Министерство образования Московской области

Государственное образовательное учреждение высшего образования Московской области

«Государственный гуманитарно-технологический университет»

**Ликино-Дулевский политехнический колледж – филиал ГГТУ**

**КУРСОВОЙ ПРОЕКТ**

«Разработка приложения для оформления регистрации по месту жительства»

МДК 02.01 «Технология разработки программного обеспечения»

**Выполнил:**

Абуталыблы Ильяс Ядигар оглы

Студент 4 курса группы ИСП.20А

09.02.07 Информационные системы и программирование

очной формы обучения

**Руководитель:**

Селиверстова Ольга Михайловна

Оценка

\_\_\_\_\_\_\_\_\_/\_\_\_­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

Подпись руководителя

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

Ликино-Дулево

2024 год

**Оглавление**

[Введение 2](#_Toc154041397)

[1. Назначение разработки 5](#_Toc154041398)

[1.2 Требования к программе 5](#_Toc154041399)

[1.2.1 Требования к функциональным характеристикам 5](#_Toc154041400)

[1.2.2 Требования к надежности и безопасности 6](#_Toc154041401)

[1.2.3 Требования к составу и параметрам технических средств 6](#_Toc154041402)

[1.2.4 Требования к информационной и программной совместимости 6](#_Toc154041403)

[2. Разработка технического проекта 7](#_Toc154041404)

[2.1. Обоснование выбора CASE – средств 7](#_Toc154041405)

[2.2. Проектирование модели данных 7](#_Toc154041406)

[2.2.1 CASE – средство MS Visio 7](#_Toc154041407)

[3. Реализация 11](#_Toc154041408)

[3.1 Обоснование выбора средств разработки 11](#_Toc154041409)

[3.2 Руководство программиста 11](#_Toc154041410)

[3.3 Руководство пользователя 14](#_Toc154041411)

[4. Тестирование и откладка 20](#_Toc154041412)

[5. Методы и средства проведения расчётов оценки трудоёмкости разработки проекта (или Методы и средства защиты БД) 24](#_Toc154041413)

[Заключение 24](#_Toc154041414)

[Приложение 25](#_Toc154041415)

[Список литературы 27](#_Toc154041416)

**Введение**

В наше время электронные государственные сервисы становятся все более популярными и удобными для граждан. Однако процесс оформления регистрации по месту жительства все еще остается достаточно сложным и требует проведения многочисленных процедур и походов в муниципальные органы. С целью упрощения и ускорения этого процесса, существует потребность в разработке приложения, которое позволит гражданам оформлять регистрацию по месту жительства через Многофункциональный центр предоставления государственных услуг (МФЦ), с использованием электронной подачи заявлений и проверки необходимых документов. В данном курсовом проекте будет представлена разработка такого приложения, с учетом требований и потребностей пользователей, а также с обеспечением безопасности и защиты персональных данных. Курсовая работа по учету регистрации жителей по месту жительства является актуальной на нынешний момент.

Целью курсовой работы является изучение предметной области в сфере Регистрации жителей по месту жительства, а также разработка автоматизированного приложения для работников различных ролей в соответствии с их должностью.

При создании проекта использовалось следующее ПО:

• Visual Studio – IDE, в которой разрабатывалось приложение по работе с недвижимости для удобства работы риелторов и клиентов;

• Microsoft SQL – система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft;

• Microsoft Word – использовалось в ведении отчетности

• Draw.io — инструмент для создания диаграмм, блок-схем, интеллект-карт, бизнес-макетов, отношений сущностей, программных блоков и другого.

**1. Назначение разработки**

Автоматизированная информационная система «MFC» предназначена для оказания услуг клиенту по регистрации пользователя по указанному адресу. Пользователями программы выступает сотрудник центра. Регистрация пользователя осуществляется на основании договоров Регистрация гражданина Российской Федерации по месту жительства, в которых оговариваются условия регистрации. Акте перерегистрации указываются: первый адрес, где клиент прописан на данный момент и конечный адрес куда он будет регистрироваться.

Данные первичных документов фиксируются в карточках учета, которые выполняют роль регистров регистрационного учета.

# 1.2 Требования к программе

## 1.2.1 Требования к функциональным характеристикам

Функциональные требования:

* Регистрация пользователя.
* Авторизация пользователя по логину и паролю.
* Добавление, редактирование и удаление данных из таблиц:
  + - «Адреса»;
    - «Паспорта»;
    - «Регистрации»;
    - «Работники»;
* Поиск данных: По названию улицы, виду паспортов и ФИО регистрации.
* Сортировка: пользователей по ФИО, паспортов по виду
* Фильтрация данных по различным критериям: названию города, типу паспорта, ФИО(Регистрации)

## 1.2.2 Требования к надежности и безопасности

Программа должна быть в достаточной степени надёжна от сбоев. На крайний случай предусмотрено сохранение данных БД в приложении «Microsoft SQL» или восстановление данных в случае завершения работы.

### 1.2.3 Требования к составу и параметрам технических средств

Таблица №1. «Состав технических средств и их характеристики»

|  |  |
| --- | --- |
| Процессор | Intel® Core™ i5-6400 CPU @ 2.70GHz |
| Оперативная Память | 2 x 4 ГБ, DDR4, DIMM, 2133 МГц |
| Разрешение экрана | 1920x1080 |
| Размер экрана | 20 дюйма |
| Устройства ввода | Мышь, клавиатура |
| Дисковое пространство | 1Гб |
| Подключаемые устройства | Принтер |
| Операционная система | Windows 10 11 |

### 1.2.4 Требования к информационной и программной совместимости

Для корректной работы программы необходимо:

ОС Windows 10/11 – операционная система, сделанная корпорацией Microsoft;

Visual Studio 2022 – лучшая интегрированная среда разработки для создания многофункциональных, привлекательных кроссплатформенных приложений для Windows;

Microsoft SQL Server Management Studio 18 – интегрированная среда для управления любой инфраструктурой SQL, от SQL Server до баз данных;

# 2. Разработка технического проекта

## Обоснование выбора CASE – средств

Средства автоматизации разработки программ (CASE-средства) — инструменты автоматизации процессов проектирования и разработки программного обеспечения для системного аналитика, разработчика ПО и программиста.

