Click(object sender, EventArgs e)

{ //Открывает таблицу 'Экскурсии'

Tub(6);

t = 6;

a = true;

dataGridView1.AllowUserToAddRows = false;

dataGridView1.AllowUserToDeleteRows = false;

dataGridView1.ReadOnly = false;

}

}

}

Листинг кода формы «Клиент»:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Туристическая\_компания1

{

public partial class Клиент : Form

{

public Клиент()

{

InitializeComponent();

}

string r;

public void ii(string b)

{

r = b;

i1();

}

public void i1()

{

if (r == "Клиент")

{

Text = "Клиент";

}

}

public void Tub(int i)

{ //Добавляет таблицы

SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП01\_ИСПП5\_Шипицына\_ОИ; Integrated Security = True");

sqlConnect.Open();

SqlDataAdapter[] so = new SqlDataAdapter[4];

so[0] = new SqlDataAdapter("select \* from Маршруты", sqlConnect);

so[1] = new SqlDataAdapter("select \* from [Пункты маршрута]", sqlConnect);

so[2] = new SqlDataAdapter("select \* from Гостиница", sqlConnect);

so[3] = new SqlDataAdapter("select \* from Экскурсии", sqlConnect);

DataSet ps = new DataSet();

so[i].Fill(ps);

dataGridView1.DataSource = ps.Tables[0];

}

public Boolean a = false;

public Boolean b = false;

int t;

private void toolStripMenuItem2\_Click(object sender, EventArgs e)

{//Переходит на форму 'Авторизация'

Авторизация AvtFrm = new Авторизация();

AvtFrm.Show();

this.Hide();

}

private void toolStripMenuItem4\_Click(object sender, EventArgs e)

{//Прекращает работу программы

Application.Exit();

}

private void Клиент\_Load(object sender, EventArgs e)

{//Загрузка данных в таблицы

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Экскурсии". При необходимости она может быть перемещена или удалена.

this.экскурсииTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Экскурсии);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Гостиница". При необходимости она может быть перемещена или удалена.

this.гостиницаTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Гостиница);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пункты\_маршрута". При необходимости она может быть перемещена или удалена.

this.пункты\_маршрутаTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пункты\_маршрута);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Маршруты". При необходимости она может быть перемещена или удалена.

this.маршрутыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Маршруты);

}

private void toolStripMenuItem7\_Click(object sender, EventArgs e)

{ //Открывает таблицу 'Маршруты'

Tub(0);

t = 0;

a = true;

dataGridView1.AllowUserToAddRows = false;

dataGridView1.AllowUserToDeleteRows = false;

dataGridView1.ReadOnly = false;

}

private void toolStripMenuItem8\_Click(object sender, EventArgs e)

{ // Открывает таблицу 'Пункты маршрута'

Tub(1);

t = 1;

a = true;

dataGridView1.AllowUserToAddRows = false;

dataGridView1.AllowUserToDeleteRows = false;

dataGridView1.ReadOnly = false;

}

private void toolStripMenuItem9\_Click(object sender, EventArgs e)

{ //Открывает таблицу 'Гостиница'

Tub(2);

t = 2;

a = true;

dataGridView1.AllowUserToAddRows = false;

dataGridView1.AllowUserToDeleteRows = false;

dataGridView1.ReadOnly = false;

}

private void toolStripMenuItem10\_Click(object sender, EventArgs e)

{ //Открывает таблицу 'Экскурсии'

Tub(3);

t = 3;

a = true;

dataGridView1.AllowUserToAddRows = false;

dataGridView1.AllowUserToDeleteRows = false;

dataGridView1.ReadOnly = false;

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{ //Выделение строки, которая подходит под введённые данные в поисковую строку

for (int i = 0; i < dataGridView1.RowCount; i++)

{

dataGridView1.Rows[i].Selected = false;

for (int j = 0; j < dataGridView1.ColumnCount; j++)

if (dataGridView1.Rows[i].Cells[j].Value != null)

if (dataGridView1.Rows[i].Cells[j].Value.ToString().Contains(textBox1.Text))

