Государственное бюджетное профессиональное образовательное учреждение

Самарской области

«Тольяттинский социально-экономический колледж»

**КУРСОВАЯ РАБОТА**

**РАЗРАБОТКА И ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «РАСЧЕТ СТОИМОСТИ УСЛУГ, ПРЕДОСТАВЛЯЕМЫХ КЛИЕНТАМ»**

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

**МДК 05.02 Разработка кода информационных систем**

**09.02.07 Информационные системы и программирование**

|  |  |  |  |
| --- | --- | --- | --- |
| Студент |  | **/** | В.А. Косточкин |
|  | *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 систем модуля, выполняемой в рамках МДК.05.02 Разработка кода информационных систем**

студента группы ИСП-31

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Косточкина Вадима Александровича\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Фамилия Имя отчество студента*

Тема курсовой работы: «Разработка информационной системы **«**Расчет стоимости услуг, предоставляемых клиентам»

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)

2.1 Разработка интерфейса

2.2 Разработка руководства пользователя

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

**КАЛЕНДАРНЫЙ ПЛАН**

выполнения курсовой работы

Студентом 3 курса группы ИСП-31

По теме Разработка и проектирование ИС «Расчет стоимости услуг, предоставляемых клиентам»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № 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 |  | **/** | В.А. Косточкин |
|  | *подпись* |  | *И.О. Фамилия* |
| \_\_.\_\_.2022 г. |  |  |  |
| Руководитель |  | **/** | Е.В. Плюснина |
|  | *подпись* |  | *И.О. Фамилия* |
| \_\_.\_\_.2022 г. |  |  |  |

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 5](#_Toc107178560)

[1 РАЗРАБОТКА БАЗЫ ДАННЫХ ИНФОРМАЦИОННОЙ СИСТЕМЫ РАСЧЕТ СТОИМОСТИ УСЛУГ, ПРЕДОСТАВЛЯЕМЫХ КЛИЕНТАМ 7](#_Toc107178561)

[1.1 Разработка физической структуры БД 7](#_Toc107178562)

[1.2 Разработка запросов ИС 10](#_Toc107178563)

[Создание остальных запросов базы данных для информационной системы происходит аналогично примеру, описанному выше. 12](#_Toc107178564)

[2 РАЗРАБОТКА ИНТЕРФЕЙСА ИНФОРМАЦИОННОЙ СИСТЕМЫ РАСЧЕТ СТОИМОСТИ УСЛУГ, ПРЕДОСТАВЛЯЕМЫХ КЛИЕНТАМ 13](#_Toc107178565)

[2.1 Разработка интерфейса 13](#_Toc107178566)

[2.2 Разработка руководства пользователя 20](#_Toc107178567)

[ЗАКЛЮЧЕНИЕ 23](#_Toc107178568)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 24](#_Toc107178569)

[ПРИЛОЖЕНИЕ А 26](#_Toc107178570)

[ЛИСТИНГ ПРОГРАММЫ 26](#_Toc107178571)

# 

# ВВЕДЕНИЕ

Для достижения максимальных успехов в своей деятельности, любому предприятию необходимо точно понимать свои затраты, прибыли, ресурсы, бизнес процессы и многое другое. Наглядная информация о происходящем поможет глубже проанализировать процесс и поможет сделать правильные выводы, что в конечном итоге приведет к росту продаж, увеличению объема производства, повысит общую эффективность.

Не секрет, что лидерами рынка становятся наиболее эффективные предприятия, имеющие минимальные издержки, высочайший уровень производительности труда и полностью контролируемые и четко отлаженные процессы. Ни что так не способствует контролю и анализу деятельности на предприятии как внедрение комплексной автоматизированной информационной системы (АИС).

Объектом исследования является Аптека, которая является учреждением здравоохранения, основной задачей которого обеспечение населения, учреждения здравоохранения и других учреждений, организаций, предприятий продукцией медико-фармацевтического назначения.

При изучении предметной области, выяснилось, что информационная система «Аптека» 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.

Аптека создается с целью обеспечения населения и лечебно-профилактические учреждения лекарственными средствами и изделиями медицинского назначения, производства лекарственных форм, предоставления услуг, а также реализация на основании полученной прибыли интересов трудового коллектива.

Выше изложенное в целом на теоретико-методологическом уровне определило проблему настоящего исследования: разработка программного модуля «Расчет стоимости» в программе MS Visual Studio 2019.

Целью курсовой работы является разработка и проектирование информационной системы «Аптека».

Для достижения поставленной цели были определены следующие задачи:

* разработать базы данных информационной системы;
* разработать физическую структуру базы данных;
* разработать запросы ИС;
* разработать интерфейс ИС;
* разработать руководство пользователя.