CASE-технологии (Computer-Aided Software/System Engineering) — инструментальные средства, используемые при проектировании систем. CASE-технологии охватывают весь спектр работ по созданию и сопровождению программного обеспечения (главным образом, анализ и разработку, составление проектной документации, кодирование и тестирование системы).

Задание на проектирование — технический документ (обычно выполняется в табличной форме), оговаривающий набор требований к продукту. Задание на проектирование в обязательном порядке подлежит утверждению Заказчиком и Исполнителем. Это может сыграть ключевую роль при урегулировании юридических и финансовых вопросов в дальнейшем. Де-юре, все работы и характеристики не указанные в задании на проектирование — это дополнительные требования и оплачиваются они отдельно.

Широкий набор функциональности: MS Visio предоставляет разнообразные типы диаграмм, включая диаграммы потоков данных, диаграммы классов, диаграммы вариантов использования и многое другое. Это позволяет разработчикам выбирать наиболее подходящий тип диаграммы для визуализации своих концепций и идей.

Интуитивно понятный пользовательский интерфейс: MS Visio имеет дружественный и интуитивно понятный пользовательский интерфейс, что делает процесс создания и редактирования диаграмм максимально простым и удобным. Разработчики смогут быстро освоить инструмент и создавать профессионально выглядящие диаграммы.

Расширяемость и настраиваемость: MS Visio позволяет расширять свои возможности с помощью сторонних плагинов и шаблонов. Разработчики могут использовать готовые шаблоны или создавать собственные, чтобы соответствовать особым требованиям и стандартам их организации.

Исходя из выбранного подхода к проектированию выбрано CASE – средство MS Visio, полностью удовлетворяющее запросам разработки программного средства.

## Проектирование модели данных

### 2.2.1 CASE – средство MS Visio

Microsoft Visio — программа для создания всевозможных видов схем. К их числу относятся блок-схемы, органиграммы, планы зданий и этажей, диаграммы DFD, схемы технологических процессов, модели бизнес-процессов, диаграммы плавательных дорожек, трехмерные карты и так далее. Основные возможности Visio:

* Диаграмма прецедентов - это графическое представление взаимодействия между системой и ее окружением, фокусирующееся на функциональности, которую система предоставляет своим пользователям (актерам). Она описывает различные прецеденты (сценарии использования) системы, используемые для взаимодействия с внешними сущностями. Диаграммы прецедентов часто используются в методологии Unified Modeling Language (UML) для моделирования требований к системе.
* Диаграмма действий – это графическое представление последовательности шагов или действий в конкретном процессе или сценарии. Она помогает визуально отобразить последовательность выполнения операций, решения проблем или других действий.
* Таблица операций обычно используется для документирования различных операций, процедур или действий, которые выполняются в рамках определенной системы или процесса.
* Таблицы описания документов могут быть использованы для организации и управления информацией о документах в организации или проекте. Эти таблицы могут включать информацию о названии документа, его типе, авторе, дате создания, версии, изменениях, статусе и другие сведения, которые помогают эффективно управлять документами. Такие таблицы облегчают отслеживание и доступ к документам в организации

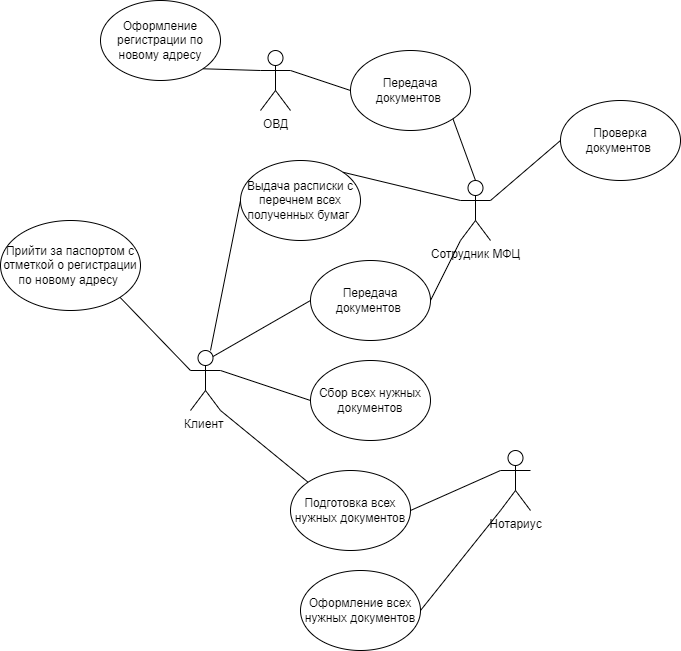


Рис. 2 «Диаграмма прецедентов»

Определение бизнес-процессов:

Таблица №2 «Таблица Бизнес-процессов»

|  |  |
| --- | --- |
| **Номер бизнес-процесса** | **Название бизнес-процесса** |
| 1 - Оформление всех нужных документов | Оформление всех нужных документов у нотариуса |
| 2 - Подготовка всех нужных документов | Подготовка всех нужных документов |
| 3 - Сбор всех нужных документов | Сбор всех нужных документов |
| 4 - Передача документов | Передача документов |
| 5 - Выдача расписки с перечнем всех полученных бумаг | Выдача расписки с перечнем всех полученных бумаг |
| 6 - Проверка документов | Проверка документов |
| 7 - Передача документов | Передача документов |
| 8 - Оформление регистрации по новому адресу | Оформление регистрации по новому адресу |
| 9 - Прийти за паспортом с отметкой о регистрации по новому адресу | Прийти за паспортом с отметкой о регистрации по новому адресу |

Словесный алгоритм бизнес-процесса «Оформление регистрации по новому адресу» осуществляется следующим образом:

1. Сотрудник МФЦ подготавливает документы и передает их в ОВД

2. Сотрудник ОВД добавляет заявку в очередь

3. Сотрудник ОВД Проверяет документы

4. При неправильном оформлении документов сотрудник ОВД отправляет документы обратно с уведомлением о неправильном оформлении документов

5. При правильном оформлении документов подтверждает регистрацию жителя

6. Сотрудник МФЦ получает готовые документы

Диаграмма действий:

