Министерство образования Московской области

Государственное образовательное учреждение высшего образования Московской области

«Государственный гуманитарно-технологический университет»

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

**CASE**

«Разработка приложения, имитирующего работу торгового автомата»

**Выполнили:**

Шадрин И.О

Кутлаков В. А.

Медведева Н. В.

Нагорный Д. А.

Гусев М. А.

Середняков Е. А.

Кротков С. А.

студенты 4 курса

09.02.07 Информационные системы и

программирование

Ликино-Дулево

2022 год

Содержание

[1. Разработка технического задания 3](#_Toc96932163)

[1.1. Требования к функциональным характеристикам 4](#_Toc96932164)

[1.2. Требования к надежности и безопасности 5](#_Toc96932165)

[1.3. Требования к составу и параметрам технических средств 5](#_Toc96932166)

[1.4. Требования к информационной и программной совместимости 6](#_Toc96932167)

[2. Проектирование экранных форм 7](#_Toc96932168)

[3. Алгоритм работы приложения 9](#_Toc96932169)

[3.1. Блок-схема работы приложения 9](#_Toc96932170)

[4. Этап проектирования 10](#_Toc96932171)

[4.1. Диаграмма прецедентов 10](#_Toc96932172)

[5. Проектирование и разработка базы данных 11](#_Toc96932173)

[6. Руководство программиста 16](#_Toc96932174)

[7. Руководство пользователя 23](#_Toc96932175)

[8. Отладка и тестирование 29](#_Toc96932176)

[9. Текст программы 31](#_Toc96932177)

[Список литературы 36](#_Toc96932178)

# Разработка технического задания

Технический прогресс все глубже проникает в торговые процессы. Сегодня уже никого не удивить интернет-магазинами, продажей товаров по заказу на дому, по образцам и т.д. Товары продают в магазинах индивидуального обслуживания и самообслуживания, в киосках и других торговых точках (пунктах торговли). Все большую популярность приобретают пункты продажи, в которых торговля осуществляется посредством специального оборудования или приборов, называемых торговыми автоматами.

В торговом автомате могут продаваться различные напитки (безалкогольные, слабоалкогольные, пиво), штучные товары (как продовольственные, так и непродовольственные). Так как есть спрос на покупки из торговых автоматов, есть и предложение торговать таким образом.

Этот вид торговли широко распространен в экономически развитых странах благодаря достижениям науки и техники в области механики, гидравлики, пневматики, оптики. Но особенно бурное развитие эта форма продажи приобрела во второй половине XX в. под влиянием фундаментальных открытий в области электротехники и электроники. В наиболее развитых странах удельный вес товаров, реализуемых через торговые автоматы, составляет 10-12%, а по некоторым товарным группам - 30-40%.

С внедрением торговых автоматов повышается эффективность торговли, снижаются издержки обращения, повышается уровень механизации и автоматизации технологических процессов.

Современный технический уровень позволяет создать торговый автомат практически для любого товара, целесообразно применять торговые автоматы, прежде всего для продажи товаров массового спроса.

## Требования к функциональным характеристикам

Автоматизированная информационная система должная обеспечивать выполнение следующих функций:

* Вносить сумму, щелкая на кнопки с номиналом монет (1, 2, 5, 10). Если монета заблокирована, подсвечивать соответствующую кнопку и блокировать нажатие. Показывать внесенную сумму.
* Выбирать напиток, щелкнув на соответствующую картинку. При этом не позволять выбирать закончившиеся напитки или напитки, стоимость которых превышает внесенную сумму. После выбора напитка количество оставшихся напитков должно уменьшиться на единицу, количество монет в автомате увеличиться, оставшаяся сумма должна быть возвращена в виде сдачи. В качестве сдачи могут выдаваться заблокированные монеты.
* Администрировать ассортимент напитков: добавление, удаление напитков, изменение их количества, стоимости и изображения.
* Управлять монетами в автомате: изменение количества монет, блокирование приема тех или иных монет.
* Доступ в административный интерфейс по секретному ключу.
* Возможность импорта напитков.
* Возможность покупки нескольких напитков перед получением сдачи.
* Секретный ключ в базе должен хранится в зашифрованном виде.

**Входные данные:**

Монеты номиналом 1, 2, 5 и 10 руб., секретный код торгового автомата, напитки и информация о них.

**Выходные данные:**

Выходными данными являются отчёт по количеству проданных и оставшихся напитков и полученной прибыли, сдача в виде монет номиналом 1, 2, 5 и 10 руб.

## Требования к надежности и безопасности

Разрабатываемое программное обеспечение должно нормально функционировать при бесперебойной работе компьютера пользователя.

При возникновении ошибок или сбоев в работе компьютера восстановление нормальной работы программы должно производиться после полной перезагрузки операционной системы, запуска исполняемого файла. При условии, если пользователь до сбоя работы сохранил внесенные данные, в таком случае данные сохранятся в базе данных.

Для защиты информации на компьютере пользователя должны быть предусмотрены необходимые меры: пароль на вход в компьютер, антивирусные программы, отсутствие на компьютере подозрительных программ, полученных с неофициальных источников.

При передаче экземпляра ПО, использовать безопасные методы передачи информации.

## Требования к составу и параметрам технических средств

Минимальные системные требования для работы программного продукта должны быть следующими: частота процессора – 1 800 Ггц, объем оперативной памяти 4 Гб, объем свободного дискового пространства 500 Мб, разрешение монитора 1280х720. Для печати договора необходим принтер.

**Характеристика компьютера**

Процессор - Intel Core i5-4460

Частота процессора - 2 ГГц

Оперативная память - 8 Гб

Монитор - Samsung sa450, расширение 1920x1200

Клавиатура - Oklick 100M

Мышь - Oklick 415MW

## Требования к информационной и программной совместимости

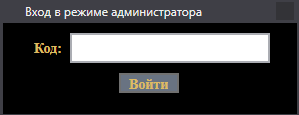
Для работы с базой данных и приложением, на рабочем компьютере должно быть установлено следующее программное обеспечение:

* СУБД – Microsoft SQL Server 2019.
* Утилита для подключения и управления БД - SQL Server Management Studio 18.
* IDE – Visual Studio 2019.