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 «Расчету стоимости».

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

# 1 РАЗРАБОТКА БАЗЫ ДАННЫХ ИНФОРМАЦИОННОЙ СИСТЕМЫ РАСЧЕТ СТОИМОСТИ УСЛУГ, ПРЕДОСТАВЛЯЕМЫХ КЛИЕНТАМ

## 1.1 Разработка физической структуры БД

Чтобы создать новую базу данных, откройте 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.

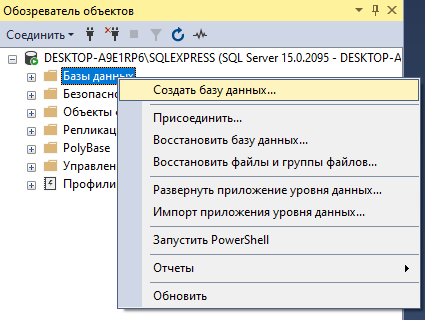


Рисунок 1.1 – Окно обозреватель объектов

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

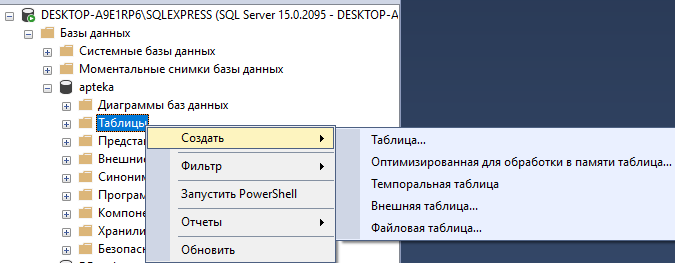


Рисунок 1.2 – Окно создания таблицы

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

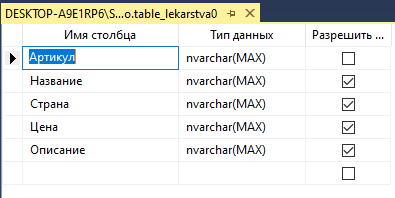


Рисунок 1.3 – Таблица «Лекарства»

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:

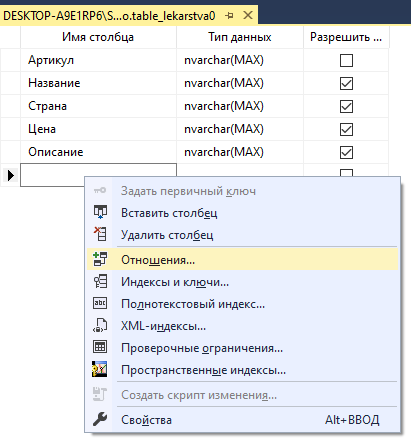


Рисунок 1.4 – Окно создания связей по внешнему ключу

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:

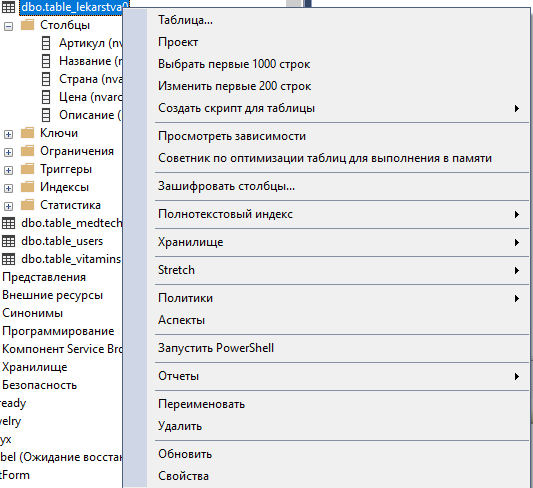


Рисунок 1.5 – Окно выбора свойств таблицы «Лекарства»

Выберите нужную схему из раскрывающегося списка «Схема».

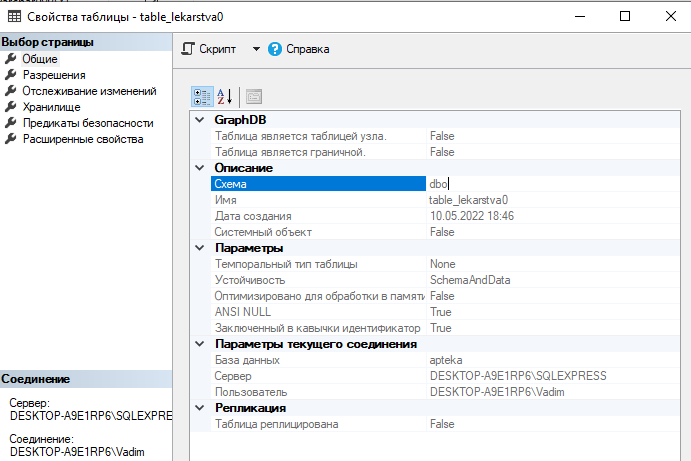


Рисунок 1.6 – Окно свойств таблицы «Лекарства»

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…».