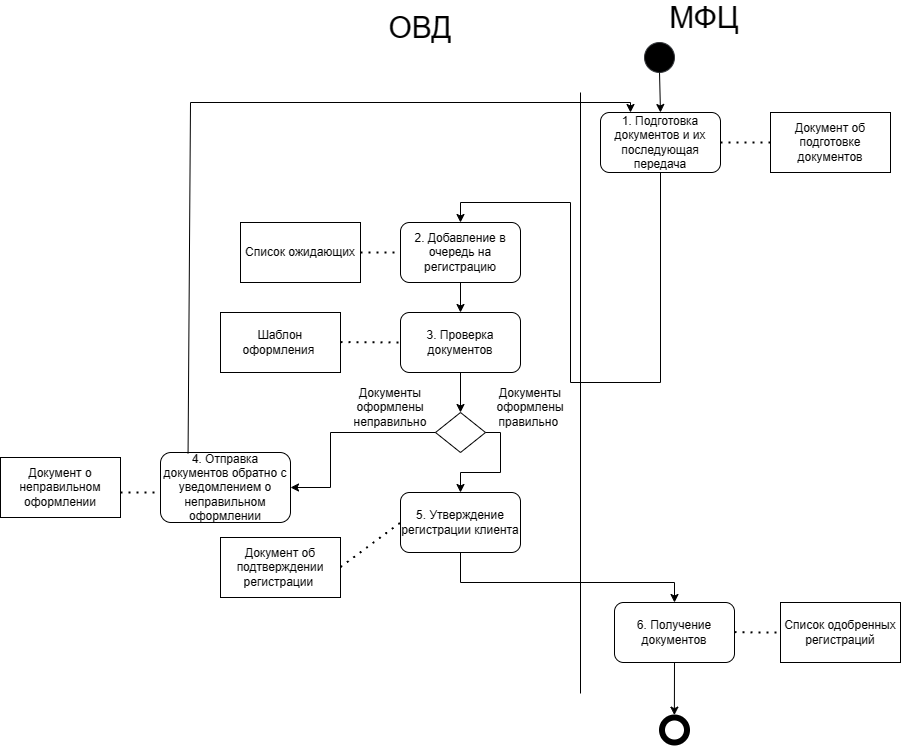


Рис. 3 «Бизнес-процесс «Оформление регистрации по новому адресу»»

Таблица №3 «Таблица описания операций

«Оформление регистрации по новому адресу\_8Ор\_Пна»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Диаграмма и номер операции на  диаграмме | Операция | Исполнитель | Как часто | Входящие  документы  (документы-  основания) | Исходящий  документ  (составляе­мый  документ) |
| 8 Ор\_Пна\_1 | Подготовка документов и их последующая передача | Сотдрудник МФЦ | По мере необходимости | Документ об подготовке документов | - |
| 8 Ор\_Пна\_2 | Добавление в очередь на регистрацию | Сотдрудник ОВД | По мере необходимости | Список ожидающих | - |
| 8 Ор\_Пна\_3 | Проверка документов | Сотдрудник ОВД | По мере необходимости | Шаблон оформления | - |
| 8 Ор\_Пна\_4 | Отправка документов обратно с уведомлением о неправильном оформлении | Сотдрудник ОВД | По мере необходимости | Документ о неправильном оформлении | - |
| 8 Ор\_Пна\_5 | Утверждение регистрации клиента | Сотдрудник ОВД | По мере необходимости | Документ об подтверждении регистрации | - |
| 8 Ор\_Пна\_6 | Получение документов | Сотдрудник МФЦ | По мере необходимости | Список одобренных регистраций | - |

# 3. Реализация

## 3.1 Обоснование выбора средств разработки

Выбран Visual Studio 2022

Разработчики зачастую сталкиваются с проблемами обработки информации, такими как необходимость быстрого и управляемого обработки данных, повышения производительности и мобильности разработчиков, а также с требованиями сокращения затрат на информационные технологии и расширения инфраструктуры в соответствии с растущими требованиями. В этом контексте Microsoft SQL Server 2022 является решением для управления и анализа данных. MS SQL Server 2018 предлагает интегрированное решение для управления и анализа данных, которое помогает:

- Строить, развертывать и управлять высоконадежными промышленными приложениями.

- Повышать производительность информационных технологий путем упрощения процессов разработки, развертывания и управления приложениями.

- Обеспечивать интеграцию данных между различными платформами, приложениями и устройствами.

- Оптимизировать стоимость без ущерба для качества выполнения, доступности, масштабируемости и безопасности.

Исходя из перечисленных выше преимуществ MS SQL Server 2018, этот продукт был выбран в качестве сервера базы данных для обеспечения повышенной эффективности и надежности работы системы.

## 3.2 Руководство программиста

Разработана схема данных в среде в Visual Studio.

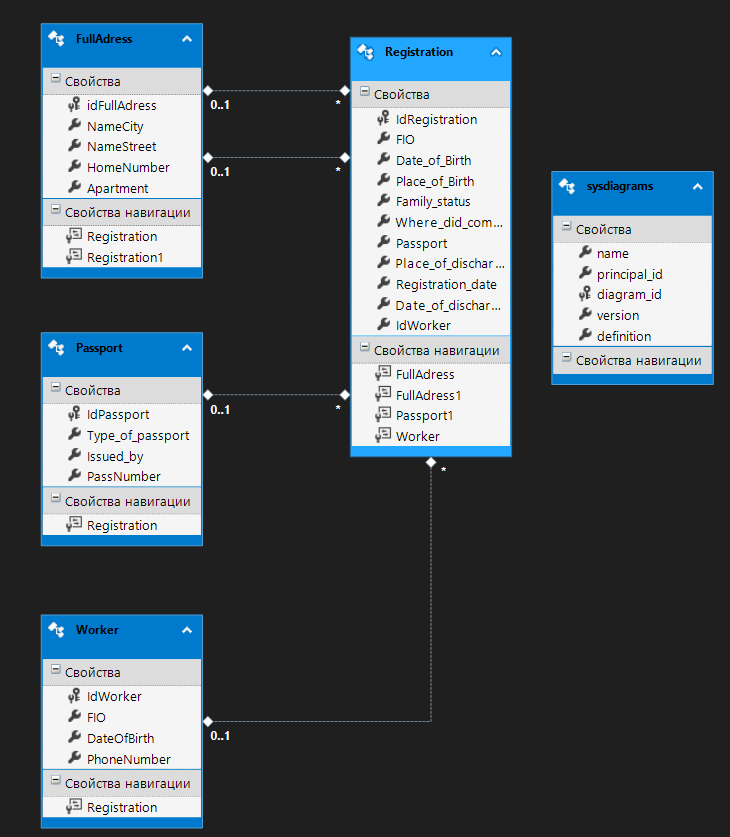


Рис. 3 «Модель данных»