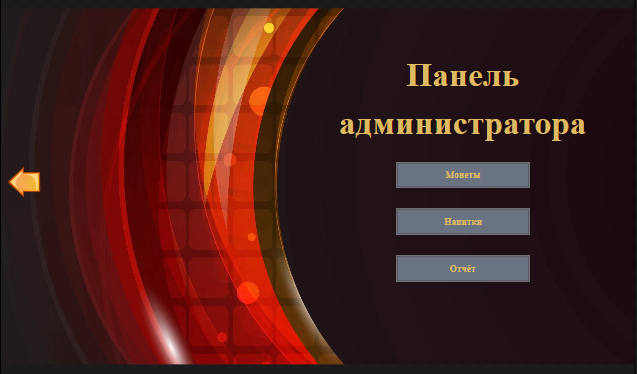
# Проектирование экранных форм



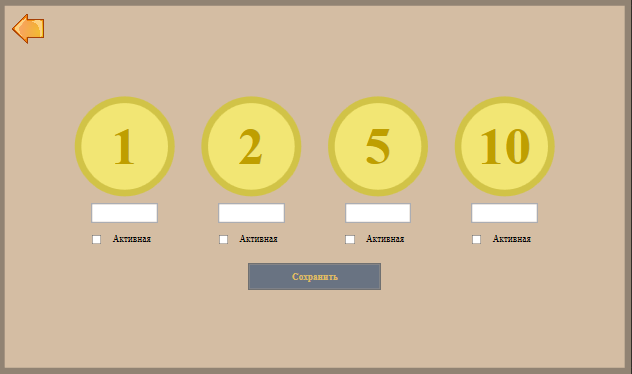
1. «Главная панель торгового автомата»



1. «Окно ввода секретного ключа администратора»



1. «Панель администратора»



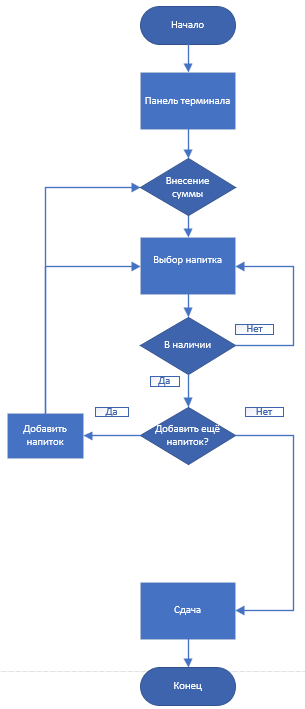
1. «Окно управления монетами»



1. «Окно добавления, удаления и редактирования напитков»

# Алгоритм работы приложения

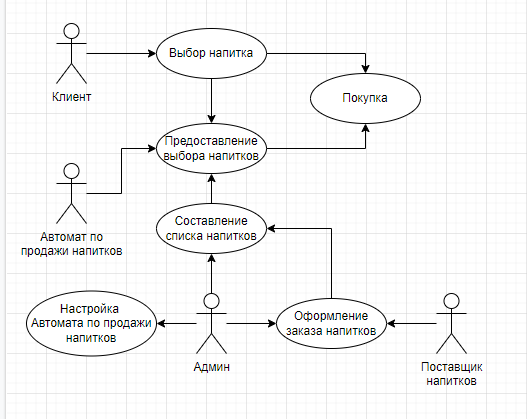
## Блок-схема работы приложения



1. «Алгоритм работы»

# Этап проектирования

## Диаграмма прецедентов

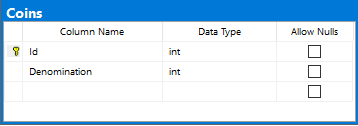


1. «Диаграмма прецедентов»

# Проектирование и разработка базы данных

**Создание таблиц**

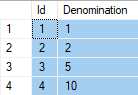
Таблица «Монеты»



1. «Таблица Coins»

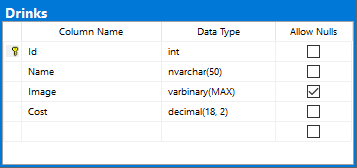
Id – код монеты (первичный ключ)

Denomination – деноминация монет



1. «Данные в таблице Coins»

Таблица «Напитки»



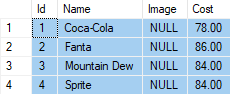
1. «Таблица Drinks»

Id – код напитка (первичный ключ)

Name – наименование напитка

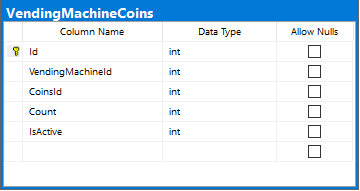
Image – изображение напитка

Cost – стоимость напитка



1. «Данные в таблице Drinks»

Таблица «Монеты в торговом автомате»



1. «Таблица VendingMachineCoins»

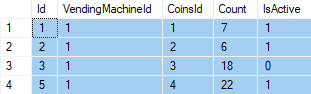
Id – код монеты в торговом автомате (первичный ключ)

VendingMachineId – код торгового автомата (вторичный ключ)

CoinsId – код монеты (вторичный ключ)

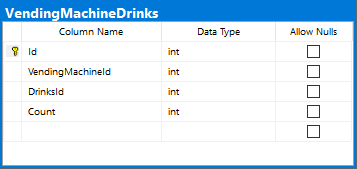
Count – количество монет в торговом автомате

IsActive – состояние монеты (активное или пассивное)



1. «Данные в таблице VendingMachineCoins»

Таблица «Напитки в торговом автомате»



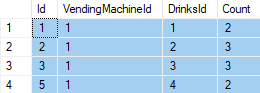
1. «Таблица VendingMachineDrinks»

Id – код напитка в торговом автомате (первичный ключ)

