Государственное бюджетное профессиональное образовательное учреждение

Самарской области

«Тольяттинский социально-экономический колледж»

**КУРСОВАЯ РАБОТА**

**РАЗРАБОТКА И ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «УЧЁТ ВЫДАЧИ И ВОЗВРАТА КНИГ»**

**ПМ.05 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ**

**МДК 05.02 Разработка кода информационных систем**

**09.02.07 Информационные системы и программирование**

|  |  |  |  |
| --- | --- | --- | --- |
| Студент |  | **/** | А.А. Храмков |
|  | *подпись* |  | *И.О. Фамилия* |
| \_\_.\_\_.2022 г. |  |  |  |
| Оценка выполнения и защиты курсовой работы | | |  |
|  | | |  |
| Руководитель |  | **/** | Е.В. Плюснина |
|  | *подпись* |  | *И.О. Фамилия* |
| \_\_.\_\_.2022 г. |  |  |  |

Тольятти, 2022

Государственное бюджетное профессиональное образовательное учреждение  
Самарской области  
«Тольяттинский социально-экономический колледж»

|  |  |  |
| --- | --- | --- |
|  |  | Утверждаю:  Заместитель директора по УР  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.С. Киронова  *« » 202 г.* |

**ЗАДАНИЕ**

на курсовую работу

**по ПМ.05 Проектирование и разработка информационных системмодуля, выполняемой в рамках МДК.05.02 Разработка кода информационных систем**

студента группы ИСП-31

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Храмкова Александра Александровича\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Фамилия Имя отчество студента*

Тема курсовой работы: «Разработка информационной системы **«**Учёт выдачи и возврата книг»

1. Содержание задания:

1.1 Разработать физическую структуру информационной системы.

1.2 Разработать запросы к информационной системе.

1.3 Разработать приложение.

1. Исходные данные:

Исходные данные для практической реализации автоматизированной информационной системы (АИС) берутся из различных информационных источников (Интернет-ресурсы, печатные издания, периодика и др.).

1. Содержание курсовой работы

Введение

1 Разработка базы данных информационной системы (название)

1.1 Разработка физической структуры БД

1.2 Разработка запросов ИС

2 Разработка интерфейса информационной системы (название)

Заключение

Список использованных источников

Приложения

Приложение А Листинг программы

Дата выдачи задания: «13» января 2022 г.

Дата сдачи работы на отделение: «\_\_»\_\_\_\_\_\_\_\_2022 г.

Руководитель курсового(ой) проекта(работы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Е.В. Плюснина

подпись расшифровка подписи

**КАЛЕНДАРНЫЙ ПЛАН**

выполнения курсовой работы

Студентом 3 курса группы ИСП-31

По теме Разработка и проектирование ИС «Учёт выдачи и возврата книг»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  этапа  работы | Содержание этапов работы | Плановый срок выполнения этапа | Планируемый объем выполнения  этапа, % | Отметка  о  выполнении  этапа |
| 1 | Выбор, обоснование темы и объекта исследования | Январь 2022 | 5% |  |
| 2 | Утверждение темы, согласование плана. Введение, библиография | Январь 2022 | 10% |  |
| 3 | Изучение и анализ информационных материалов по теме | Февраль 2022 | 15% |  |
| 4 | Обоснование актуальности выбранной темы применительно к профессиональной деятельности (введение) | Февраль 2022 | 20% |  |
| 5 | Изложение материала основной части по теме курсовой работы | Февраль 2022 | 20% |  |
| 6 | Подведение итогов проведенного анализа, формулировка выводов УИР применительно к профессиональной деятельности (заключение) | Март 2022 | 20% |  |
| 7 | Оформление работы и сдача на проверку | Март 2022 | 10% |  |
| 8 | Защита работы |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Студент |  | **/** | А.А. Храмков |
|  | *подпись* |  | *И.О. Фамилия* |
| \_\_.\_\_.2022 г. |  |  |  |
| Руководитель |  | **/** | Е.В. Плюснина |
|  | *подпись* |  | *И.О. Фамилия* |
| \_\_.\_\_.2022 г. |  |  |  |

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 5](#__RefHeading___Toc94035240)

1. [1 РАЗРАБОТКА БАЗЫ ДАННЫХ ИНФОРМАЦИОННОЙ СИСТЕМЫ РАСЧЕТ СТОИМОСТИ УСЛУГ, ПРЕДОСТАВЛЯЕМЫХ КЛИЕНТАМ 5](#__RefHeading___Toc94035241)

[1.1 Разработка физической структуры БД 5](#__RefHeading___Toc94035242)

[1.2 Разработка запросов ИС 5](#__RefHeading___Toc94035243)

[2 РАЗРАБОТКА ИНТЕРФЕЙСА ИНФОРМАЦИОННОЙ СИСТЕМЫ (НАЗВАНИЕ) 6](#__RefHeading___Toc94035244)

[ЗАКЛЮЧЕНИЕ 8](#__RefHeading___Toc94035246)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 9](#__RefHeading___Toc94035247)

[ПРИЛОЖЕНИЕ А.](#__RefHeading___Toc94035248) [ЛИСТИНГ ПРОГРАММЫ 10](#__RefHeading___Toc94035249)

# ВВЕДЕНИЕ

В современных реалиях многие организации постепенно переходят на автоматизированные информационные системы для обеспечения более быстрого и качественного выполнения задач. В последние годы, влияние компьютерных технологий значительно изменило представление о библиотеке и её функциях. Теперь электронные ресурсы и информационные технологии должны быть неотъемлемой составной библиотек.

В библиотеках хранится множество различных книг, журналов. Каждый день читатели приходят выбрать что-то для себя. Кто-то предпочитает читать в библиотеке, кто-то забрать на определенное время с собой. Учёт выдачи и получения книг без использования информационных систем будет довольно долгим и проблематичным, поскольку часто приходится проверять наличие и количество книжных экземпляров, которых сотни тысяч.

Библиотека не осталась в стороне. Для обеспечения быстрого доступа к большому количеству информации о книгах, журналах, буклетах и других бумажных носителях необходимо прибегнуть к помощи автоматизированной информационной системы.

Библиотеки позволяют практически любому желающему прийти и прочитать какую-либо книгу, либо забрать её на определенный срок, после чего принести обратно. Несмотря на постепенный переход к электронным книгам, библиотека остаётся востребованной по сей день.

Разумеется, никак не обойтись без базы данных. Существует множество СУБД для их редактирования, но целью является создание приложения, занимающего небольшой объем, простое в использовании и позволяющее редактировать базу данных без установленной СУБД, а также выполняющее учёт выдачу и получение книг. Именно этому и посвящена данная курсовая работа.

# При изучении предметной области, выяснилось, что информационная система «Учёт выдачи и возврата книг» на первый взгляд простая, но для ее реализации необходимо обратится к некоторым разделам, таким как создание базы данных и создание приложения для взаимодействия с этой базой данных.

# Объектом исследования курсовой работы является процесс учёта получения и выдачи книг в библиотеки.

# Предметом исследования процесса будет ИС «Учёт выдачи и возврата книг».

# Вышеизложенное в целом на теоретико-методологическом уровне определило проблему настоящего исследования: разработка программного модуля «Расчет стоимости» в программе MSVisualStudio 2022.

# Целью курсовой работы является разработка и проектирование информационной системы «Учёт выдачи и возврата книг».

# Для достижения поставленной цели были определены следующие задачи:

# анализировать предметную область;

# обосновать актуальность разработки информационной системы;

# разработать функциональную модель;

# описать средства разработки информационной системы;

* + - спроектировать информационную систему, построив различные виды диаграмм.

# Для разработки информационной системы будут применяться методы: анализ деятельности предприятия с выявлением его функции, а также функциональное моделирование системы.

# Практическая значимость заключается в использовании разработанной системы на различных предприятиях по «Учёт выдачи и возврата книг».

# Структура работы соответствует логике исследования и включает в себя введение, две главы, заключение, список использованной литературы, приложение.

# 1 РАЗРАБОТКА БАЗЫ ДАННЫХ ИНФОРМАЦИОННОЙ СИСТЕМЫ УЧЁТ ВЫДАЧИ И ВОЗВРАТА КНИГ

## 1.1 Разработка физической структуры БД

Чтобы создать новую базу данных откроем SQL Server Management Studio 18. В обозревателе объектов подключитесь к экземпляру компонента Компонент SQL Server Database Engine и разверните его. Щелкните правой кнопкой мыши узел Базы данных и выберите команду создать базу данных. В поле Новая база данных введите имя базы данных. Чтобы создать базу данных, приняв все значения по умолчанию, нажмите кнопку ОК.

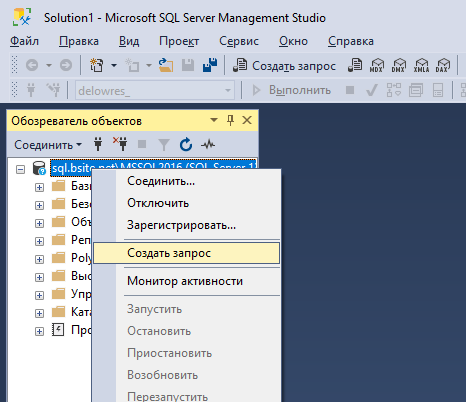


Рисунок 1.1 – Окно обозревателя объектов

В обозревателе объектов открываем контейнер «Базы данных», затем открываем нужную базу данных и щелкаем правой кнопкой мыши по пункту «Таблицы», и выбираем «Таблица».