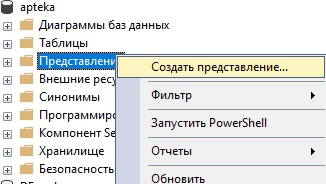


Рисунок 1.7 – Окно создания представления

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.

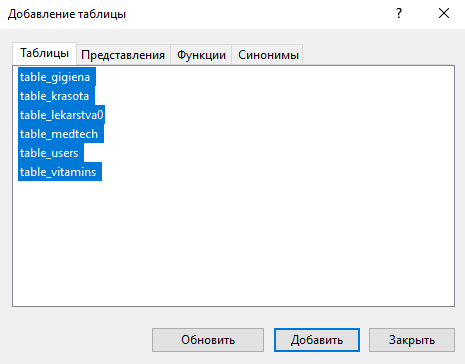


Рисунок 1.8 – Диалоговое окно добавление таблицы

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:

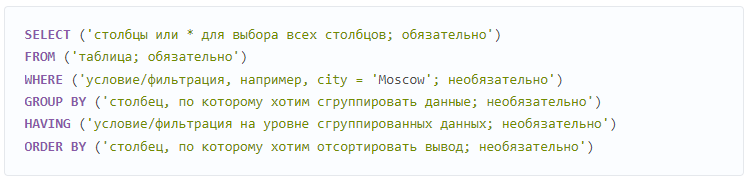


Рисунок 1.9 – Общая структура запроса

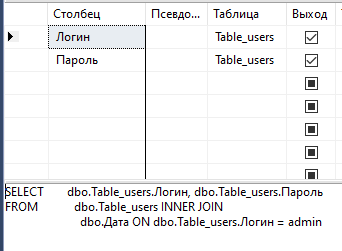


Рисунок 1.10 – Запрос Select для информационной системы

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

# 

# 2 РАЗРАБОТКА ИНТЕРФЕЙСА ИНФОРМАЦИОННОЙ СИСТЕМЫ РАСЧЕТ СТОИМОСТИ УСЛУГ, ПРЕДОСТАВЛЯЕМЫХ КЛИЕНТАМ

## 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.

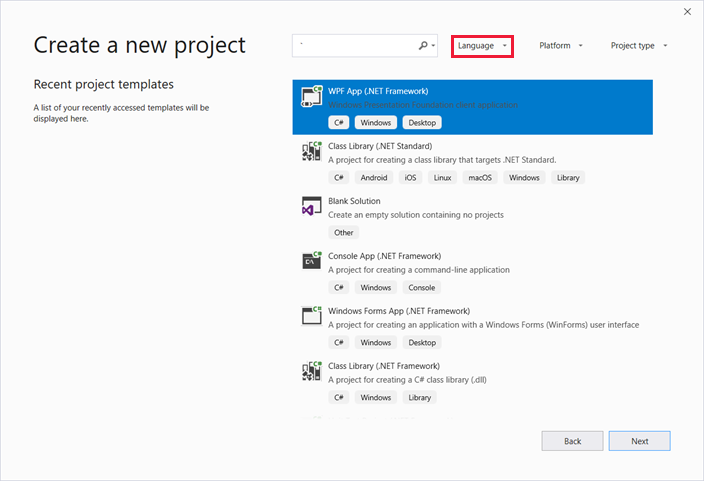


Рисунок 2.1 – Окно создания приложения WPF

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

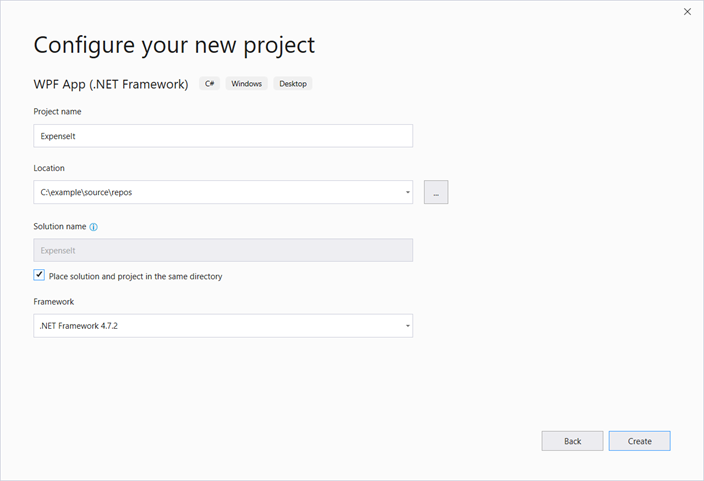


Рисунок 2.2 – Окно создания приложения WPF

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, Image для визуального восприятия информации; TextBox, PasswordBox для ввода логина и пароля; CheckBox для просмотра пароля; Button для закрытия окна и для входа в систему.