Структура приложения в обозревателе решений:

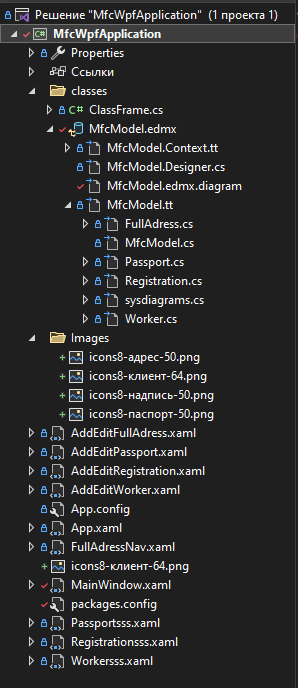


Рис. 4 «Структура приложения»

Класс Frame, служащий для вывода данных с базы данных:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Controls;

namespace SalesOfMaterials.Classes

{

public class ClassFrame

{

public static Frame frmObj;

}

}

Класс MFCDBEntities, служащий для связи с базой данных:

using System;

using System.Data.Entity;

using System.Data.Entity.Infrastructure;

public partial class MFCDBEntities : DbContext

{

private static MFCDBEntities \_context;

public MFCDBEntities()

: base("name=MFCDBEntities")

{

}

public static MFCDBEntities GetContext()

{

if (\_context == null)

\_context = new MFCDBEntities();

return \_context;

}

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

throw new UnintentionalCodeFirstException();

}

public virtual DbSet<FullAdress> FullAdress { get; set; }

public virtual DbSet<Passport> Passport { get; set; }

public virtual DbSet<Registration> Registration { get; set; }

public virtual DbSet<sysdiagrams> sysdiagrams { get; set; }

public virtual DbSet<Worker> Worker { get; set; }

}

**Входные и выходные данные**

Таблица №2 «Словарь данных»

|  |  |  |  |
| --- | --- | --- | --- |
| **ключ** | **поле** | **обязательное** | **примечание** |
| **Адрес** | | | |
| Первичный | IdExpenseIvoices | Да | Идентификатор |
|  | NameCity | Да | Название города |
|  | NameStreet | Да | Название улицы |
|  | HomeNumber | Да | Номер дома |
|  | Apartment | Да | Номер квартиры |
| **Паспорт** | | | |
| Таблица №2 «Словарь данных» продолжение таблицы | | | |
| Первичный | IdPassport | Да | Идентификатор |
|  | Type\_of\_passport | Да | Тип Паспорта |
|  | Issued\_by | Да | Кем выдан |
|  | PassNumber | Да | Номер Паспорта |
| **Работники** | | | |
| Первичный | IdWorker | Да | Идентификатор |
|  | FIO | Да | ФИО |
|  | DateOfBirth | Да | Дата рождения |
|  | PhoneNumber | Да | Номер телефона |
| **Тип пользователей** | | | |
| Первичный | IdRegistration | Да | Идентификатор |
|  | FIO | Да | ФИО |
|  | Date\_of\_Birth | Да | Дата рождения |
| Внешний | Place\_of\_Birth | Да | Внешний ключ к таблице адрес |
|  | Family\_status | Да | Семейное положение |
| Внешний | Where\_did\_come\_from | Да | Внешний ключ к таблице адрес |
| Внешний | Passport | Да | Внешний ключ к таблице паспорт |
| Внешний | Place\_of\_discharge | Да | Внешний ключ к таблице адрес |
|  | Registration\_date | Да | Дата регистрации |
|  | Date\_of\_discharge | Да | Дата когда выбыл |
| Внешний | IdWorker | Да | Внешний ключ к таблице работник |

## 3.3 Руководство пользователя

При открытии приложения выпадает главное окно с 4 кнопками для переходов на страницы: Адреса, Паспорта, Регистрации, Работники

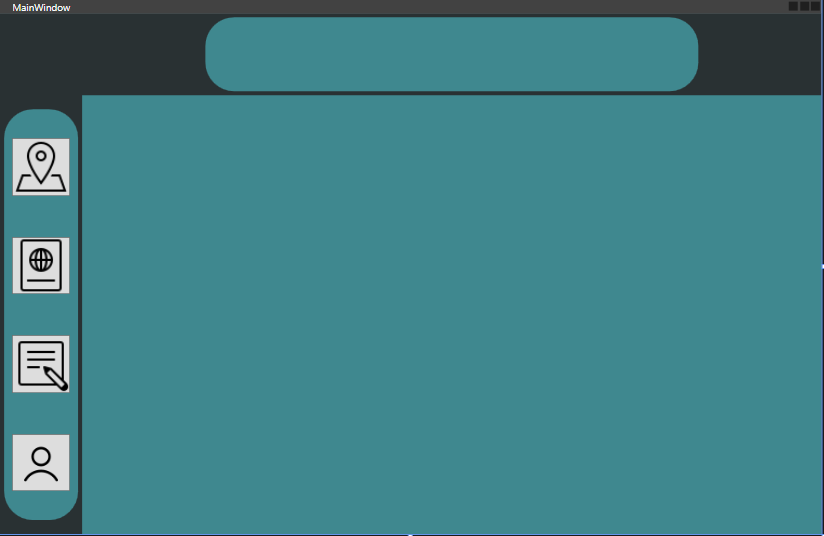


Рис. 4 «Главное меню»

Страница Адресов на этой странице вы можете добавлять удалять и редактировать адреса