VendingMachineId – код торгового автомата (вторичный ключ)

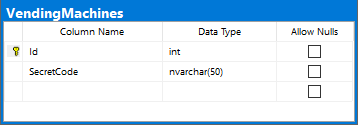
DrinksId – код напитка (вторичный ключ)

Count – количество напитков в торговом автомате



1. «Данные в таблице VendingMachineDrinks»

Таблица «Торговые автоматы»



1. «Таблица VendingMachines»

Id – код торгового автомата (первичный ключ)

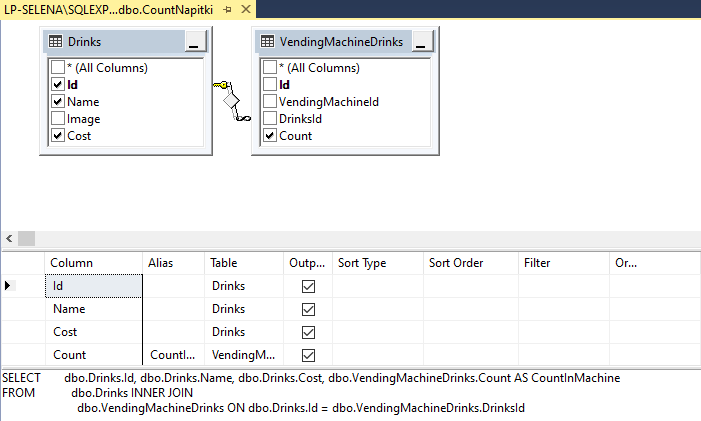
SecretCode – секретный код доступа для администратора



1. «Данные в таблице VendingMachines»

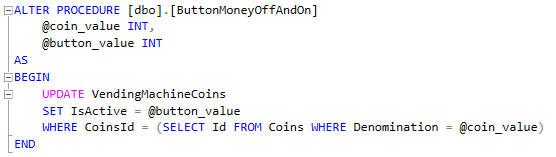
**Реализация запросов**

Представление для создания и вывода отчёта о работе автомата



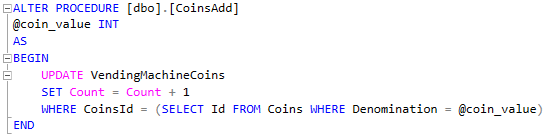
1. «Представление CountNapitki»

Хранимая процедура для изменения состояния активности монеты



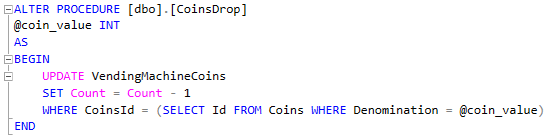
1. «Хранимая процедура ButtonMoneyOffAndOn»

Хранимая процедура для увеличения количества монет указанной деноминации в автомате на единицу.



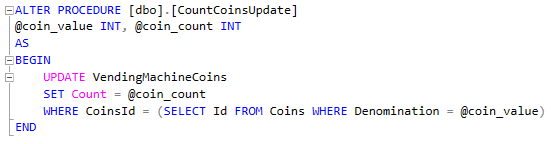
1. «Хранимая процедура CoinsAdd»

Хранимая процедура для уменьшения количества монет указанной деноминации в автомате на единицу.



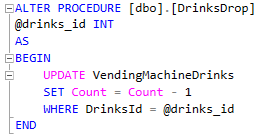
1. «Хранимая процедура CoinsDrop»

Хранимая процедура для изменения текущего количества монет на новое указанной деноминации в автомате.

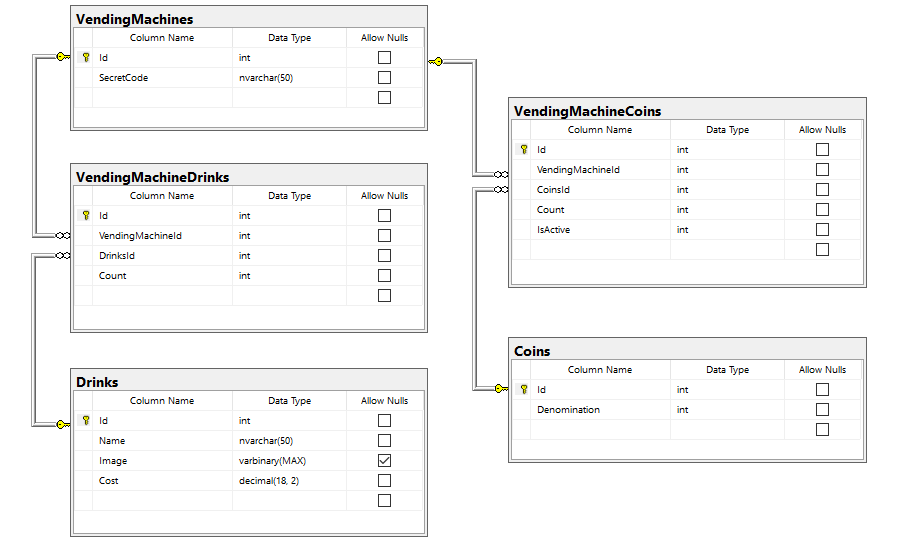


1. «Хранимая процедура CountCoinsUpdate»

Хранимая процедура для уменьшения количества напитков в автомате на единицу по указанному ключу.