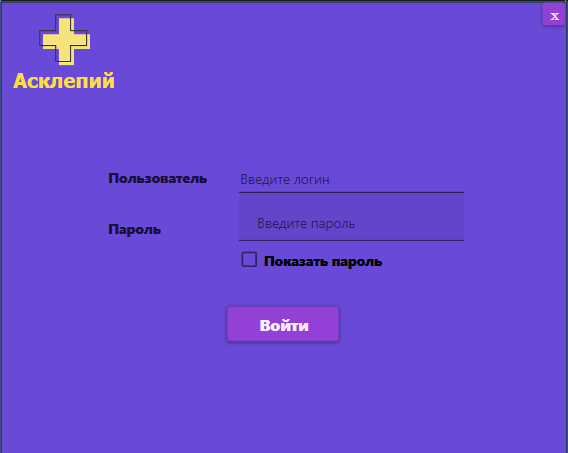


Рисунок 2.3 – Интерфейс окна «Авторизация»

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 окна авторизации



Рисунок 2.6 – XAML-код окна авторизации

Интерфейс окна авторизации будет выглядеть следующим образом:

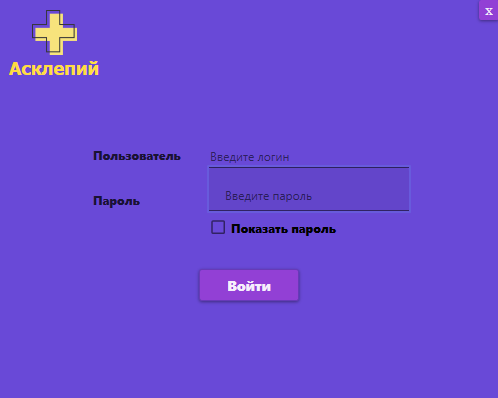


Рисунок 2.7 – Интерфейс окна «Авторизация» с проверкой ввода логина или пароля

Следом перейдем к созданию кода на языке C# для окна авторизации. Нажмем правой кнопкой мыши по окну, выберем пункт «Посмотреть код».

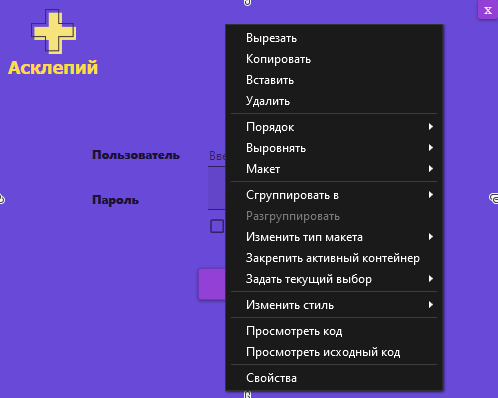


Рисунок 2.8 – Свойства окна

Пропишем появление заставки приложение перед инициализацией компонентов. Затем сделаем элемент CheckBox «Показать пароль» функциональным, прописав видимые и невидимые символы в зависимости от флажка на CheckBox.