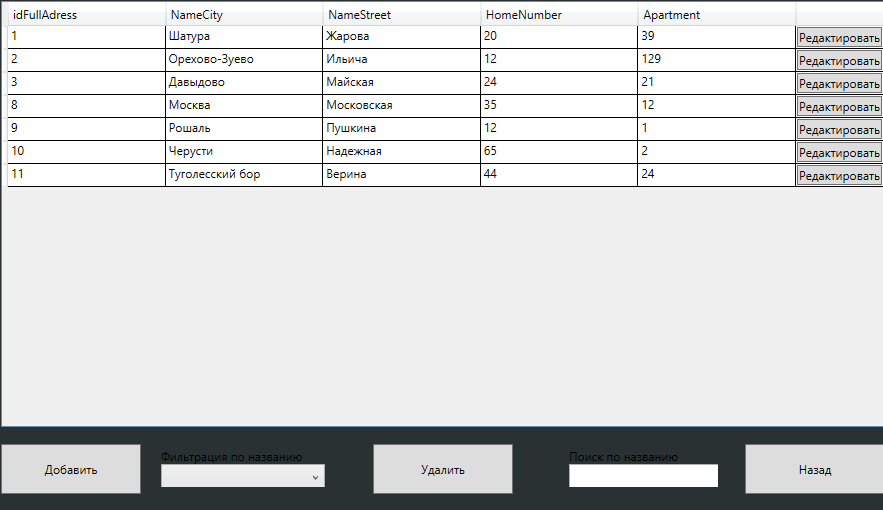


Рис. 5 «Страница адреса»

Страница Паспортов на этой странице можно добавлять удалять и редактировать Паспортные данные, а также фильтровать по типу паспорта и осуществлять поиск по типу паспорта

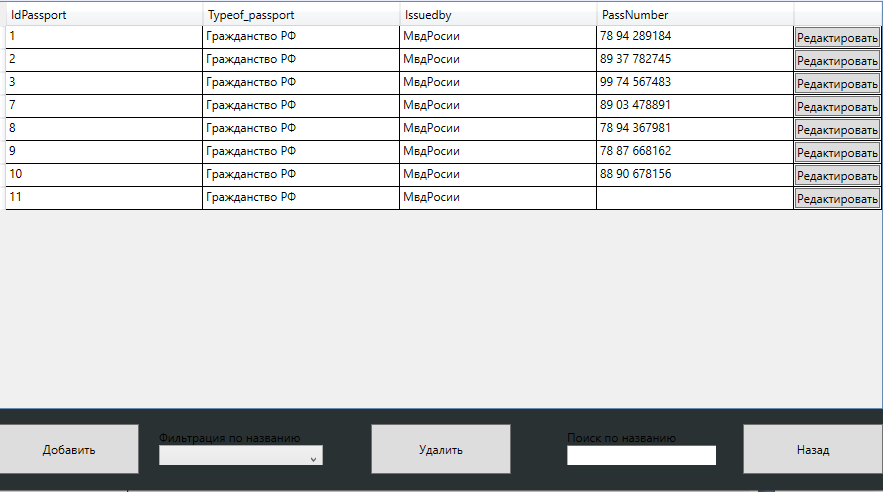


Рис. 6 «Страница паспорта»

Страница Регистрации на этой странице можно добавлять удалять и редактировать Регистрационные данные а также фильтровать по ФИО и осуществлять поиск по ФИО

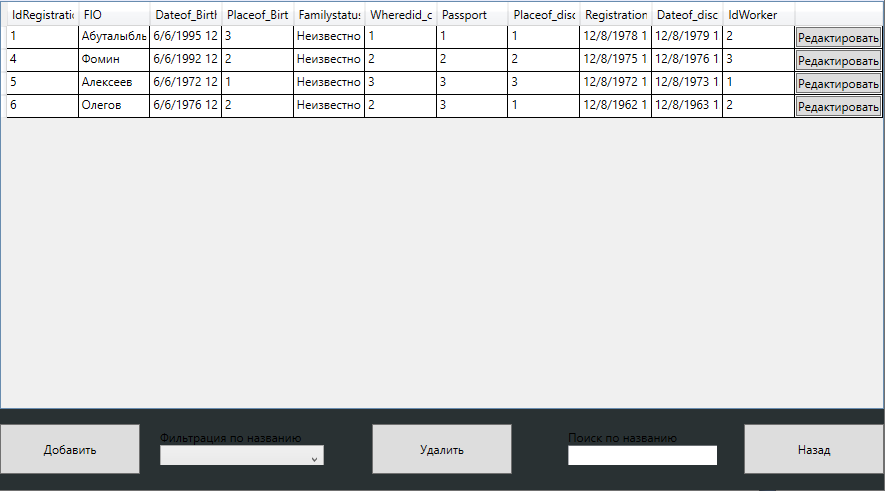


Рис. 7 «Страница Регистрации»

Страница Работников на этой странице можно добавлять удалять и редактировать данные о работниках а также фильтровать по ФИО и осуществлять поиск по ФИО

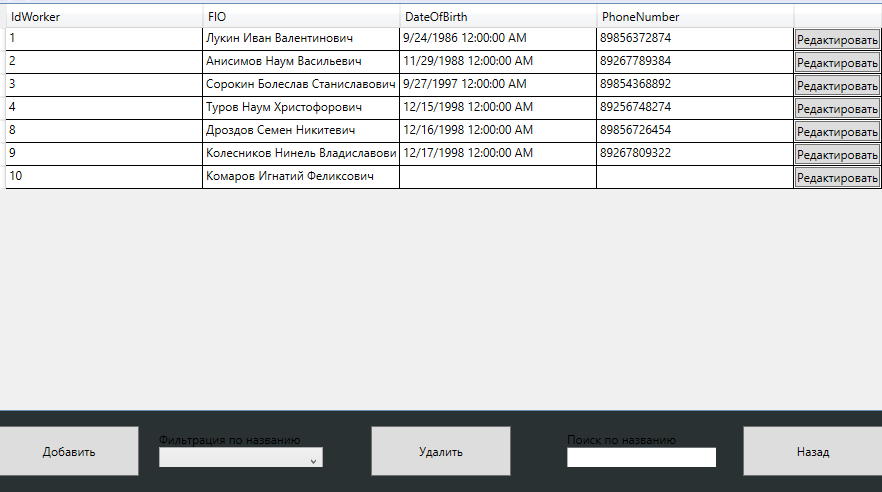


Рис. 8 «Страница Работников»

Страница добавления на примере добавления адреса на данной странице мы можем добавлять адрес по: названию города, названию улицы, номеру дома и номеру квартиры