{

dataGridView1.Rows[i].Selected = true;

break;

}

}

}

}

}

Листинг кода формы «Гость»:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Туристическая\_компания1

{

public partial class Гость : Form

{

public Гость()

{

InitializeComponent();

}

string r;

public void ii(string b)

{

r = b;

i1();

}

public void i1()

{

if (r == "Гость")

{

Text = "Гость";

}

}

//Добавляет таблицы

public void Tub(int i)

{

SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП01\_ИСПП5\_Шипицына\_ОИ; Integrated Security = True");

sqlConnect.Open();

SqlDataAdapter[] so = new SqlDataAdapter[4];

so[0] = new SqlDataAdapter("select \* from Маршруты", sqlConnect);

so[1] = new SqlDataAdapter("select \* from [Пункты маршрута]", sqlConnect);

so[2] = new SqlDataAdapter("select \* from Гостиница", sqlConnect);

so[3] = new SqlDataAdapter("select \* from Экскурсии", sqlConnect);

DataSet ps = new DataSet();

so[i].Fill(ps);

dataGridView1.DataSource = ps.Tables[0];

}

public Boolean a = false;

public Boolean b = false;

int t;

private void toolStripMenuItem4\_Click(object sender, EventArgs e)

{ //Прекращает работу программы

Application.Exit();

}

//Загрузка данных в таблицы

private void Гость\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Гостиница". При необходимости она может быть перемещена или удалена.

this.гостиницаTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Гостиница);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Экскурсии". При необходимости она может быть перемещена или удалена.

this.экскурсииTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Экскурсии);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пункты\_маршрута". При необходимости она может быть перемещена или удалена.

this.пункты\_маршрутаTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Пункты\_маршрута);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Шипицына\_ОИDataSet.Маршруты". При необходимости она может быть перемещена или удалена.

this.маршрутыTableAdapter.Fill(this.уП01\_ИСПП5\_Шипицына\_ОИDataSet.Маршруты);

}

private void toolStripMenuItem2\_Click(object sender, EventArgs e)

{//Переходит на форму 'Авторизация'

Авторизация AvtFrm = new Авторизация();

AvtFrm.Show();

this.Hide();

}

private void toolStripMenuItem7\_Click(object sender, EventArgs e)

{//Открывает таблицу 'Маршруты'

Tub(0);

t = 0;

a = true;

dataGridView1.AllowUserToAddRows = false;

dataGridView1.AllowUserToDeleteRows = false;

dataGridView1.ReadOnly = false;

}

private void toolStripMenuItem8\_Click(object sender, EventArgs e)

{//Открывает таблицу 'Пункты маршрута'

Tub(1);

t = 1;

a = true;

dataGridView1.AllowUserToAddRows = false;

dataGridView1.AllowUserToDeleteRows = false;

dataGridView1.ReadOnly = false;

}

private void toolStripMenuItem9\_Click(object sender, EventArgs e)

{//Открывает таблицу 'Гостиница'

Tub(2);

t = 2;

a = true;

dataGridView1.AllowUserToAddRows = false;

dataGridView1.AllowUserToDeleteRows = false;

dataGridView1.ReadOnly = false;

}

private void toolStripMenuItem10\_Click(object sender, EventArgs e)

{//Открывает таблицу 'Экскурсии'

Tub(3);

t = 3;

a = true;

dataGridView1.AllowUserToAddRows = false;

dataGridView1.AllowUserToDeleteRows = false;

dataGridView1.ReadOnly = false;

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

//Выделение строки, которая подходит под введённые данные в поисковую строку

for (int i = 0; i < dataGridView1.RowCount; i++)

{

dataGridView1.Rows[i].Selected = false;

for (int j = 0; j < dataGridView1.ColumnCount; j++)

if (dataGridView1.Rows[i].Cells[j].Value != null)

if (dataGridView1.Rows[i].Cells[j].Value.ToString().Contains(textBox1.Text))

{

dataGridView1.Rows[i].Selected = true;

break;

}

}

}

}

}