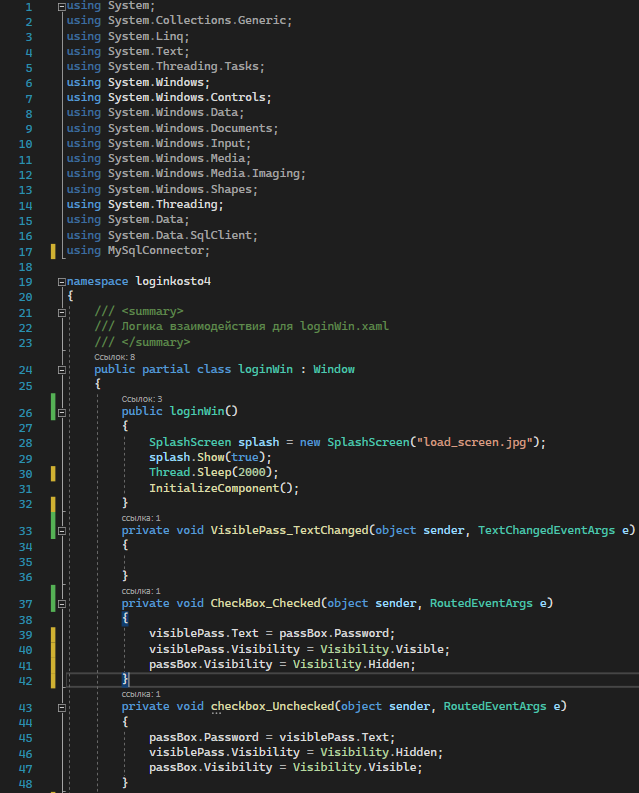


Рисунок 2.9 – C#-код окна

Далее поставим условия входа – при удачном вводе логина и пароля, пользователь попадает на главное окно (winZakaz). При неправильном появляется ошибка ввода логина или пароля и открывается окно (winCaptcha) с проверкой на робота (Captcha) с возможностью её обновления.

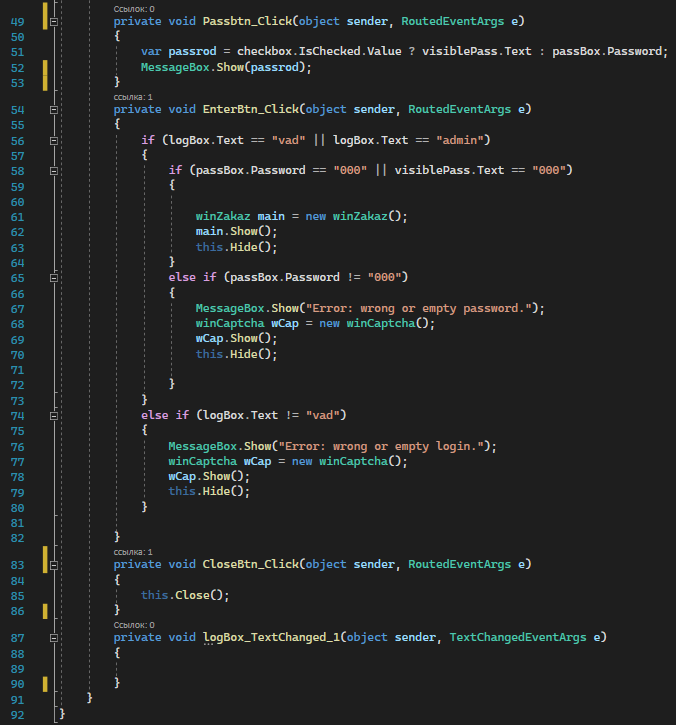


Рисунок 2.10 – C#-код окна

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».

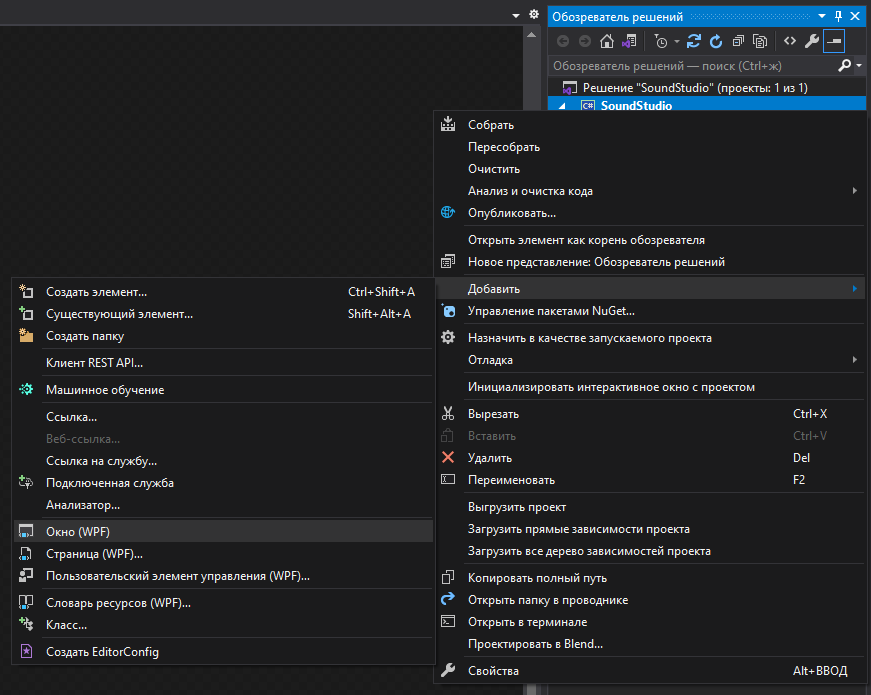


Рисунок 2.11 – Добавление главного окна проекта

После нужно указать название и нажать кнопку «Далее».