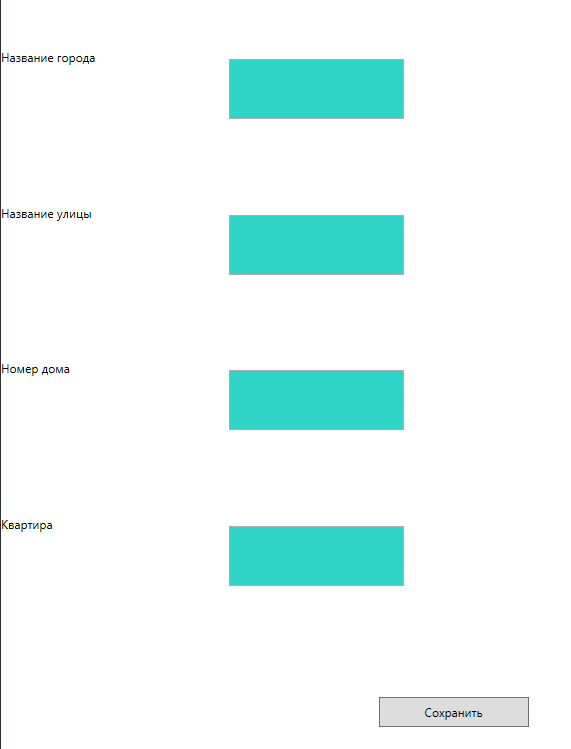


Рис. 9 «Страница Добавления»

Страница редактирования на примере редактирования адреса на данной странице мы можем редактировать адрес по: названию города, названию улицы, номеру дома и номеру квартиры

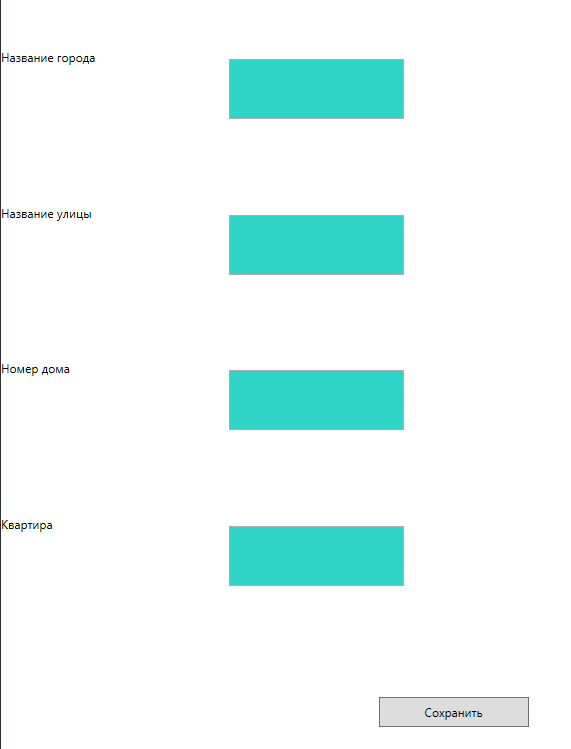


Рис. 10 «Страница Редактирования»

Функция Фильтрации по названию города на странице Адреса:

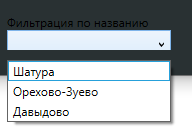


Рис. 11 «Функция фильтрации»

Функция Поиска по названию города на примере страницы Адресов:

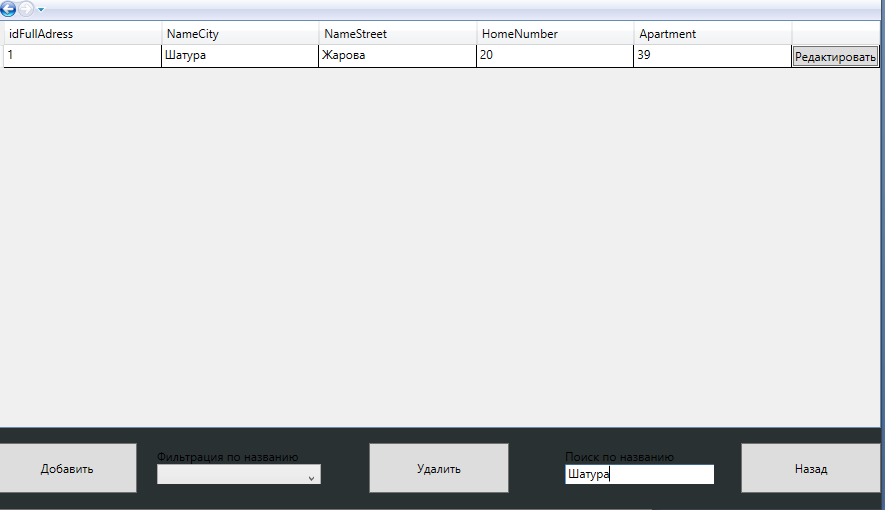


Рис. 12 «Функция поиска по названию»

Функция удаления на примере удаления Работника наводимся на запись и жмем удалить далее подтверждаем:

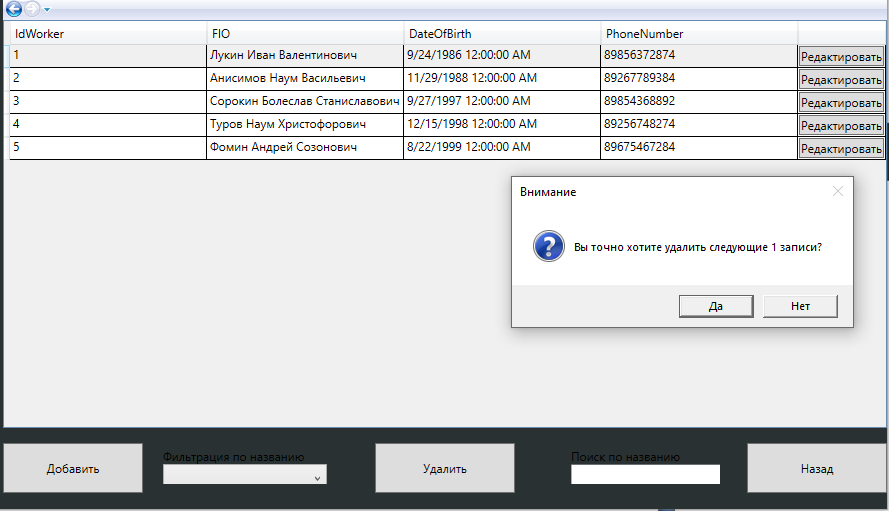


Рис. 13 «Функция удаления до удаления»