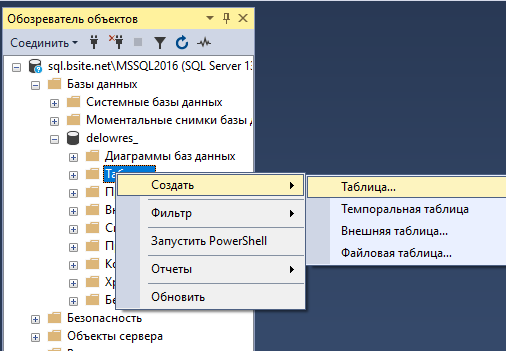
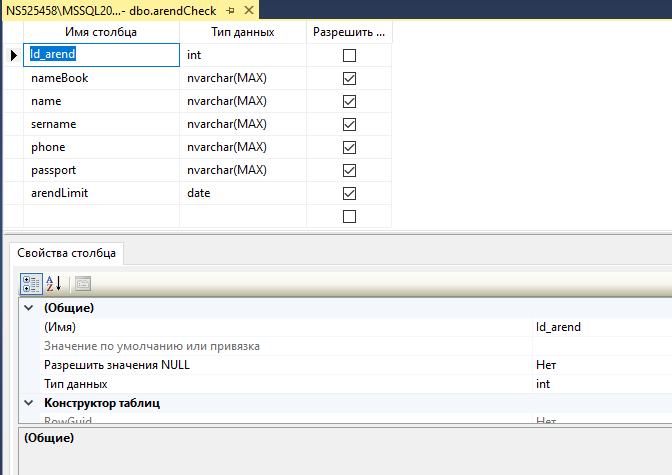
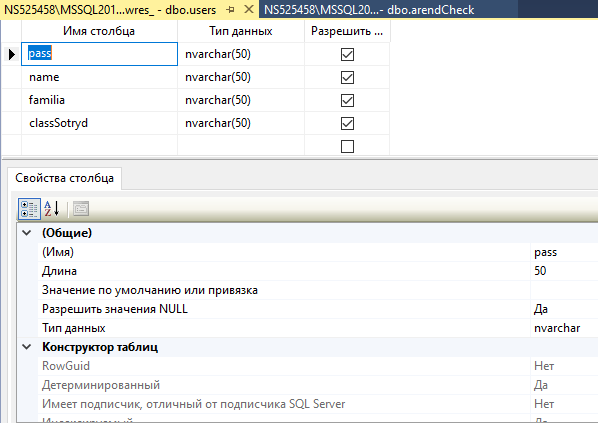


Рисунок 1.2 – Окно создания таблицы

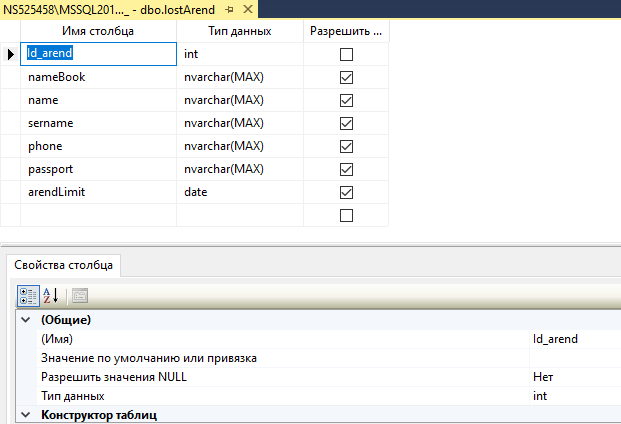
Введите имена столбцов, выберите типы данных и определите для каждого столбца, могут ли в нем присутствовать значения NULL, как показано на следующей иллюстрации:

**

*Рисунок 1.3 – Таблица «Книги в аренде»*

**

*Рисунок 1.4 - Таблица «Пользователи»*

**

*Рисунок 1.5 - Таблица «Просроченная выдача»*

Вы также можете задать другие свойства столбца, например, является ли этот столбец столбцом идентификаторов или вычисляемым столбцом. Для этого щелкните столбец на вкладке свойств столбцов. Чтобы указать, что столбец является столбцом первичного ключа, щелкните его правой кнопкой мыши и выберите «Задать первичный ключ». Чтобы создать связи по внешнему ключу, проверочные ограничения или индексы, щелкните правой кнопкой мыши панель конструктора таблиц и выберите в списке объект, как показано на следующей иллюстрации:

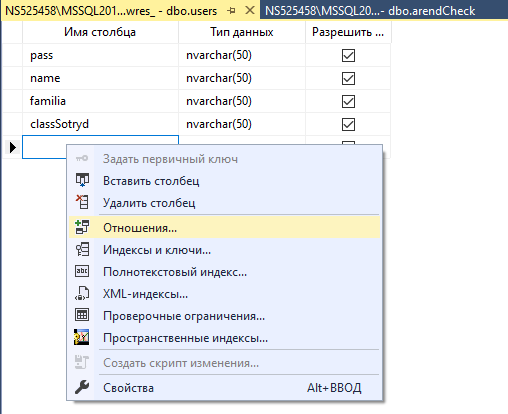


Рисунок 1.6 – Окно создания связей по внешнему ключу

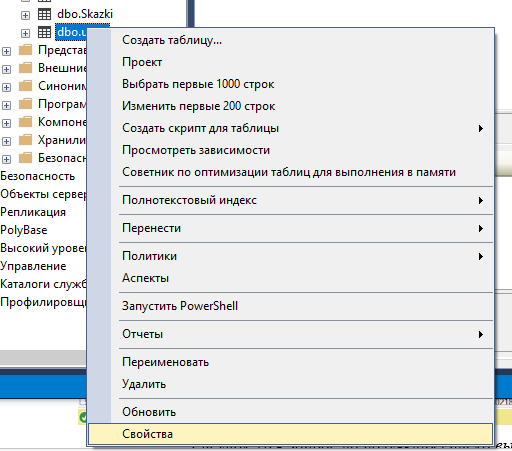
По умолчанию таблица содержится в схеме dbo. Чтобы указать другую схему для таблицы, щелкните правой кнопкой мыши панель конструктора таблиц и выберите «Свойства», как показано на следующей иллюстрации:

Рисунок 1.7 – Окно выбора свойств таблицы «Проверка аренды»

Выберите нужную схему из раскрывающегося списка «Схема».

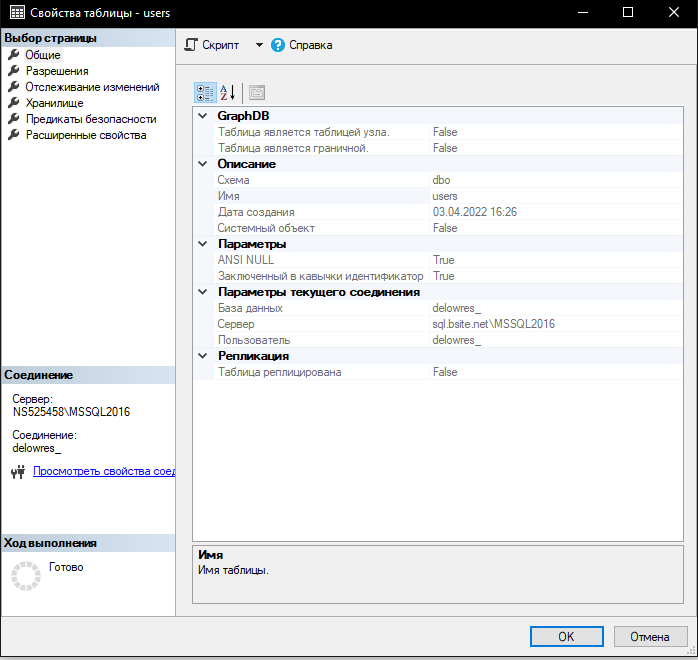
**

Рисунок 1.8 – Окно свойств таблицы «Пользователи»

В меню Файл выберите команду «Сохранить» и введите название таблицы – «имя\_таблицы».

Чтобы просмотреть новую таблицу, в обозревателе объектов разверните узел Таблицы, а затем нажмите клавишу F5, чтобы обновить список объектов. Новая таблица будет отображена в списке таблиц.

Создание остальных таблиц базы данных для информационной системы происходит аналогично примеру, описанному выше.

## 1.2 Разработка запросов ИС

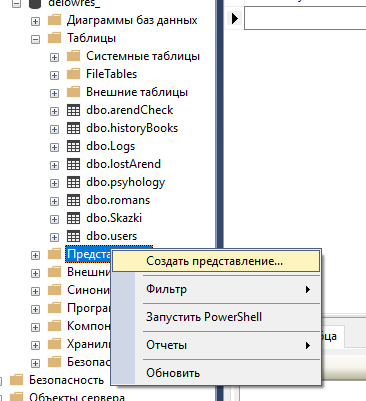
Чтобы создать представление, в обозревателе объектов открываем контейнер «Базы данных», затем открываем нужную базу данных и щелкаем правой кнопкой мыши по пункту «Представления», после «Создать представление…».

Рисунок 1.9 – Окно создания представления

В диалоговом окне «Добавление таблицы» выберите один или несколько элементов, которые необходимо включить в новое представление, на одной из следующих вкладок: «Таблицы», «Представления», «Функции» и «Синонимы». Щелкните «Добавить», а затем выберите Закрыть.

На Панели диаграмм выберите столбцы или другие элементы для включения в новое представление. На Панели критериев выберите дополнительные условия сортировки или фильтрации для столбцов.