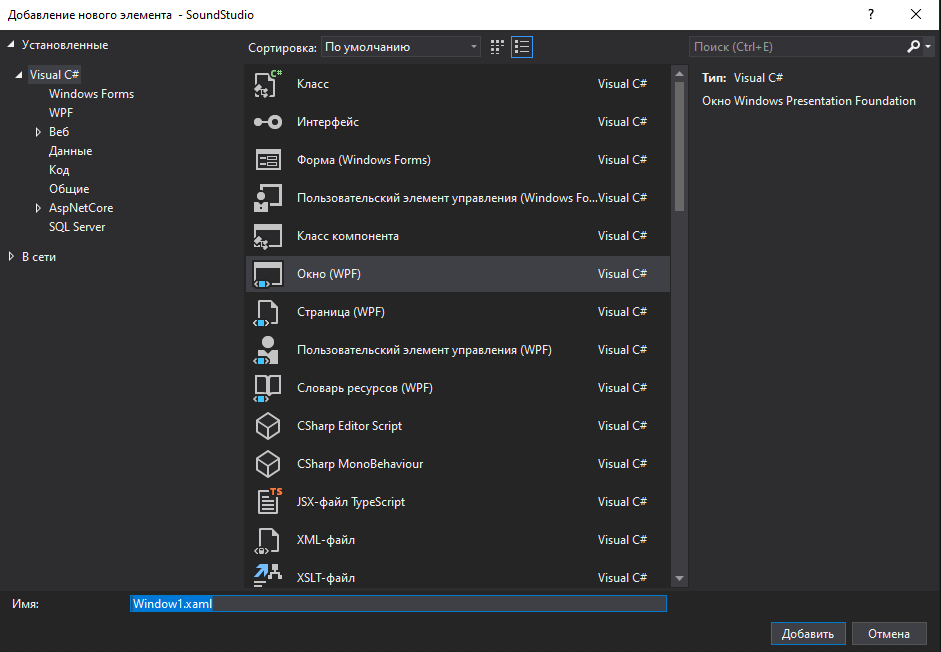


Рисунок 2.12 – Добавление имени главного окна проекта

Интерфейс главного окна и программный код разрабатываются аналогично описанному выше примеру.

## 2.2 Разработка руководства пользователя

Информационная система «Аптека» представляет собой программный комплекс для ознакомления заказчиков и потенциальных клиентов с деятельностью и юридической информацией об организации ООО «Аптека».

Данное руководство предназначено для пользователей приложения, установленного на устройствах, работающих под управлением операционной системы Windows 10.

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 8Гб 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/VGA/HDMI;
    - 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 логин и пароль в открывшемся окне при входе в приложение, затем нажмите кнопку «ВОЙТИ». Для просмотра введенного вами пароля поставьте галочку рядом с «Показать пароль».

Открытие главного окна обозначит успешный вход в приложение.

Диалоговое окно «Error: wrong or empty login/password.» обозначит ошибку ввода логина или пароля. Нажмите «Ок» и попробуйте ввести данные заново, дополнительно пройдя проверку (Captcha). Если вам сложно различить символы в предложенной автоматически проверке, вы можете обновить ее, нажав рядом кнопку «Обновить».

Для закрытия окна авторизации нажмите кнопку «Х» в правом верхнем углу экрана.

В боковом меню, расположенном на главном окне слева, представлены категории товаров:

* + - Лекарства;
    - Красота;
    - Гигиена;
    - Витамины;
    - Медтехника.

Для возвращения на окно авторизации с главного окна заказа нажмите кнопку «<--» слева от «Х».

Для закрытия главного окна нажмите кнопку «X» в правом верхнем углу экрана.

# ЗАКЛЮЧЕНИЕ

В первой части курсовой работы была произведена разработка базы данных информационной системы «Аптека». Была разработана физическая структура базы данных для информационной системы. Также, был рассмотрен процесс разработки запросов ИС «Аптека». В дальнейшем создана база данных системы в программной среде MS SQL Server Management Studio 18.

Во второй части курсовой работы на основе рассмотренных процессов отдела «Аптека» был разработан и подробно описан интерфейс информационной системы в программе MS Visual Studio 19, а также было разработано руководство пользователя.

Затем был разработан программный модуль автоматизированной информационной системы «Аптека» с собственным интерфейсом и запросами при помощи программ Microsoft Visual Studio 2019 и MS SQL Server 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.