Результат ,после удаления у нас осталось 4 записи:

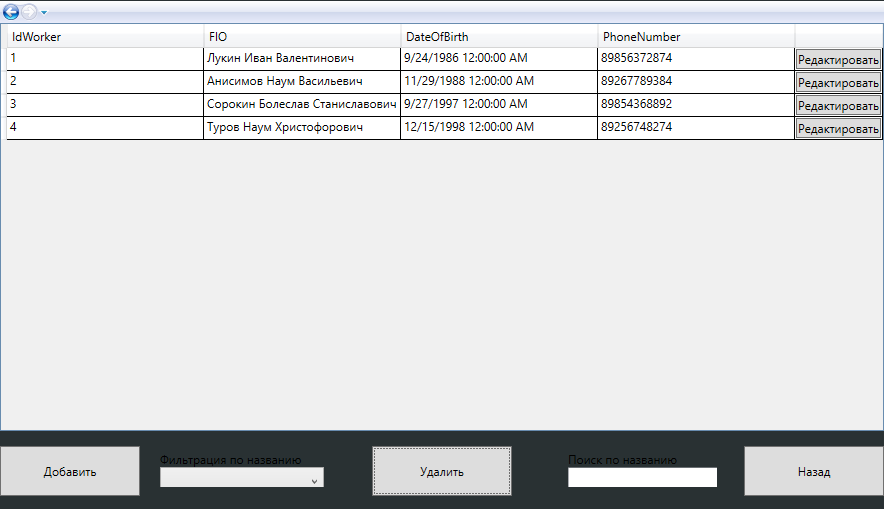


Рис. 14 «Функция удаления после удаления»

Диалоговые окна для корректной работы пользователя с приложением:

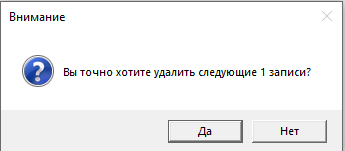


Рис. 15 «Диалоговое окно удаления»

# 4. Тестирование и откладка

Жизненный цикл тестирования программного обеспечения — это процесс выполнения различных действий в ходе проведения тестирования. Как пример, сюда относится составление тест-плана, анализ требований, поиск дефектов, поддержка после релиза и так далее.

Тестирование ПО — 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 (ISO/IEC TR 19759:2005).

Цель тестирования — проверка соответствия ПО предъявляемым требованиям, обеспечение уверенности в качестве ПО, поиск очевидных ошибок в программном обеспечении, которые должны быть выявлены до того, как их обнаружат пользователи программы.

Тестовый сценарий (test case) — это артефакт, описывающий совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или её части.

Методы тестирования:

* Тестирование белого ящика — метод тестирования ПО, который предполагает, что внутренняя структура/устройство/реализация системы известны тестировщику.
* Тестирование серого ящика — метод тестирования ПО, который предполагает комбинацию White Box и Black Box подходов. То есть, внутреннее устройство программы нам известно лишь частично.
* Тестирование чёрного ящика — также известное как тестирование, основанное на спецификации или тестирование поведения — техника тестирования, основанная на работе исключительно с внешними интерфейсами тестируемой системы.

Тестовые сценарии, выполненные по методу белого ящика:

Тестовый сценарий № 1:

|  |  |
| --- | --- |
| **Тестовый пример #** | 1 |
| **Приоритет тестирования** | Низкий |
| **Заголовок/название теста** | Перейдите на страницу «Адреса» и нажмите на пункт «Фильтр по названию улицы» и выберете любой параметр |
| **Краткое изложение теста** | Должно поменяться количество записей при использовании фильтрации |
| **Этапы теста** | 1. Запустить программу через Visual Studio  2. Попасть в главное меню. Оттуда перейти на страницу «Адреса»  3. Затем нажать в контекстном меню у пункта «Фильтр по названию улицы» любой параметр |
| **Тестовые данные** | После использования «Фильтра по названию улицы» с любым параметром все записи должны отфильтроваться по выбранному параметру |
| **Ожидаемый результат** | Если будет использоваться фильтрация, то количество записей должно изменяться |
| **Фактический результат** | Количество записей не изменилось |
| **Статус** | Незачет |
| **Предварительное условие** | Исправьте код, чтобы фильтр по названию улицы была выполнена корректно |
| **Постусловие** | Фильтрация |
| **Примечания/комментарии** | - |

Тестовый сценарий № 2:

|  |  |
| --- | --- |
| **Тестовый пример #** | 2 |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Перейдите на страницу «Адреса» и введите текст в «Поиск» |
| **Краткое изложение теста** | Должен выводиться список Адресов c поиском по названию улицы |
| **Этапы теста** | 1. Запустить программу через Visual Studio.  2. Попасть в главное меню. Оттуда перейти на страницу «Адреса».  3. Затем нажать ввести в поиск любой текст |
| **Тестовые данные** | После ввода текста, находящегося в одном из полей строки, данные выводятся правильно |
| **Ожидаемый результат** | Если будет использоваться поиск, то он будет выводить данные по всем полям |
| **Фактический результат** | Список Адреса выводится корректно |
| **Статус** | Зачет |
| **Предварительное условие** | - |
| **Постусловие** | Вывод данных с поиском по всем полям |
| **Примечания/комментарии** | Поиск осуществляется через выражение LINQ |

Участок кода с навигацией на страницу Адрес работающий некорректно:

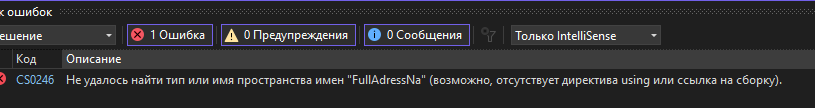


Рис. 16 «Неправильный тип или имя пространства имен»

Участок кода с навигацией на страницу Адрес работающий корректно:

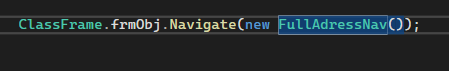


Рис. 17 «Навигация на страницу FullAdressNav»

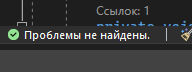


Рис. 18 «Исправленная функция»

Участок кода с присвоением переменной данных из БД работающий некорректно:



Рис. 19 «Неправильное название функции Binding»

Участок кода с присвоением переменной данных из БД работающий корректно:

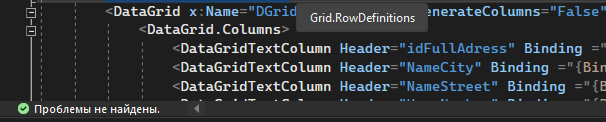


Рис. 20 «Исправленное название функции Binding»

# 5. Методы и средства защиты БД

Методы защиты баз данных в различных СУБД несколько отличаются друг от друга. Анализ современных СУБД показывает, что они условно делятся на две группы: основные и дополнительные.