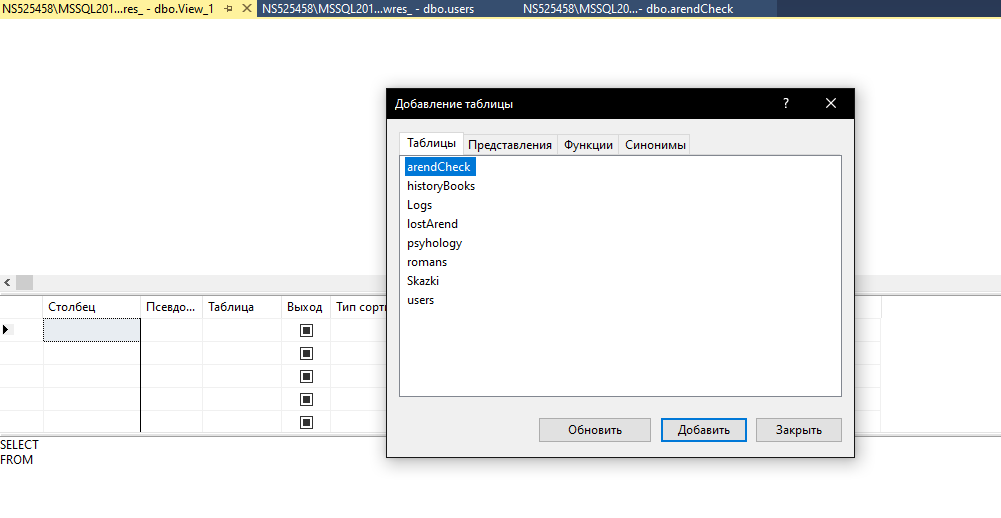


Рисунок 1.10 – Диалоговое окно добавление таблицы

В меню Файл выберите пункт «Сохранить» и введите название представления – «имя\_представления».

Затем создадим запрос для информационной системы путем выбора столбцов или других элементов представления. На панели критериев выберите дополнительные условия сортировки или фильтрации для столбцов.

Общая структура запроса выглядит следующим образом:

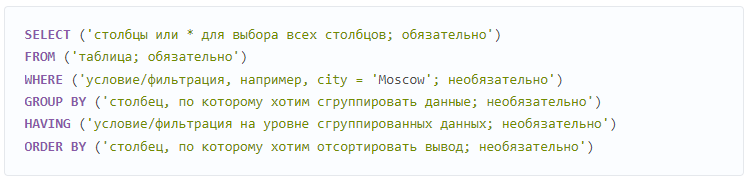


Рисунок 1.11 – Общая структура запроса

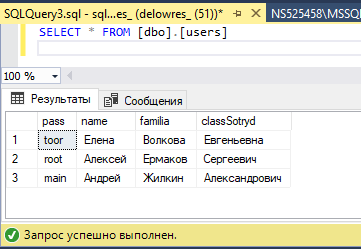
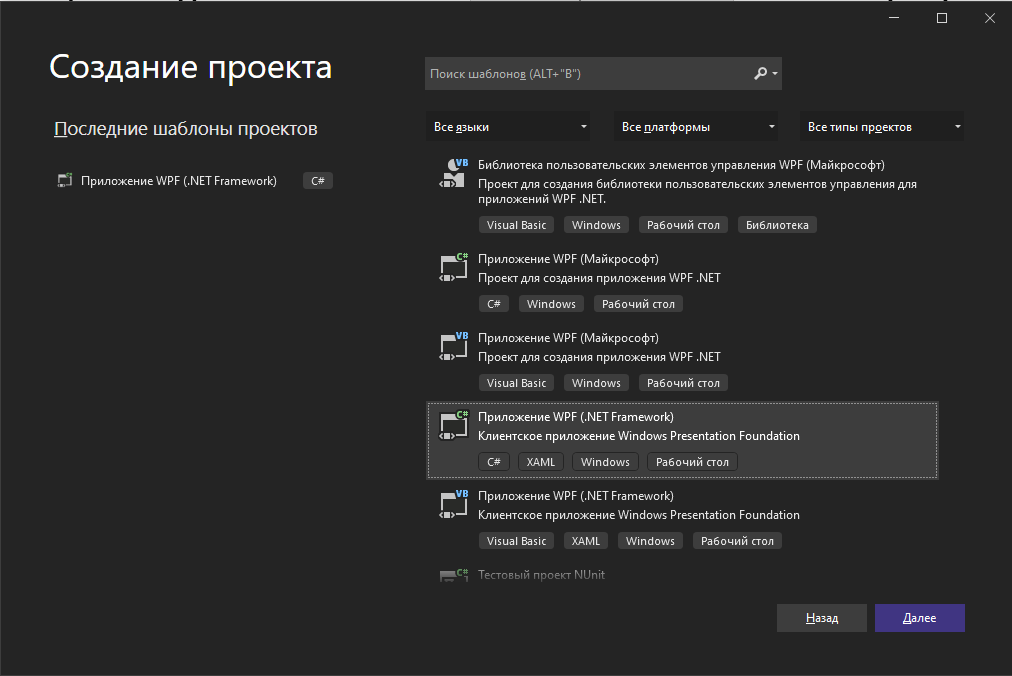


Рисунок 1.12 – Запрос Select для информационной системы

# Создание остальных запросов базы данных для информационной системы происходит аналогично примеру, описанному выше.

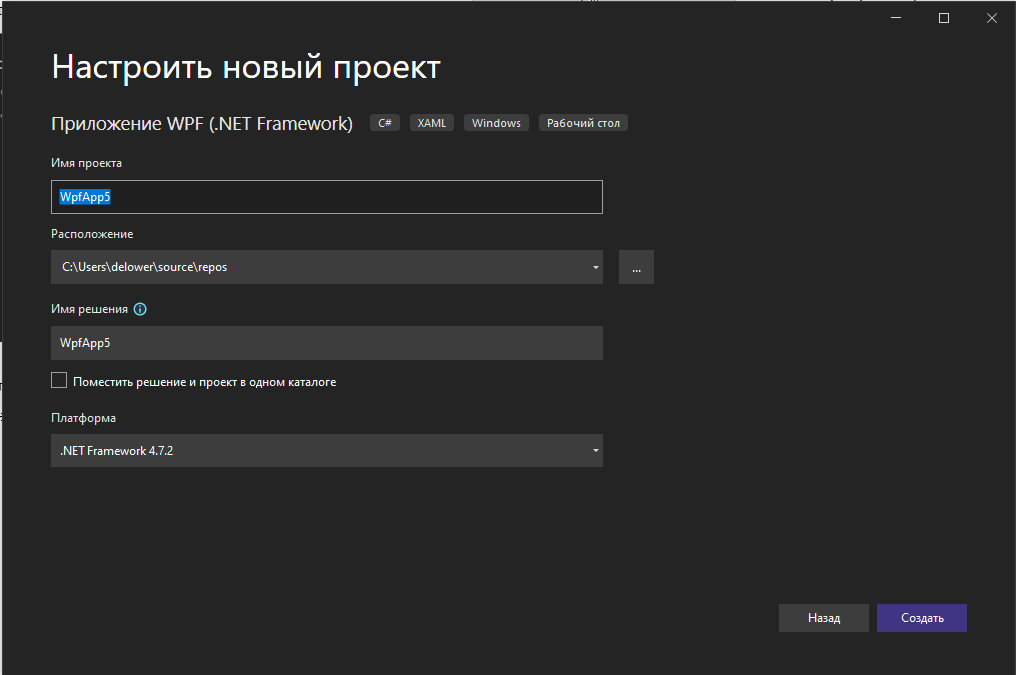
# 2 РАЗРАБОТКА ИНТЕРФЕЙСА ИНФОРМАЦИОННОЙ СИСТЕМЫ «УЧЁТ ВЫДАЧИ И ВОЗВРАТА КНИГ»

**2.1 Разработка интерфейса**

Первым шагом является создание инфраструктуры приложения. Создайте новый проект приложения WPF в Visual Basic или Visual C# и введите название. Откройте Visual Studio и выберите создать новый проект в меню начало работы. Откроется диалоговое окно «Создание нового проекта». В раскрывающемся списке язык выберите C#. Выберите шаблон приложения WPF (платформа .NET Framework) и нажмите кнопку далее.

2.1 Разработка руководства пользователя

Откроется диалоговое окно «Настройка нового проекта». Введите имя проекта и нажмите кнопку создать.

Рисунок 2.2 – Окно создания приложения WPF

Visual Studio создает проект и открывает конструктор для окна приложения по умолчанию с именем MainWindow.xaml.

Чтобы как-то взаимодействовать с пользователем, получать от пользователя ввод с клавиатуры или мыши и использовать введенные данные в программе, нам нужны элементы управления.

Создаем интерфейс окна авторизации, используя элементы управления WPF – Grid, Label, Border, Image для визуального восприятия информации; PasswordBox для ввода пароля; Button для закрытия окна и для входа в систему.

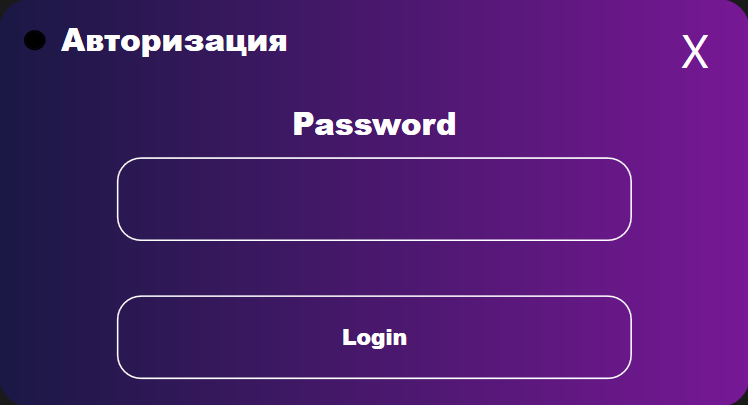


Рисунок 2.3 – Интерфейс окна «Авторизация»

Затем откроем App.xaml (C#). Этот XAML-файл определяет приложение WPF и все ресурсы приложения. Этот файл также используется для указания пользовательского интерфейса, в данном случае окна авторизации, которое автоматически отображается при запуске приложения.