# 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.Threading;

using System.Data;

using System.Data.SqlClient;

using MySqlConnector;

namespace loginkosto4

{

/// <summary>

/// Логика взаимодействия для loginWin.xaml

/// </summary>

public partial class loginWin : Window

{

public loginWin()

{

SplashScreen splash = new SplashScreen("load\_screen.jpg");

splash.Show(true);

Thread.Sleep(2000);

InitializeComponent();

}

private void VisiblePass\_TextChanged(object sender, TextChangedEventArgs e)

{

}

private void CheckBox\_Checked(object sender, RoutedEventArgs e)

{

visiblePass.Text = passBox.Password;

visiblePass.Visibility = Visibility.Visible;

passBox.Visibility = Visibility.Hidden;

}

private void checkbox\_Unchecked(object sender, RoutedEventArgs e)

{

passBox.Password = visiblePass.Text;

visiblePass.Visibility = Visibility.Hidden;

passBox.Visibility = Visibility.Visible;

}

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

{

var passrod = checkbox.IsChecked.Value ? visiblePass.Text : passBox.Password;

MessageBox.Show(passrod);

}

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

{

if (logBox.Text == "vad" || logBox.Text == "admin")

{

if (passBox.Password == "000" || visiblePass.Text == "000")

{

winZakaz main = new winZakaz();

main.Show();

this.Hide();

}

else if (passBox.Password != "000")

{

MessageBox.Show("Error: wrong or empty password.");

winCaptcha wCap = new winCaptcha();

wCap.Show();

this.Hide();

}

}

else if (logBox.Text != "vad")

{

MessageBox.Show("Error: wrong or empty login.");

winCaptcha wCap = new winCaptcha();

wCap.Show();

this.Hide();

}

}

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

{

this.Close();

}

private void logBox\_TextChanged\_1(object sender, TextChangedEventArgs e)

{

}

}

}

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;

namespace loginkosto4

{

/// <summary>

/// Логика взаимодействия для winZakaz.xaml

/// </summary>

public partial class winZakaz : Window

{

public winZakaz()

{

InitializeComponent();

ListApteka.Visibility = Visibility.Hidden;

}

public class lekarstvaS

{

public string codc { get; set; }

public string name { get; set; }

public string country { get; set; }

public string price { get; set; }

public string basic { get; set; }

}

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

{

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

SqlConnection sqlConnection = new SqlConnection("server=DESKTOP-A9E1RP6\\SQLEXPRESS;Trusted\_Connection=Yes;DataBase=apteka;");

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;

}

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

{

}

private void ListView\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

}

private void ListLekarstva\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

}

// КНОПКА ЛЕКАРСТВА

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

{

ListApteka.Items.Clear();

//ListKrasota.Visibility = Visibility.Hidden;

ListApteka.Visibility = Visibility.Visible;

LoadLekarstva();

}

void LoadLekarstva()

{

DataTable dt\_lekarstva = Select("SELECT \* FROM [dbo].[table\_lekarstva0]");

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

{

lekarstvaS dataUser = new lekarstvaS() // создаём экземпляр класса

{

codc = dt\_lekarstva.Rows[i][0].ToString(),

name = dt\_lekarstva.Rows[i][1].ToString(),

country = dt\_lekarstva.Rows[i][2].ToString(),

price = dt\_lekarstva.Rows[i][3].ToString(),

basic = dt\_lekarstva.Rows[i][4].ToString(),

};

ListApteka.Items.Add(dataUser); // выводим строку в список

}

}

// КНОПКА КРАСОТА

private void btnLekarstva\_Krasota\_Click(object sender, RoutedEventArgs e)

{

ListApteka.Items.Clear();

//ListLekarstva.Visibility = Visibility.Hidden;

//ListKrasota.Visibility = Visibility.Visible;

ListApteka.Visibility = Visibility.Visible;

LoadKrasota();

}

void LoadKrasota()

{

DataTable dt\_krasota = Select("SELECT \* FROM [dbo].[table\_krasota]");

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

{

lekarstvaS dataUser = new lekarstvaS() // создаём экземпляр класса

{

codc = dt\_krasota.Rows[i][0].ToString(),

name = dt\_krasota.Rows[i][1].ToString(),

country = dt\_krasota.Rows[i][2].ToString(),

price = dt\_krasota.Rows[i][3].ToString(),

basic = dt\_krasota.Rows[i][4].ToString(),

};

ListApteka.Items.Add(dataUser); // выводим строку в список

}

}

// КНОПКА ГИГИЕНА

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

{

ListApteka.Items.Clear();

//ListLekarstva.Visibility = Visibility.Hidden;

//ListKrasota.Visibility = Visibility.Visible;

ListApteka.Visibility = Visibility.Visible;

LoadGigiena();

}

void LoadGigiena()

{

DataTable dt\_gigiena = Select("SELECT \* FROM [dbo].[table\_gigiena]");

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

{

lekarstvaS dataUser = new lekarstvaS() // создаём экземпляр класса

{

codc = dt\_gigiena.Rows[i][0].ToString(),

name = dt\_gigiena.Rows[i][1].ToString(),

country = dt\_gigiena.Rows[i][2].ToString(),

price = dt\_gigiena.Rows[i][3].ToString(),

basic = dt\_gigiena.Rows[i][4].ToString(),

};