1. «Хранимая процедура DrinksDrop»

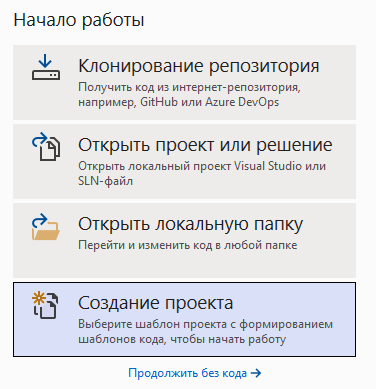


1. «Модель базы данных»

# Руководство программиста

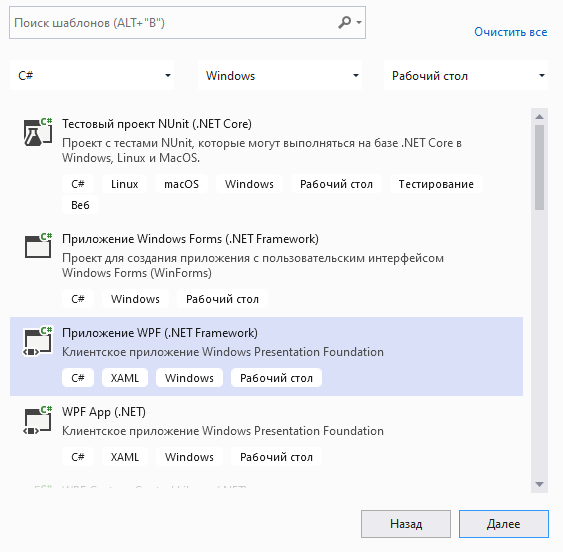
Программа разработана на Visual Studio 2019

Создание проекта



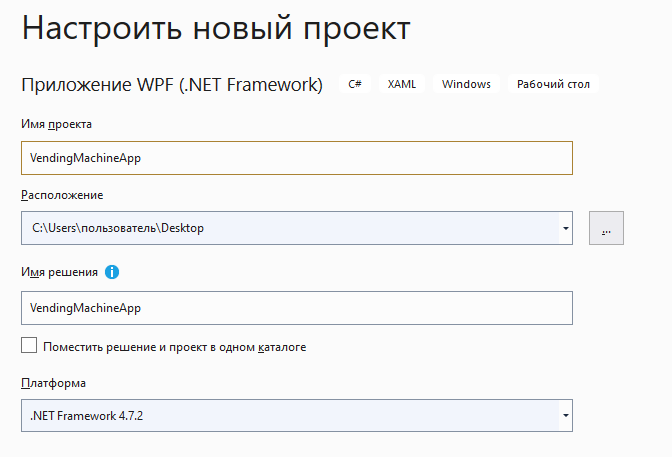
1. «Окно начала работы»

Выбор типа проекта



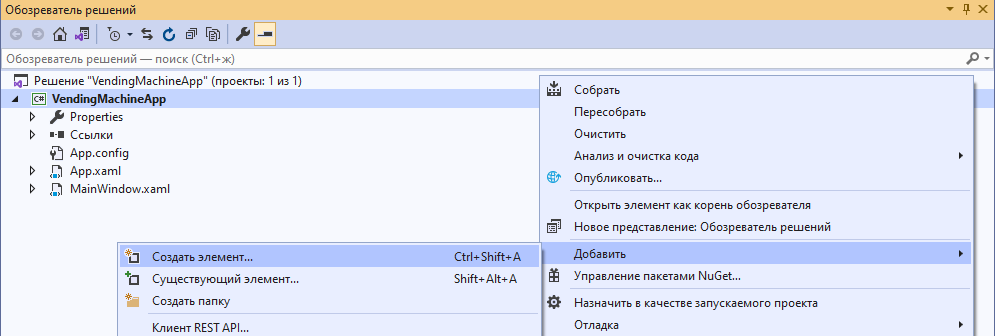
1. «Тип проекта Приложение WPF»

Настройка расположения и имени проекта



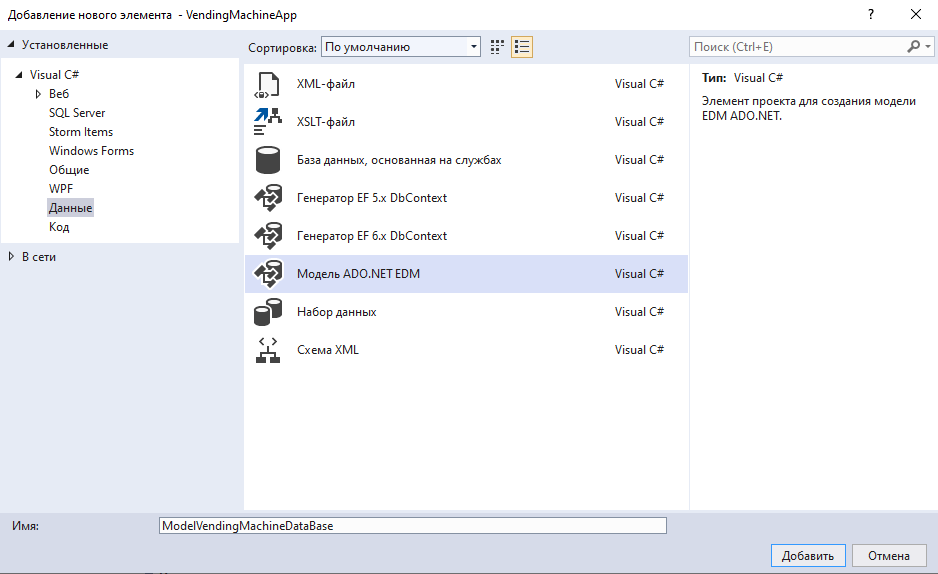
1. «Настройка нового проекта»

Подключение к базе данных ADO.NET



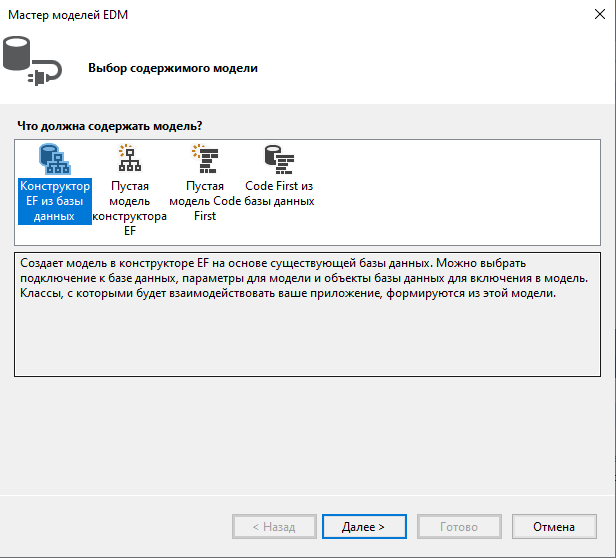
1. «Добавление нового элемента в проект»