Код XAML для окна авторизации выглядит следующим образом:



Рисунок 2.4 – XAML-код окна авторизации



Рисунок 2.5 – XAML-код Grid



Рисунок 2.6 – XAML-код Border с кнопками

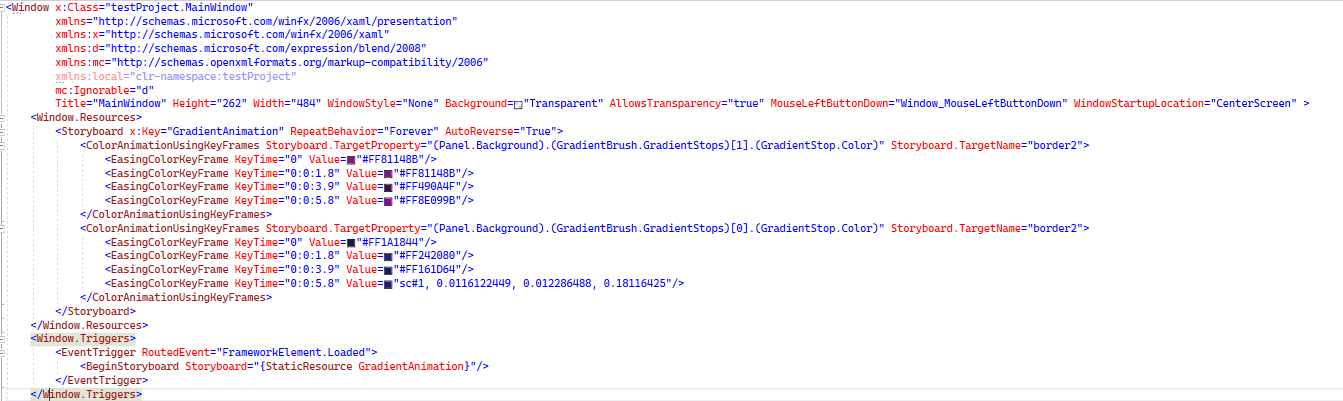


Рисунок 2.7 – XAML-код Window



Рисунок 2.8 – XAML-код Button

Интерфейс окна авторизации с проверкой пароля будет выглядеть следующим образом:

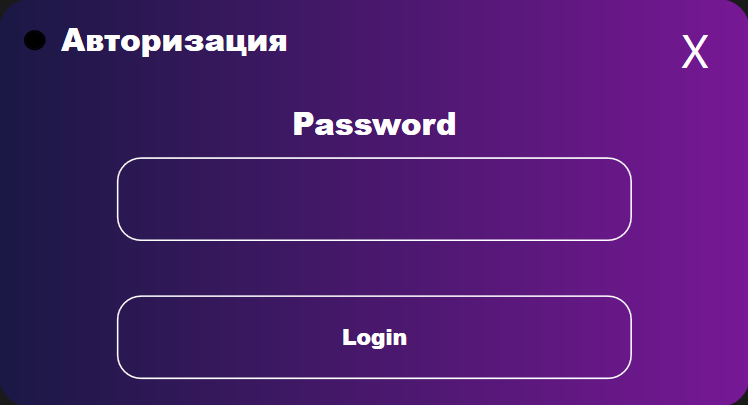
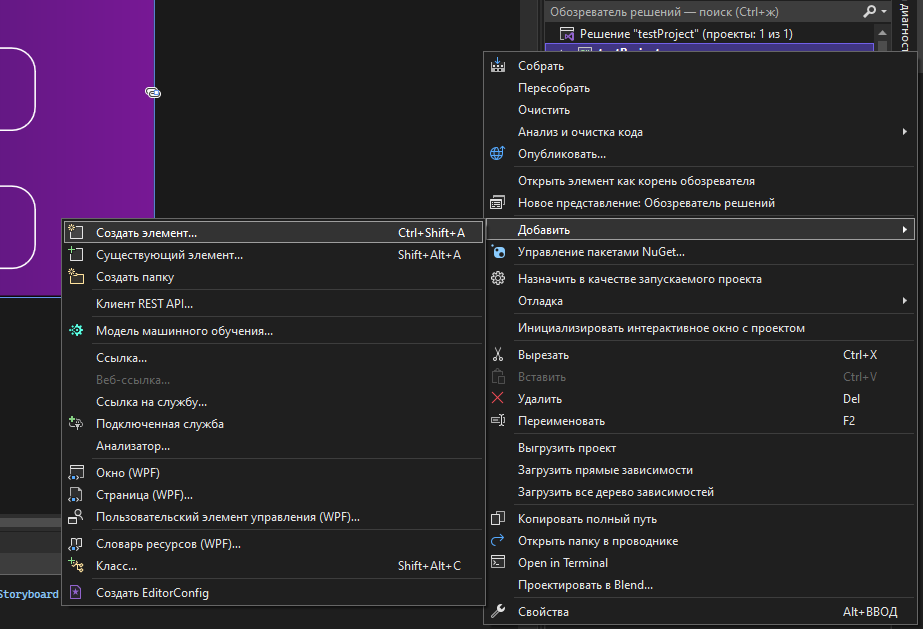


Рисунок 2.9 – Интерфейс окна «Авторизация» с проверкой ввода пароля

Следом перейдем к созданию кода на языке C# для окна авторизации. В обозревателе решений нажмем правой кнопкой мыши по проекту, выберем пункт «Добавить» - «Создать элемент».

 Рисунок 2.10 – Окно обозреватель решений

В открывшемся окне «Добавление нового элемента» выберем элемент «Класс Visual C#» и дадим ему название. Нажмем кнопку «Добавить» и перейдем к нему.

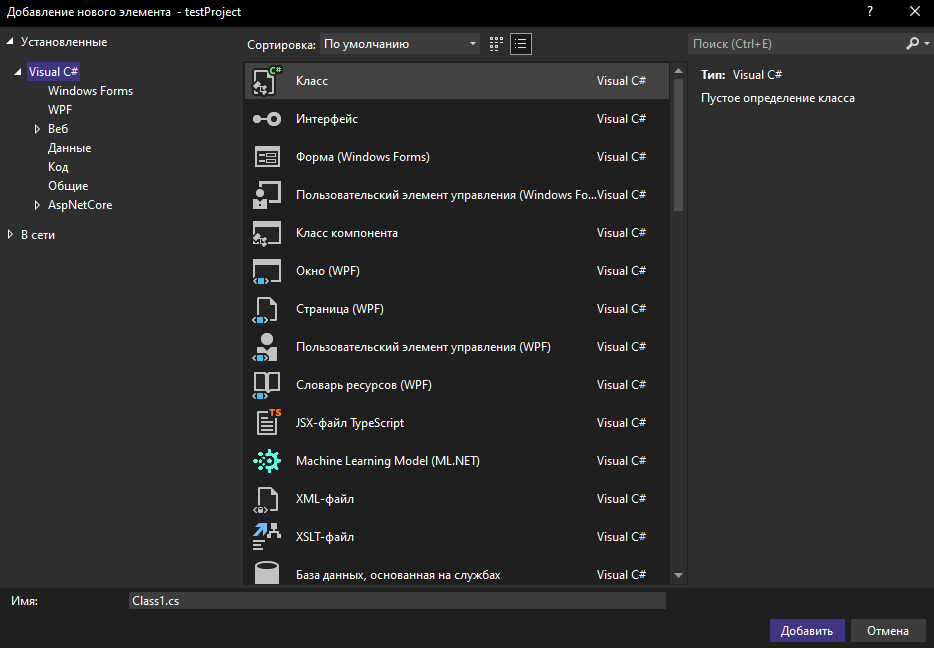


Рисунок 2.11 – Окно добавления нового класса

Создадим класс Connect, который будет брать данные о пользователе из sql-базы данных, используя метод Select.

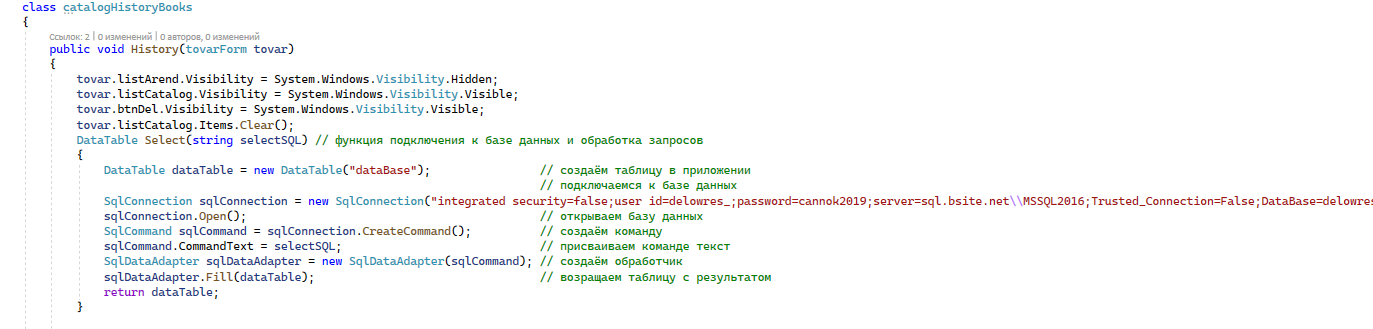


Рисунок 2.12– Код класса Connection

Затем сформируем класс Login, отвечающий за вход в систему. В этом классе будет обращение к классу Connection для получения информации из sql-базы данных о пользователе. Поставим условия входа – при удачном вводе пароля, пользователь попадает на главное окно. При ошибке ввода добавляется надпись “Неверные данные”