К основным средствам защиты относится:

* защита паролем;
* шифрование данных и программ;
* разграничение прав доступа к объектам базы данных;
* защита полей и записей таблиц БД.

Защита паролем представляет собой простой и эффективный способ зашиты БД от несанкционированного доступа. Пароли устанавливаются пользователями или администраторами БД. Учет и хранение паролей выполняется самой СУБД. Обычно, пароли хранятся в определенных системных файлах СУБД в зашифрованном виде. После ввода пароля пользователю СУБД предоставляются все возможности по работе с БД.

Разрешение на доступ к конкретным объектам базы данных сохраняется в файле рабочей группы. Файл рабочей группы содержит данные о пользователях группы и считывается во время запуска. Файл содержит следующую информацию: имена учетных записей пользователей, пароли пользователей, имена групп, в которые входят пользователи. По отношению к таблицам могут предусматриваться следующие права доступа:

* просмотр (чтение) данных;
* изменение (редактирование) данных;
* добавление новых записей;
* добавление и удаление данных;
* изменение структуры таблицы.

К данным, имеющимся в таблице, могут применяться меры защиты по отношению к отдельным полям и отдельным записям. Защита данных в полях таблиц предусматривает следующие уровни прав доступа:

* полный запрет доступ;
* только чтение;
* разрешение всех операций (просмотр. ввод новых значений, удаление и изменение).

К дополнительным средствам защиты БД можно отнести такие, которые нельзя прямо отнести к средствам защиты, но которые непосредственно влияют на безопасность данных. Их составляют следующие средства:

* встроенные средства контроля значений данных в соответствии с типами;
* повышения достоверности вводимых данных;
* обеспечения целостности связей таблиц;
* организации совместного использования объектов БД в сети.

Дополнительное внимание хотелось бы обратить на “ Штатный аудит и мониторинг ” - Это средство защиты часто используется коммерческими организациями и входит в состав систем управления базами данных (далее — СУБД). Механизм работы штатного аудита заключается в настройке и включении триггеров, а также создании специфических процедур, которые начинают срабатывать во время запроса доступа к чувствительной информации. При этом ведется журнал запросов и подключений к системе управления базами данных в виде таблицы, где указаны данные о том, в какое время, кем и какой запрос был сделан.

## Заключение

В рамках данной курсовой работы разработана программа для управления учетом регистраций жителей, которая успешно соответствует всем требованиям пользователя.

В первую очередь рассмотрена общая информация о внутренней структуре организации, а также её бизнес-процессы. Это позволило глубже понять специфику работы организации и корректно отразить её в разрабатываемой программе.

В ходе выполнения работы создана база данных, предназначенная для учета регистраций жителей. Эта база данных обеспечивает полную информацию о регистрации, паспортных данных, а также о адресах.

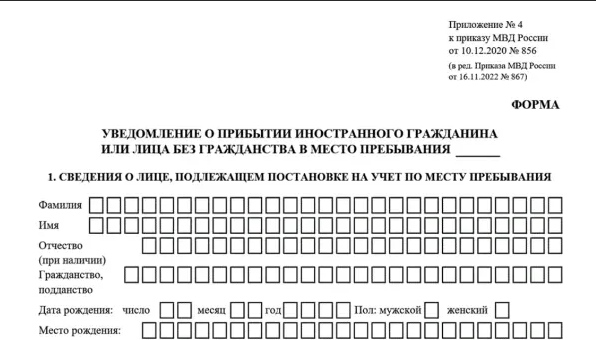
Следует отметить, что данная программа создана с использованием среды разработки Visual Studio . Выбор этой среды обусловлен её мощностью и многофункциональностью, своевременным обновлением функционала и способностью создавать решения для удобного использования информацией. Таким образом, программа полностью удовлетворяет поставленным требованиям в курсовой работе.

Руководство программиста позволит сопровождать и модифицировать программное решение другим специалистам.

В результате, разработанная программа представляет собой эффективное решение для управления предприятием, реализованы основные этапы жизненного цикла программного обеспечения

## Приложение

Приложение 1 «Форма Регистрации»



Приложение 2 «Свидетельство»



## Список литературы

1. Дейт, К. Дж. SQL и реляционная теория. Как грамотно писать код на SQL / К.Дж. Дейт. - М.: Символ-плюс, 2017. - 480 c.

2. Грабер, Мартин SQL для простых смертных / Мартин Грабер. - М.: ЛОРИ, 2014. - 378 c.

3. Гудсон, Джон Практическое руководство по доступу к данным (+ DVD-ROM) / Джон Гудсон , Роб Стюард. - М.: БХВ-Петербург, 2013. - 304 c.

4. Филатова, В. 1С для начинающих. Понятный самоучитель / В. Филатова. - М.: Питер, 2018. - 256 c.

5. Рудаков А.В., Федорова Г.Н. – Технология разработки программных продуктов. - М.: Академия, 2018. - 206 с.

6. Рудаков А.В., Федорова Г.Н. – Технология разработки программных продуктов. Практикум. - М.: Академия, 2014. - 189 с.

7. Бьюли, А. Изучаем SQL / А. Бьюли. - М.: Символ-плюс, 2014. - 108 c.

8. Дунаев, В. В. Базы данных. Язык SQL для студента / В.В. Дунаев. - М.: БХВ-Петербург, 2016. - 288 c.

9. Левинсон Джефф Тестирование ПО с помощью Visual Studio 2010; ЭКОМ Паблишерз - М., 2012. - 314 c.

10. Прайс, Джейсон Oracle Database 11g: SQL. Операторы SQL и программы PL/SQL / Джейсон Прайс. - М.: ЛОРИ, 2016. - 660 c.