Выбрать в левом столбце пункт «Данные», нажать на «Модель ADO.NET» и указать имя новой модели



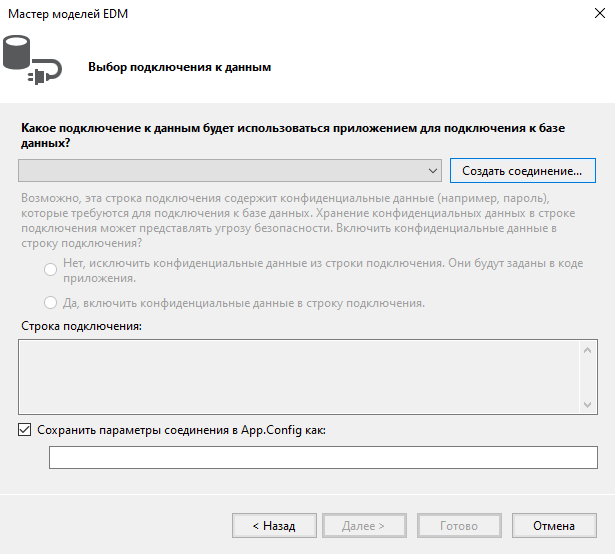
1. «Добавление элемента ADO.NET»

Подключение базы данных к приложению



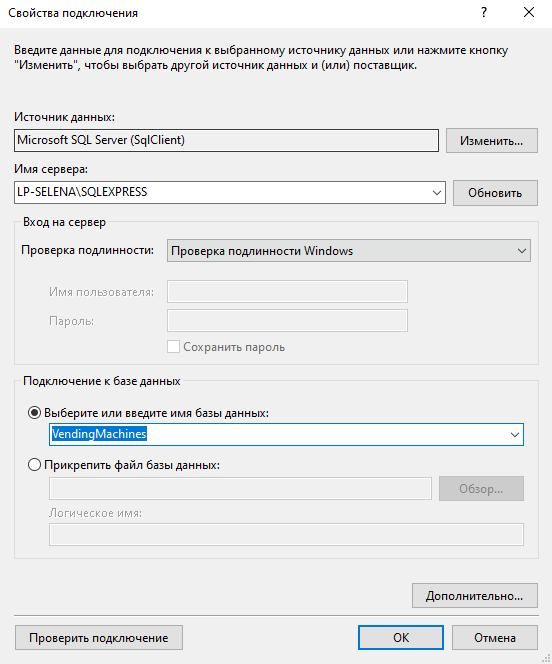
1. «Выбор содержимого модели»

Нажать на «Создать соединение» или выбрать существующее



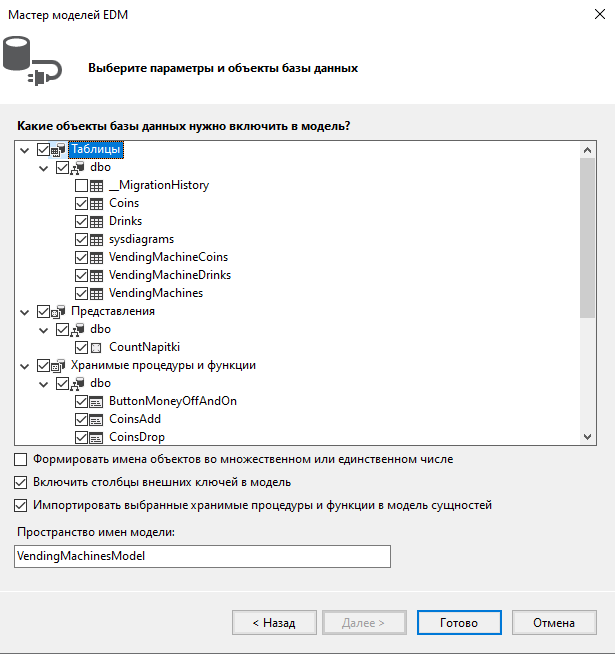
1. «Выбор подключения к данным»

Ввести «Имя сервера» и выбрать нужную базу данных



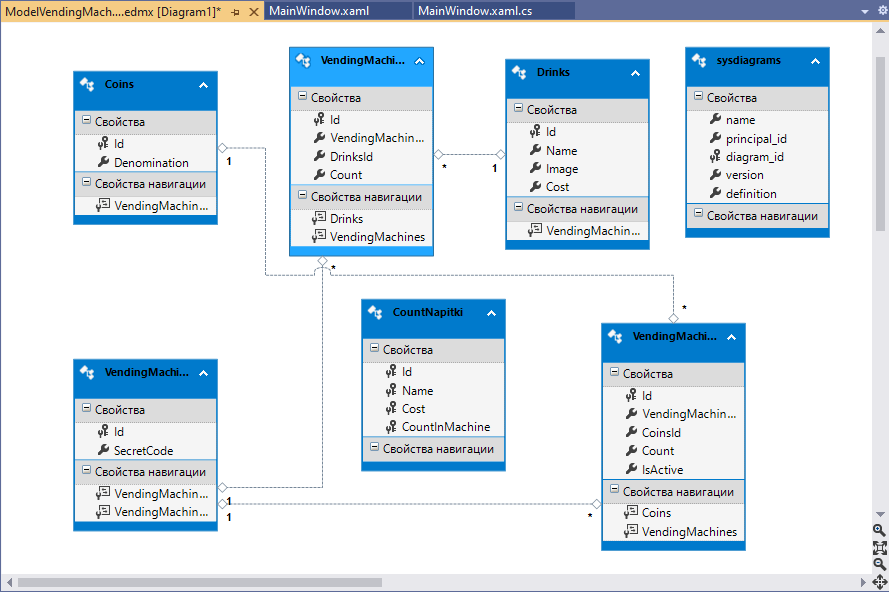
1. «Свойства подключения базы данных»