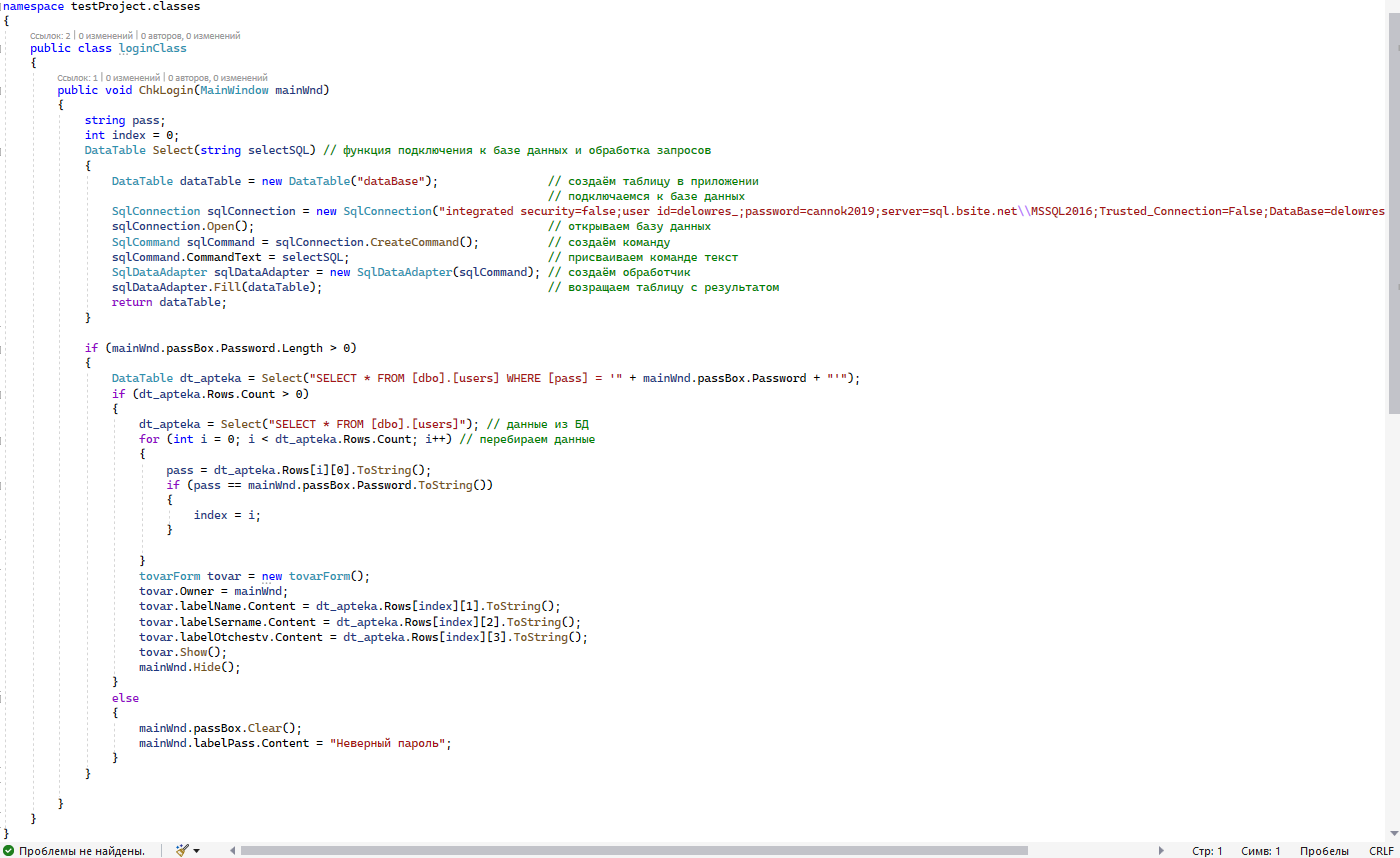
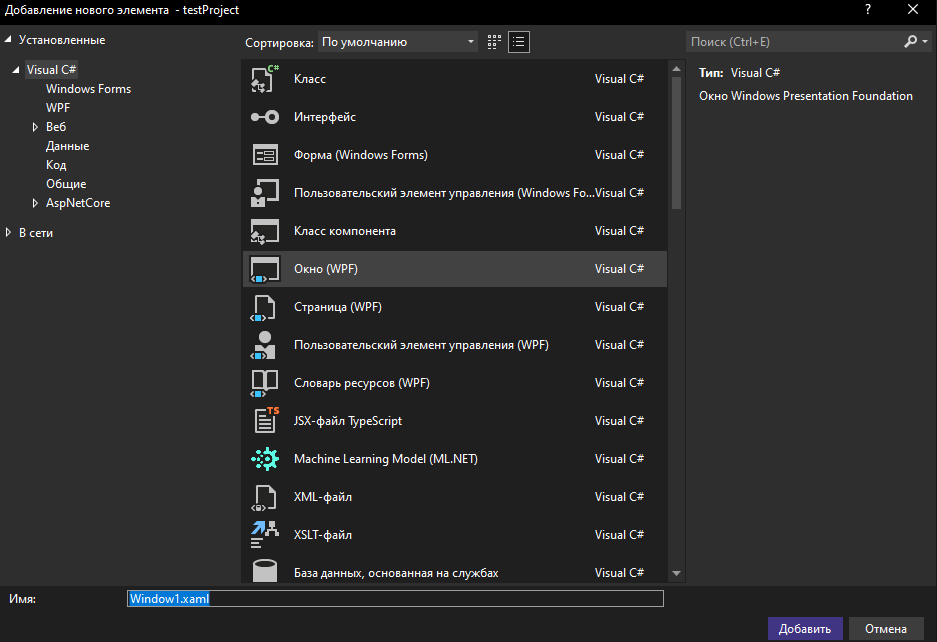


Рисунок 2.13 – Код класса Login с проверкой введенного пароля

## Следом добавим главное окно проекта. Чтобы осуществить открытие нового окна в WPF приложении, для начала нужно создать окно. Для этого правой кнопкой мыши (ПКМ) нажимаем на название проекта в «Обозреватель решений» — пункт «Добавить» — «Окно».

Рисунок 2.14 – Добавление главного окна проекта

После нужно указать название и нажать кнопку «Далее».

 Рисунок 2.15 – Добавление имени главного окна проекта

Интерфейс главного окна и программный код разрабатываются аналогично описанному выше примеру

## 2.5 Разработка руководства пользователя

Информационная система «BookCheck» представляет собой программный комплекс для сотрудников библиотеки с деятельностью и юридической информацией об организации ООО «BookCheck».

Данное руководство предназначено для пользователей программного продукта, предназначенного для операционной системы Windows 10.

Документ «Руководство Пользователя» предназначен для конечных пользователей продукта, администраторов системы и другого персонала.

Для установки приложения компьютер конечного пользователя должен удовлетворять требованиям.

Минимальная конфигурация сервера:

* + - тип процессора – Pentium IV и выше или совместимый с ним;
    - объем оперативного запоминающего устройства 4Гб и более;
    - жесткий диск 80 Гб;
    - модем, для выхода в Интернет;
    - монитор, клавиатура, мышь.

Требования, предъявляемые к конфигурации клиентских станций:

* + - процессор, с тактовой частотой не менее 1200 MHz,
    - 2 GB оперативной памяти;
    - монитор – SVGA;
    - клавиатура - 101/102 клавиши;
    - манипулятор типа «мышь».

Для реализации данной системы требуется установка следующего ПО: Microsoft Visual Studio 2021 и СУБД Microsoft SQL Server 2018 Express Edition.

Минимальная конфигурация:

* + - процессор x86 (Intel, AMD) с частотой как минимум 1,8 ГГц;
    - 2 ГБ ОЗУ; рекомендуется 4 ГБ ОЗУ;
    - доступное место на жестком диске объемом от 2 до 10 ГБ.

Для авторизации введите пароль в открывшемся окне при входе в приложение.

При введении верного пароля вы попадаете в основное окно программы.

Диалоговое окно «Неправильный пароль» обозначит ошибку ввода логина или пароля. Попробуйте ввести пароль заново для входа в программу.

На главном окне находится основной функционал программы.

В боковом меню находятся разделы:

* Каталог книг;
* Книги в аренде;

В верхнем меню находится:

* ФИО пользователя, вошедшего в систему;
* Список книг с просроченным возвратом и их количество;
* Кнопка выхода из учетной записи;
* Кнопка выхода из программы

В нижнем меню находится:

* Кнопка “Удалить”;
* Кнопка “Добавить”;

## Окна приложения

|  |  |
| --- | --- |
| **Окно** | **Функционал** |
| Главная | Содержит главную страницу приложения. |
| Каталог | Содержит список книг, разделенный на определенные жанры. Выдаёт информацию о наличии книг и информацию о них. |
| Книги в аренде | Список книг, выданных читателям на определенный срок. Содержит информацию о выданной книге, дате возврата и контактных данных читателя. |
| Просроченный возврат | Список книг, которые не были вовремя возвращены. |
| Отчёт | Выводит отчет о выданных и возвращенных книгах |

# ЗАКЛЮЧЕНИЕ

В первой части курсовой работы был произведен [анализ предметной области](#_Toc473964330)системы автоматизации информационной системы «Учёт выдачи и возврата книг». Было разработано техническое задание, распространяемое на разработку информационной системы, а также раскрывающее назначение, требования к надёжности, функциям, эргономике, видам обеспечения и к системе в общем. Также, был рассмотрен бизнес-процесс отдел «Учёт выдачи и возврата книг», в дальнейшем построена функциональная модель системы в программной среде ComputerAssociatesBPWin. Произведена [декомпозиция задачи, выполнен структурно-функциональный анализ объекта управления.](#_Toc473964331)

Во второй части курсовой работы на основе рассмотренных бизнес-процессов отдела «Расчет стоимости услуг, предоставляемых клиентам» были разработаны и подробно описаны логическая и физическая модели будущей АИС при помощи программStarUMLи Free Online Diagram Editor.

Затем был разработан программный модуль автоматизированной информационной системы «Учёт выдачи и возврата книг» с собственным интерфейсом и запросами при помощи программ Microsoft Visual Studio 2022 и MSSQLServer Management Studio 18.