ListApteka.Items.Add(dataUser); // выводим строку в список

}

}

// КНОПКА ВИТАМИНЫ

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

{

ListApteka.Items.Clear();

//ListLekarstva.Visibility = Visibility.Hidden;

//ListKrasota.Visibility = Visibility.Visible;

ListApteka.Visibility = Visibility.Visible;

LoadVitamins();

}

void LoadVitamins()

{

DataTable dt\_vitamins = Select("SELECT \* FROM [dbo].[table\_vitamins]");

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

{

lekarstvaS dataUser = new lekarstvaS() // создаём экземпляр класса

{

codc = dt\_vitamins.Rows[i][0].ToString(),

name = dt\_vitamins.Rows[i][1].ToString(),

country = dt\_vitamins.Rows[i][2].ToString(),

price = dt\_vitamins.Rows[i][3].ToString(),

basic = dt\_vitamins.Rows[i][4].ToString(),

};

ListApteka.Items.Add(dataUser); // выводим строку в список

}

}

// КНОПКА МЕДТЕХНИКА

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

{

ListApteka.Items.Clear();

//ListLekarstva.Visibility = Visibility.Hidden;

//ListKrasota.Visibility = Visibility.Visible;

ListApteka.Visibility = Visibility.Visible;

LoadMedtech();

}

void LoadMedtech()

{

DataTable dt\_medtech = Select("SELECT \* FROM [dbo].[table\_medtech]");

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

{

lekarstvaS dataUser = new lekarstvaS() // создаём экземпляр класса

{

codc = dt\_medtech.Rows[i][0].ToString(),

name = dt\_medtech.Rows[i][1].ToString(),

country = dt\_medtech.Rows[i][2].ToString(),

price = dt\_medtech.Rows[i][3].ToString(),

basic = dt\_medtech.Rows[i][4].ToString(),

};

ListApteka.Items.Add(dataUser); // выводим строку в список

}

}

// КНОПКА ЗАКРЫТИЯ

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

{

this.Close();

}

// КНОПКА НАЗАД

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

{

loginWin logwin = new loginWin();

logwin.Show();

this.Hide();

}

}

}

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;

namespace loginkosto4

{

/// <summary>

/// Логика взаимодействия для winCaptcha.xaml

/// </summary>

public partial class winCaptcha : Window

{

string captcha = "";

public winCaptcha()

{

InitializeComponent();

}

void loadcap()

{

captcha = "";

string ABC = "1,2,3,4,5,6,7,8,9,0,Q,W,E,R,T,Y,U,I,O,P,A,S,D,F,G,H,J,K,L,Z,X,C,V,B,N,M," +

"q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z,x,c,v,b,n,m";

char[] a = { ',' };

string[] ar = ABC.Split(a);

string temp = " ";

Random rnd = new Random();

for (int i = 0; i < 4; i++)

{

temp = ar[rnd.Next(0, ar.Length)];

captcha += temp;

}

capbox.Text = captcha;

if (vvodBox == null)

{

MessageBox.Show("0!");

}

}

private void TextBox\_TextChanged(object sender, TextChangedEventArgs e)

{

}

private void RefreshBtn(object sender, RoutedEventArgs e)

{

loadcap();

}

private void EnterBtn(object sender, RoutedEventArgs e)

{

if (vvodBox.Text == "")

{

MessageBox.Show("Поле не может быть пустым");

}

else if (vvodBox.Text == captcha)

{

MessageBox.Show("Верно!");

loginWin w = new loginWin();

w.Show();

this.Hide();

}

else

{

winBlock block = new winBlock();

block.Show();

this.Hide();

}

}

private void vvodBox\_TextChanged(object sender, TextChangedEventArgs e)

{

}

private void message\_TextChanged(object sender, TextChangedEventArgs e)

{

}

}

}

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.Windows.Threading;

namespace loginkosto4

{

/// <summary>

/// Логика взаимодействия для winBlock.xaml

/// </summary>

public partial class winBlock : Window

{

public winBlock()

{

InitializeComponent();

secondmer();

}

// Секундомер сеанса

private void secondmer()

{

DispatcherTimer dt = new DispatcherTimer();

dt.Start();

dt.Interval = TimeSpan.FromSeconds(1);

dt.Tick += dtTicker;

}

private int increment = 10;

private void dtTicker(object sender, EventArgs e)

{

ltimer.Content = increment.ToString();

increment--;

if (increment == 0)

{

loginWin w = new loginWin();

w.Show();

this.Close();

}

}

}

}