Выбрать нужные объекты и нажать «Готово»



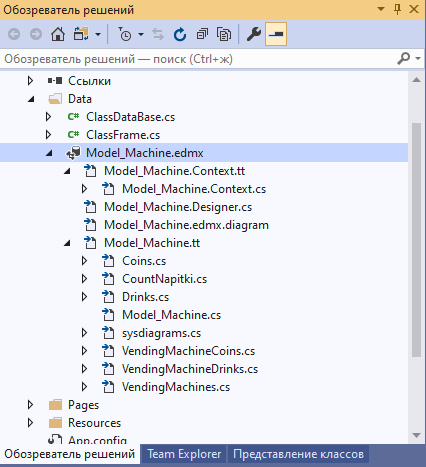
1. «Выбор объектов для покдлючения из базы данных»

Полученная модель базы данных



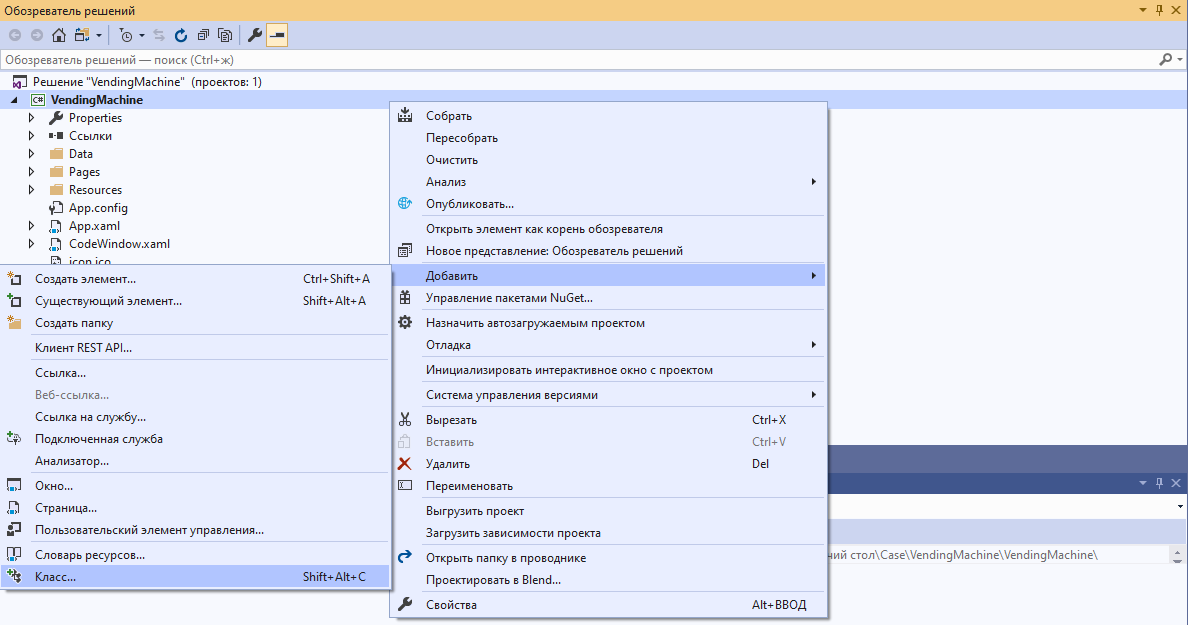
1. «Модель базы данных ModelVendingMachineDataBase»

Модель базы данных в Обозревателе решений



1. «Модель базы данных в Обозревателе решений»

Создание и добавление класса в приложение



1. «Создание класса»

Класс ClassDataBase – помогает взаимодействовать с базой данных

public class ClassDataBase

{

public static Vending\_Machines\_Entities dbObject;

}

Класс ClassFrame – помогает при переходе между страницами

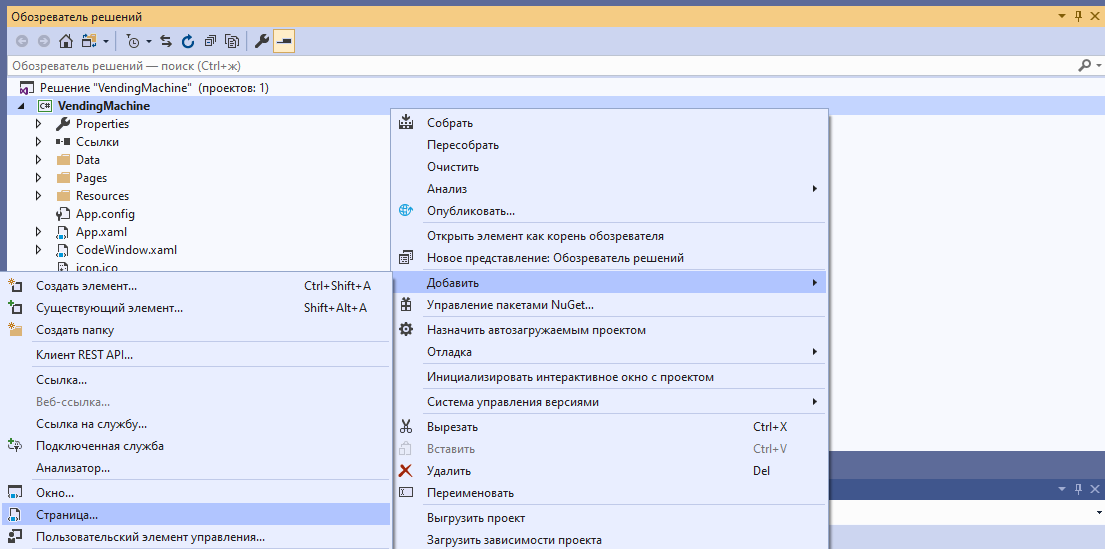
public class ClassFrame

{

public static Frame frameObj;

}

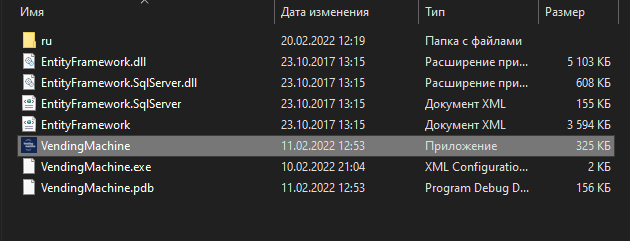
Создание и добавление страницы в приложении



1. «Создание страницы»

# Руководство пользователя

Запустить файл VendingMachine.exe и дождаться запуска программы.