Подробно описан и проиллюстрирован функционал и структура самого модуля, а также проведено тестирование готового программного продукта. В последнюю очередь была составлена сопутствующая документация: руководство пользователя и протокол испытаний АИС.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 34.601 – 90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.
2. ГОСТ 34.602 - 2020. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.
3. ГОСТ 19.201 - 78 ЕСПД. Техническое задание. Требования к содержанию и оформлению.
4. ГОСТ 19.202 - 78 ЕСПД. Спецификация. Требования к содержанию и оформлению.
5. ГОСТ Р ИСО/МЭК 12207. Процессы жизненного цикла программных средств.
6. Золотов С.Ю. Проектирование информационных систем: Учебно-методическое пособие. Методические рекомендации для выполнения курсового проекта, лабораторных работ и практических занятий по дисциплине «Проектирование информационных систем» - Томск: ТУСУР, 2013. - 34 с.
7. Шнайдер, Роберт Microsoft SQL Server 6.5. Проектирование высокопроизводительных баз данных; М.: Лори, 2010. - 361 c
8. Петкович, Душан Microsoft SQL Server 2012. Руководство для начинающих / Душан Петкович. - М.: БХВ-Петербург, 2012. - 460 c.
9. Тейлор, Аллен SQL для чайников / Аллен Тейлор. - М.: Вильямс, 2014. - 416 c.
10. Браст, Э.Дж. Разработка приложений на основе Microsoft SQL Server 2008 / Э.Дж. Браст. - М.: Русская Редакция, 2010. - 751 c.
11. Хетагуров, Я. А. Проектирование автоматизированных систем обработки информации и управления (АСОИУ). Учебник / Я.А. Хетагуров. - М.: Бином. Лаборатория знаний, 2015. - 240 c.
12. Кристофер, Д. Маннинг Введение в информационный поиск / Кристофер Д. Маннинг, ПрабхакарРагхаван ,ХайнрихШютце. - М.: Вильямс, 2014. - 528 c.
13. Бишоп Дж. С# в кратком изложении; Бином. Лаборатория знаний - М., 2015. - 234 c.
14. ГриффитсИэн Программирование на C# 5.0; Эксмо - М., 2014. - 580 c.
15. Гуриков С. Р. Введение в программирование на языке Visual C#; ИЛ - Москва, 2013. - 448 c.
16. Шилдт Герберт C# 4.0. Полное руководство; Вильямс - М., 2015. - 291 c.
17. Эндрю Троелсен Язык программирования C# 5.0 и платформа .NET 4.5; Диалектика / Вильямс - М., 2015. - 126 c.

# ПРИЛОЖЕНИЕ А

# ЛИСТИНГ ПРОГРАММЫ

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using System.Data;

using System.Data.SqlClient;

using System.Data.OleDb;

namespace testProject

{

/// <summary>

/// Логика взаимодействия для tovarForm.xaml

/// </summary>

public partial class tovarForm : Window

{

public string testbutton;

classes.classArendaLim arendaLim = new classes.classArendaLim();

classes.classSelectButton classSelectButton = new classes.classSelectButton();

public tovarForm()

{

InitializeComponent();

arendaLim.checktime(this);

}

classes.HideClass animclass = new classes.HideClass();

classes.addClass addClass = new classes.addClass();

private void btnExit\_Click(object sender, RoutedEventArgs e)

{

Environment.Exit(0);

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

MainWindow authorizationWindow = new MainWindow();

authorizationWindow.Show();

this.Close();

}

private void btnArenda\_Click(object sender, RoutedEventArgs e)

{

testbutton = "arendCheck";

listArend.Items.Clear();

classes.ArendaCkass arendaCkass = new classes.ArendaCkass();

arendaCkass.ArendCheck(this);

listCatalog.Visibility = Visibility.Hidden;

listArend.Visibility = Visibility.Visible;

listSrokLast.Visibility = Visibility.Hidden;

btnAdd.Visibility = Visibility.Visible;

btnChange.Visibility = Visibility.Visible;

classSelectButton.selectText = testbutton;

classSelectButton.SelectItemAnim(this);

}

private void Window\_MouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

this.DragMove();

}

public void Button\_Click\_1(object sender, RoutedEventArgs e)

{

testbutton = "romans";

animclass.anim(this);

classes.catalogRomans sqlOpen = new classes.catalogRomans();

sqlOpen.Romans(this);

classSelectButton.selectText = testbutton;

classSelectButton.SelectItemAnim(this);

}

private void Button\_Click\_2(object sender, RoutedEventArgs e)

{

testbutton = "Skazki";

classes.catalogSkazky catalogSkazky = new classes.catalogSkazky();

catalogSkazky.Skazky(this);

animclass.anim(this);

classSelectButton.selectText = testbutton;

classSelectButton.SelectItemAnim(this);

}

private void Button\_Click\_3(object sender, RoutedEventArgs e)

{

testbutton = "historyBooks";

classes.catalogHistoryBooks historyBooks = new classes.catalogHistoryBooks();

historyBooks.History(this);

animclass.anim(this);

classSelectButton.selectText = testbutton;

classSelectButton.SelectItemAnim(this);

}

private void Button\_Click\_4(object sender, RoutedEventArgs e)

{

animclass.anim(this);

}

private void Button\_Click\_5(object sender, RoutedEventArgs e)

{

testbutton = "psyhology";

classes.catalogPsyhology catalogPsyhology = new classes.catalogPsyhology();

catalogPsyhology.Psyhology(this);

animclass.anim(this);

classSelectButton.selectText = testbutton;

classSelectButton.SelectItemAnim(this);

}

private void btnArendaSrok\_Click(object sender, RoutedEventArgs e)

{

testbutton = "lostArend";

listArend.Items.Clear();

classes.classArendaLim arendaLim = new classes.classArendaLim();

arendaLim.checktime(this);

arendaLim.AddProsrBD(this);

listCatalog.Visibility = Visibility.Hidden;

listArend.Visibility = Visibility.Hidden;

listSrokLast.Visibility = Visibility.Visible;

classSelectButton.selectText = testbutton;

classSelectButton.SelectItemAnim(this);

}

private void btnDel\_Click(object sender, RoutedEventArgs e)

{

if (testbutton != null)

{

classes.deleteClass classDel = new classes.deleteClass();

classDel.selectButton = testbutton;

classDel.DeleteItems(this);

}

}

addWindow addWind = new addWindow();

private void btnAdd\_Click(object sender, RoutedEventArgs e)

{

if (testbutton != null && addWind.IsVisible == false)

{

if (testbutton == "arendCheck")

{

addWind.labelPhoto.Content = "Название книги";

addWind.labelNalich.Content = "Имя клиента";

addWind.labelName.Content = "Фамилия клиента";

addWind.labelDescription.Content = "Номер телефона клиента";

addWind.labelPhoto.Content = "Название книги";

}

else

{

addWind.labelPhoto.Content = "Ссылка на фото";

addWind.labelNalich.Content = "Наличие(число)";

addWind.labelName.Content = "Название книги";

addWind.labelDescription.Content = "Описание книги";

addWind.labelPhoto.IsEnabled = false;

addWind.textPassport.IsEnabled = false;

}

addWind.testButton = testbutton;

addWind.Show();

}

else

{

addWind.Hide();

}

}

private void btnLogsShow\_Click(object sender, RoutedEventArgs e)

{

testbutton = "Logs";

classes.classLogsShow logsShow = new classes.classLogsShow();

classSelectButton.selectText = testbutton;

classSelectButton.SelectItemAnim(this);

logsShow.Logs(this);

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace testProject

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void Window\_MouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

this.DragMove();

}

private void btnExit\_Click(object sender, RoutedEventArgs e)

{

Environment.Exit(0);

}

private void btnLogin\_Click(object sender, RoutedEventArgs e)

{

classes.loginClass login = new classes.loginClass();

login.ChkLogin(this);

}

private void passBox\_PreviewMouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

if (passBox.Password != null)

{

labelPass.Content = "Password";

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using System.Data.SqlClient;

using System.Data.OleDb;

namespace testProject.classes

{

public class loginClass

{

public void ChkLogin(MainWindow mainWnd)

{

string pass;

int index = 0;

DataTable Select(string selectSQL) // функция подключения к базе данных и обработка запросов

{

DataTable dataTable = new DataTable("dataBase"); // создаём таблицу в приложении

// подключаемся к базе данных

SqlConnection sqlConnection = new SqlConnection("integrated security=false;user id=delowres\_;password=cannok2019;server=sql.bsite.net\\MSSQL2016;Trusted\_Connection=False;DataBase=delowres\_;");

sqlConnection.Open(); // открываем базу данных

SqlCommand sqlCommand = sqlConnection.CreateCommand(); // создаём команду

sqlCommand.CommandText = selectSQL; // присваиваем команде текст

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(sqlCommand); // создаём обработчик

sqlDataAdapter.Fill(dataTable); // возращаем таблицу с результатом

return dataTable;

}

if (mainWnd.passBox.Password.Length > 0)

{

DataTable dt\_apteka = Select("SELECT \* FROM [dbo].[users] WHERE [pass] = '" + mainWnd.passBox.Password + "'");

if (dt\_apteka.Rows.Count > 0)

{

dt\_apteka = Select("SELECT \* FROM [dbo].[users]"); // данные из БД

for (int i = 0; i < dt\_apteka.Rows.Count; i++) // перебираем данные

{

pass = dt\_apteka.Rows[i][0].ToString();

if (pass == mainWnd.passBox.Password.ToString())

{

index = i;

}

}

tovarForm tovar = new tovarForm();

tovar.Owner = mainWnd;

tovar.labelName.Content = dt\_apteka.Rows[index][1].ToString();

tovar.labelSername.Content = dt\_apteka.Rows[index][2].ToString();

tovar.labelOtchestv.Content = dt\_apteka.Rows[index][3].ToString();

tovar.Show();

mainWnd.Hide();

}

else

{

mainWnd.passBox.Clear();

mainWnd.labelPass.Content = "Неверный пароль";

}

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using System.Data.SqlClient;

using System.Data.OleDb;

namespace testProject.classes

{

class catalogSkazky

{

public void Skazky(tovarForm tovar)

{

tovar.listArend.Visibility = System.Windows.Visibility.Hidden;

tovar.listCatalog.Visibility = System.Windows.Visibility.Visible;

tovar.btnDel.Visibility = System.Windows.Visibility.Visible;

tovar.listCatalog.Items.Clear();

DataTable Select(string selectSQL) // функция подключения к базе данных и обработка запросов

{

DataTable dataTable = new DataTable("dataBase"); // создаём таблицу в приложении

// подключаемся к базе данных

SqlConnection sqlConnection = new SqlConnection("integrated security=false;user id=delowres\_;password=cannok2019;server=sql.bsite.net\\MSSQL2016;Trusted\_Connection=False;DataBase=delowres\_;");

sqlConnection.Open(); // открываем базу данных

SqlCommand sqlCommand = sqlConnection.CreateCommand(); // создаём команду

sqlCommand.CommandText = selectSQL; // присваиваем команде текст

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(sqlCommand); // создаём обработчик

sqlDataAdapter.Fill(dataTable); // возращаем таблицу с результатом

return dataTable;

}

DataTable dt\_apteka = Select("SELECT \* FROM [dbo].[Skazki]"); // данные из БД

for (int i = 0; i < dt\_apteka.Rows.Count; i++) // перебираем данные

{

rents Lol = new rents() // создаём экземпляр класса

{

id\_preparat = dt\_apteka.Rows[i][0].ToString(),

image = dt\_apteka.Rows[i][3].ToString(), // указываем изображение из таблицы

name = dt\_apteka.Rows[i][1].ToString(), // указываем название товара

opisanie = dt\_apteka.Rows[i][4].ToString(), // указываем описание

nalichie = dt\_apteka.Rows[i][2].ToString() // указываем наличие

};

tovar.listCatalog.Items.Add(Lol); // выводим строку в список

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using System.Data.SqlClient;

using System.Data.OleDb;

namespace testProject.classes

{

class addClass

{

public string selectionText;

public void additems(addWindow adw)

{

if (selectionText == "arendCheck")

{

DataTable dataTable = new DataTable("dataBase");

SqlConnection sqlConnection = new SqlConnection("integrated security=false;user id=delowres\_;password=cannok2019;server=sql.bsite.net\\MSSQL2016;Trusted\_Connection=False;DataBase=delowres\_;");

sqlConnection.Open(); // открываем базу данных

SqlCommand sqlCommand = sqlConnection.CreateCommand(); // создаём команду

sqlCommand.CommandText = "Insert into " + selectionText + " (nameBook,name,sername,phone,passport,arendLimit)values(@nb,@na,@sr,@ph,@pa,@ar)";

sqlCommand.Parameters.AddWithValue("@nb", adw.textPhoto.Text);

sqlCommand.Parameters.AddWithValue("@na", adw.textNalichie.Text);

sqlCommand.Parameters.AddWithValue("@sr", adw.textName.Text);

sqlCommand.Parameters.AddWithValue("@ph", adw.textOpisanie.Text);

sqlCommand.Parameters.AddWithValue("@pa", adw.textPassport.Text);

sqlCommand.Parameters.AddWithValue("@ar", adw.dateLast.SelectedDate);// присваиваем команде текст

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(sqlCommand); // создаём обработчик

sqlDataAdapter.Fill(dataTable);

sqlConnection.Close();

DataTable dataTable1 = new DataTable("dataBase");

SqlConnection sqlConnection1 = new SqlConnection("integrated security=false;user id=delowres\_;password=cannok2019;server=sql.bsite.net\\MSSQL2016;Trusted\_Connection=False;DataBase=delowres\_;");

sqlConnection.Open(); // открываем базу данных

SqlCommand sqlCommand1 = sqlConnection1.CreateCommand(); // создаём команду

sqlCommand1.CommandText = "Insert into Logs (nameBook,name,sername,phone,passport,arendLimit)values(@nb,@na,@sr,@ph,@pa,@ar)";

sqlCommand1.Parameters.AddWithValue("@nb", adw.textPhoto.Text);

sqlCommand1.Parameters.AddWithValue("@na", adw.textNalichie.Text);

sqlCommand1.Parameters.AddWithValue("@sr", adw.textName.Text);

sqlCommand1.Parameters.AddWithValue("@ph", adw.textOpisanie.Text);

sqlCommand1.Parameters.AddWithValue("@pa", adw.textPassport.Text);

sqlCommand1.Parameters.AddWithValue("@ar", adw.dateLast.SelectedDate);// присваиваем команде текст

SqlDataAdapter sqlDataAdapter1 = new SqlDataAdapter(sqlCommand1); // создаём обработчик

sqlDataAdapter1.Fill(dataTable1);

sqlConnection1.Close();

}

else

{

DataTable dataTable = new DataTable("dataBase");

SqlConnection sqlConnection = new SqlConnection("integrated security=false;user id=delowres\_;password=cannok2019;server=sql.bsite.net\\MSSQL2016;Trusted\_Connection=False;DataBase=delowres\_;");

sqlConnection.Open(); // открываем базу данных

SqlCommand sqlCommand = sqlConnection.CreateCommand(); // создаём команду

sqlCommand.CommandText = "Insert into " + selectionText + " (Name,Nalichie,Image,Opisanie)values(@nm,@nl,@nm,@op)";

sqlCommand.Parameters.AddWithValue("@ph", adw.textPhoto.Text);

sqlCommand.Parameters.AddWithValue("@nl", adw.textNalichie.Text);

sqlCommand.Parameters.AddWithValue("@nm", adw.textName.Text);

sqlCommand.Parameters.AddWithValue("@op", adw.textOpisanie.Text);

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(sqlCommand); // создаём обработчик

sqlDataAdapter.Fill(dataTable);

}

adw.textNalichie.Clear();

adw.textName.Clear();

adw.textOpisanie.Clear();

adw.textPassport.Clear();

adw.textPhoto.Clear();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using System.Data.SqlClient;

using System.Data.OleDb;

namespace testProject.classes

{

class ArendaCkass

{

public void ArendCheck(tovarForm tovar)

{

DataTable Select(string selectSQL) // функция подключения к базе данных и обработка запросов

{

DataTable dataTable = new DataTable("dataBase"); // создаём таблицу в приложении

// подключаемся к базе данных

SqlConnection sqlConnection = new SqlConnection("integrated security=false;user id=delowres\_;password=cannok2019;server=sql.bsite.net\\MSSQL2016;Trusted\_Connection=False;DataBase=delowres\_;");

sqlConnection.Open(); // открываем базу данных

SqlCommand sqlCommand = sqlConnection.CreateCommand(); // создаём команду

sqlCommand.CommandText = selectSQL; // присваиваем команде текст

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(sqlCommand); // создаём обработчик

sqlDataAdapter.Fill(dataTable); // возращаем таблицу с результатом

return dataTable;

}

DataTable dt\_apteka = Select("SELECT \* FROM [dbo].[arendCheck]"); // данные из БД

for (int i = 0; i < dt\_apteka.Rows.Count; i++) // перебираем данные

{

rentsArendClass arend = new rentsArendClass();

{

arend.id\_arend = dt\_apteka.Rows[i][0].ToString(); // указываем название товара

arend.nameBook = dt\_apteka.Rows[i][1].ToString();

arend.name = dt\_apteka.Rows[i][2].ToString();

arend.sername = dt\_apteka.Rows[i][3].ToString();

arend.phone = dt\_apteka.Rows[i][4].ToString();

arend.passport = dt\_apteka.Rows[i][5].ToString();

arend.arendLimit = dt\_apteka.Rows[i][6].ToString();

}

tovar.listArend.Items.Add(arend); // выводим строку в список

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using System.Data.SqlClient;

using System.Data.OleDb;

namespace testProject.classes

{

class catalogHistoryBooks

{

public void History(tovarForm tovar)

{

tovar.listArend.Visibility = System.Windows.Visibility.Hidden;

tovar.listCatalog.Visibility = System.Windows.Visibility.Visible;

tovar.btnDel.Visibility = System.Windows.Visibility.Visible;

tovar.listCatalog.Items.Clear();

DataTable Select(string selectSQL) // функция подключения к базе данных и обработка запросов

{

DataTable dataTable = new DataTable("dataBase"); // создаём таблицу в приложении

// подключаемся к базе данных

SqlConnection sqlConnection = new SqlConnection("integrated security=false;user id=delowres\_;password=cannok2019;server=sql.bsite.net\\MSSQL2016;Trusted\_Connection=False;DataBase=delowres\_;");

sqlConnection.Open(); // открываем базу данных

SqlCommand sqlCommand = sqlConnection.CreateCommand(); // создаём команду

sqlCommand.CommandText = selectSQL; // присваиваем команде текст

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(sqlCommand); // создаём обработчик

sqlDataAdapter.Fill(dataTable); // возращаем таблицу с результатом

return dataTable;

}

DataTable dt\_apteka = Select("SELECT \* FROM [dbo].[historyBooks]"); // данные из БД

for (int i = 0; i < dt\_apteka.Rows.Count; i++) // перебираем данные

{

rents Lol = new rents() // создаём экземпляр класса

{

id\_preparat = dt\_apteka.Rows[i][0].ToString(),

image = dt\_apteka.Rows[i][3].ToString(), // указываем изображение из таблицы

name = dt\_apteka.Rows[i][1].ToString(), // указываем название товара

opisanie = dt\_apteka.Rows[i][4].ToString(), // указываем описание

nalichie = dt\_apteka.Rows[i][2].ToString() // указываем наличие

};

tovar.listCatalog.Items.Add(Lol); // выводим строку в список

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using System.Data.SqlClient;

using System.Data.OleDb;

namespace testProject.classes

{

class catalogPsyhology

{

public void Psyhology(tovarForm tovar)

{

tovar.listCatalog.Visibility = System.Windows.Visibility.Visible;

tovar.listArend.Visibility = System.Windows.Visibility.Hidden;

tovar.btnDel.Visibility = System.Windows.Visibility.Visible;

tovar.listCatalog.Items.Clear();

DataTable Select(string selectSQL) // функция подключения к базе данных и обработка запросов