«Файл VendingMachine.exe»

После запуска программы, откроется главная панель торгового автомата, где по нажатию кнопок монет номиналом 1, 2, 5 и 10 рублей можно внести нужную сумму в торговый автомат и нажатием по изображению напитка совершить покупку выбранного напитка.



1. «Главная панель торгового автомата»

В случае если внесённых средств недостаточно для покупки выбранного напитка, автомат выведет на экран соответствующее сообщение.



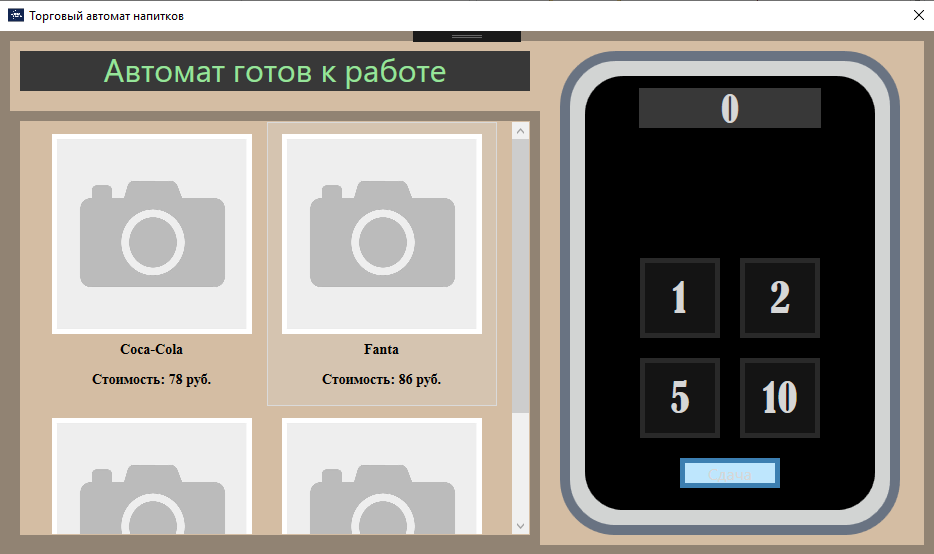
1. «Покупка напитка за 86 руб. при балансе 80 руб.»

Можно доложить в торговый автомат недостающую сумму и повторить попытку покупки, после чего покупка будет совершена, а сумма напитка будет вычтена из внесённой в торговый автомат суммы. Таким же образом можно осуществить покупку сразу нескольких напитков.



1. «Покупка напитка за 86 руб. после внесения ещё 10 руб.»

Если после совершения покупки осталась в автомате некая сумма, то при нажатии кнопки сдачи, оставшиеся деньги будут возвращены покупателю. Также можно потребовать сдачи после внесения суммы в автомат без покупки напитка, в случае, если это необходимо.



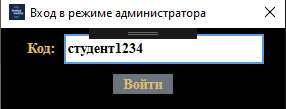
1. «Возврат оставшихся после покупки 4 руб.»

Для вызова поля ввода секретного ключа необходимо 10 раз нажать кнопку "Сдача".

При этом количество внесённых монет на текущий момент должно быть равно нулю.

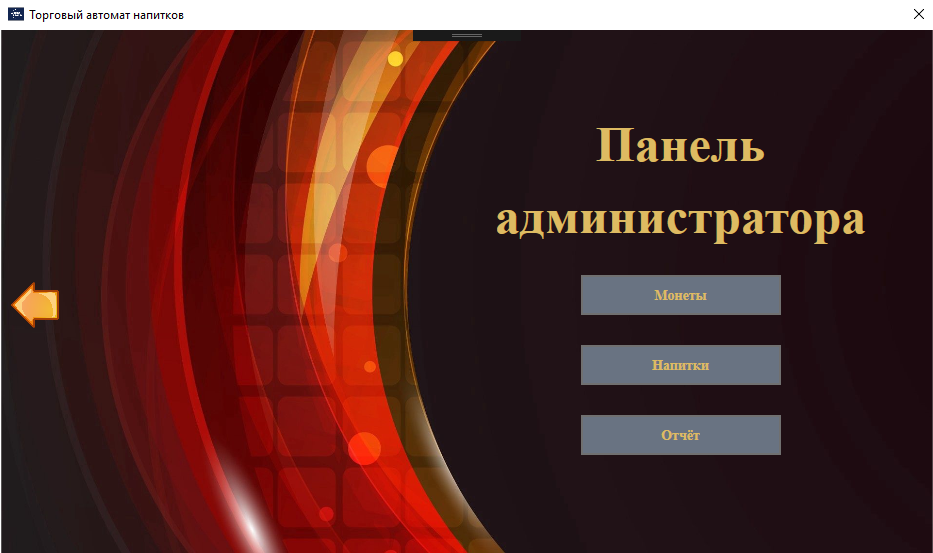
В противном случае, для вызова поля ввода секретного ключа необходимо 11 раз нажать кнопку "Сдача".