{

DataTable dataTable = new DataTable("dataBase"); // создаём таблицу в приложении

// подключаемся к базе данных

SqlConnection sqlConnection = new SqlConnection("integrated security=false;user id=delowres\_;password=cannok2019;server=sql.bsite.net\\MSSQL2016;Trusted\_Connection=False;DataBase=delowres\_;");

sqlConnection.Open(); // открываем базу данных

SqlCommand sqlCommand = sqlConnection.CreateCommand(); // создаём команду

sqlCommand.CommandText = selectSQL; // присваиваем команде текст

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(sqlCommand); // создаём обработчик

sqlDataAdapter.Fill(dataTable); // возращаем таблицу с результатом

return dataTable;

}

DataTable dt\_apteka = Select("SELECT \* FROM [dbo].[psyhology]"); // данные из БД

for (int i = 0; i < dt\_apteka.Rows.Count; i++) // перебираем данные

{

rents Lol = new rents() // создаём экземпляр класса

{

id\_preparat = dt\_apteka.Rows[i][0].ToString(),

image = dt\_apteka.Rows[i][3].ToString(), // указываем изображение из таблицы

name = dt\_apteka.Rows[i][1].ToString(), // указываем название товара

opisanie = dt\_apteka.Rows[i][4].ToString(), // указываем описание

nalichie = dt\_apteka.Rows[i][2].ToString() // указываем наличие

};

tovar.listCatalog.Items.Add(Lol); // выводим строку в список

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using System.Data.SqlClient;

using System.Data.OleDb;

namespace testProject.classes

{

class classArendaLim

{

public void AddProsrBD(tovarForm tovar)

{

for (int i = 0; i < tovar.listSrokLast.Items.Count; i++)

{

var selectedItem = (dynamic)tovar.listSrokLast.Items[i];

var id = selectedItem.id\_arend;

var nameBook = selectedItem.nameBook;

var name = selectedItem.name;

var sername = selectedItem.sername;

var phone = selectedItem.phone;

var passport = selectedItem.passport;

var arendLimit = selectedItem.arendLimit;

DataTable Select(string selectSQL) // функция подключения к базе данных и обработка запросов

{

DataTable dataTable1 = new DataTable("dataBase"); // создаём таблицу в приложении

// подключаемся к базе данных

SqlConnection sqlConnection1 = new SqlConnection("integrated security=false;user id=delowres\_;password=cannok2019;server=sql.bsite.net\\MSSQL2016;Trusted\_Connection=False;DataBase=delowres\_;");

sqlConnection1.Open(); // открываем базу данных

SqlCommand sqlCommand1 = sqlConnection1.CreateCommand(); // создаём команду

sqlCommand1.CommandText = selectSQL; // присваиваем команде текст

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(sqlCommand1); // создаём обработчик

sqlDataAdapter.Fill(dataTable1); // возращаем таблицу с результатом

return dataTable1;

}

DataTable dataTable = new DataTable("dataBase"); // создаём таблицу в приложении

// подключаемся к базе данных

SqlConnection sqlConnection = new SqlConnection("integrated security=false;user id=delowres\_;password=cannok2019;server=sql.bsite.net\\MSSQL2016;Trusted\_Connection=False;DataBase=delowres\_;");

sqlConnection.Open(); // открываем базу данных

SqlCommand sqlCommand = sqlConnection.CreateCommand();

DataTable delowers\_ = Select ("SELECT \* FROM [dbo].[lostArend] WHERE id\_arend = '" + id + "'");

for (int z = 0; z < delowers\_.Rows.Count; z++)

{

rentsArendClass rentsArend = new rentsArendClass();

{

rentsArend.id\_arend = delowers\_.Rows[i][0].ToString();

if (rentsArend.id\_arend != id)

{

sqlCommand.CommandText = "Insert into lostArend (id\_arend,nameBook,name,sername,phone,passport,arendLimit)values(@id,@nb,@na,@sr,@ph,@pa,@ar)";

sqlCommand.Parameters.AddWithValue("@id", id);

sqlCommand.Parameters.AddWithValue("@nb", nameBook);

sqlCommand.Parameters.AddWithValue("@na", name);

sqlCommand.Parameters.AddWithValue("@sr", sername);

sqlCommand.Parameters.AddWithValue("@ph", phone);

sqlCommand.Parameters.AddWithValue("@pa", passport);

sqlCommand.Parameters.AddWithValue("@ar", arendLimit);// присваиваем команде текст

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(sqlCommand); // создаём обработчик

sqlDataAdapter.Fill(dataTable);

}

}

}

}

}

public void checktime(tovarForm tovar)

{

tovar.listSrokLast.Items.Clear();

tovar.btnAdd.Visibility = System.Windows.Visibility.Hidden;

tovar.btnChange.Visibility = System.Windows.Visibility.Hidden;

tovar.btnDel.Visibility = System.Windows.Visibility.Visible;

int schetProsrochka = 0;

DateTime date;

DataTable Select(string selectSQL) // функция подключения к базе данных и обработка запросов

{

DataTable dataTable = new DataTable("dataBase"); // создаём таблицу в приложении

// подключаемся к базе данных

SqlConnection sqlConnection = new SqlConnection("integrated security=false;user id=delowres\_;password=cannok2019;server=sql.bsite.net\\MSSQL2016;Trusted\_Connection=False;DataBase=delowres\_;");

sqlConnection.Open(); // открываем базу данных

SqlCommand sqlCommand = sqlConnection.CreateCommand(); // создаём команду

sqlCommand.CommandText = selectSQL;

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(sqlCommand); // создаём обработчик

sqlDataAdapter.Fill(dataTable); // возращаем таблицу с результатом

return dataTable;

}

DataTable dt\_apteka = Select("SELECT \* FROM [dbo].[arendCheck]"); // данные из БД

for (int i = 0; i < dt\_apteka.Rows.Count; i++) // перебираем данные

{

rentsArendClass arend = new rentsArendClass();

{

date = (DateTime)dt\_apteka.Rows[i][6];

if (DateTime.Today > date)

{

schetProsrochka++;

tovar.labelArend.Content = schetProsrochka.ToString();

arend.arendLimit = dt\_apteka.Rows[i][6].ToString();

arend.id\_arend = dt\_apteka.Rows[i][0].ToString(); // указываем название товара

arend.nameBook = dt\_apteka.Rows[i][1].ToString();

arend.name = dt\_apteka.Rows[i][2].ToString();

arend.sername = dt\_apteka.Rows[i][3].ToString();

arend.phone = dt\_apteka.Rows[i][4].ToString();

arend.passport = dt\_apteka.Rows[i][5].ToString();

arend.arendLimit = dt\_apteka.Rows[i][6].ToString();

tovar.listSrokLast.Items.Add(arend); // выводим строку в список

}

}

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using System.Data.SqlClient;

using System.Data.OleDb;

namespace testProject.classes

{

class deleteClass

{

public string selectButton;

public void DeleteItems(tovarForm tovar)

{

if (selectButton == "lostArend")

{

dynamic itemSelectList = tovar.listSrokLast.SelectedItem;

var id = itemSelectList.id\_arend;

DataTable dataTable = new DataTable("dataBase");

SqlConnection sqlConnection = new SqlConnection("integrated security=false;user id=delowres\_;password=cannok2019;server=sql.bsite.net\\MSSQL2016;Trusted\_Connection=False;DataBase=delowres\_;");

sqlConnection.Open(); // открываем базу данных

SqlCommand sqlCommand = sqlConnection.CreateCommand(); // создаём команду

sqlCommand.CommandText = "DELETE FROM " + selectButton + " WHERE [Id\_arend] = @id";

sqlCommand.CommandText = "DELETE FROM [arendCheck] WHERE [Id\_arend] = @id";

sqlCommand.Parameters.AddWithValue("@id", id); // присваиваем команде текст

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(sqlCommand); // создаём обработчик

sqlDataAdapter.Fill(dataTable);

}

else if (selectButton == "arendCheck")

{

// tovar.listArend.Items.Refresh();

dynamic itemSelectList = tovar.listArend.SelectedItem;

var id = itemSelectList.id\_arend;

DataTable dataTable = new DataTable("dataBase");

SqlConnection sqlConnection = new SqlConnection("integrated security=false;user id=delowres\_;password=cannok2019;server=sql.bsite.net\\MSSQL2016;Trusted\_Connection=False;DataBase=delowres\_;");

sqlConnection.Open(); // открываем базу данных

SqlCommand sqlCommand = sqlConnection.CreateCommand(); // создаём команду

sqlCommand.CommandText = "DELETE FROM " + selectButton + " WHERE [Id\_arend] = @id";

sqlCommand.Parameters.AddWithValue("@id", id); // присваиваем команде текст

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(sqlCommand); // создаём обработчик

sqlDataAdapter.Fill(dataTable);

}

else

{

dynamic itemSelectList = tovar.listCatalog.SelectedItem;

var id = itemSelectList.id\_preparat;

DataTable dataTable = new DataTable("dataBase");

SqlConnection sqlConnection = new SqlConnection("integrated security=false;user id=delowres\_;password=cannok2019;server=sql.bsite.net\\MSSQL2016;Trusted\_Connection=False;DataBase=delowres\_;");

sqlConnection.Open(); // открываем базу данных

SqlCommand sqlCommand = sqlConnection.CreateCommand(); // создаём команду

sqlCommand.CommandText = "DELETE FROM " + selectButton + " WHERE [id\_Preparat] = @id";

sqlCommand.Parameters.AddWithValue("@id", id); // присваиваем команде текст

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(sqlCommand); // создаём обработчик

sqlDataAdapter.Fill(dataTable);

}

geniusClass genius = new geniusClass();

genius.selectText = selectButton;

genius.startAll(tovar);

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace testProject.classes

{

class rentsArendClass

{

public string id\_arend { get; set; }

public string nameBook { get; set; }

public string name { get; set; }

public string sername { get; set; }

public string phone { get; set; }

public string passport { get; set; }

public string arendLimit { get; set; }

}

}