Секретный ключ для входа администратора - студент1234



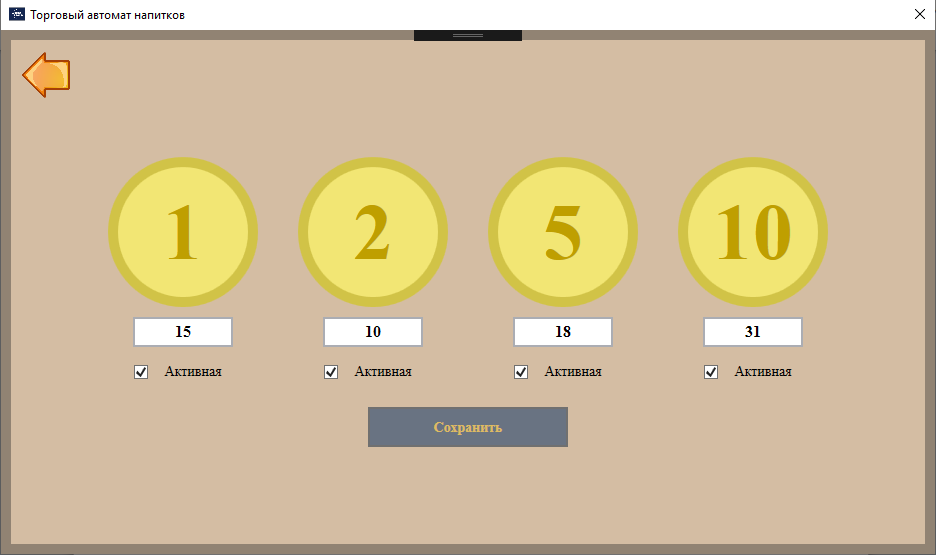
1. «Ввод кода для входа в панель администратора»

После ввода кода и нажатия кнопки «Войти» в случае правильного ввода пароля откроется панель администратора, на которой находятся кнопки для перехода на все страницы управления торговым автоматом.



1. «Панель администратора»

На странице «Монеты» можно просмотреть или изменить количество монет разного номинала в автомате, а также отключить кнопки на торговом автомате соответствующего номинала.



1. «Окно управления монетами в автомате»

При отключении всех кнопок на странице управления монетами, при переходе на главную панель торгового автомата, напитки и кнопки будут неактивны, а автомат будет отображать сообщение о том, что он сейчас не работает.



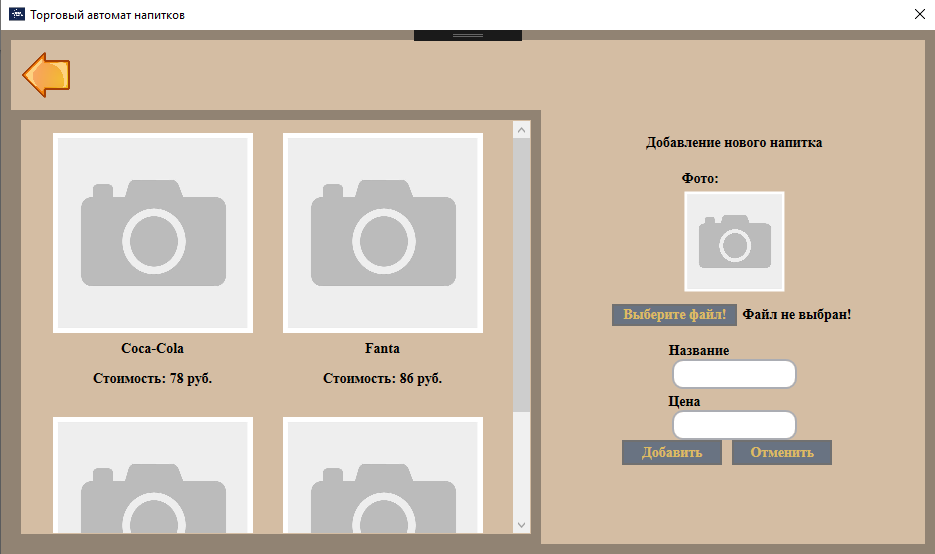
1. «Выключенная главная панель торгового автомата»

Если монет номиналом 1 руб. меньше 10 шт., 2 руб. меньше 5 шт., 5 руб. меньше 2 шт. или 10 руб. меньше 1 шт., то на главной панели торгового автомата появится сообщение, что автомат сдачи не выдаёт.



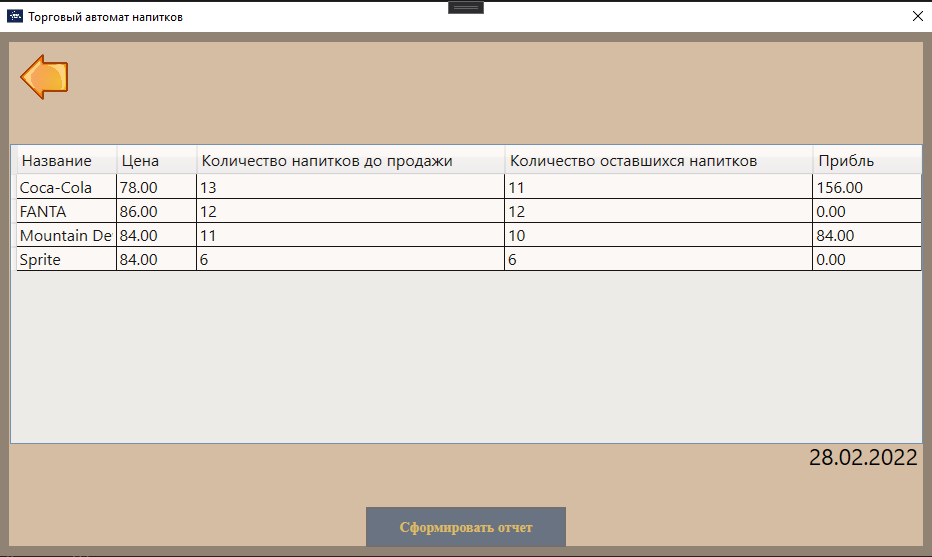
1. «Сообщение о сдачи при наличии монет 1 руб. в количестве 8 шт.»

На странице «Напитки» можно произвести добавление нового напитка или удаление и изменение существующих напитков.



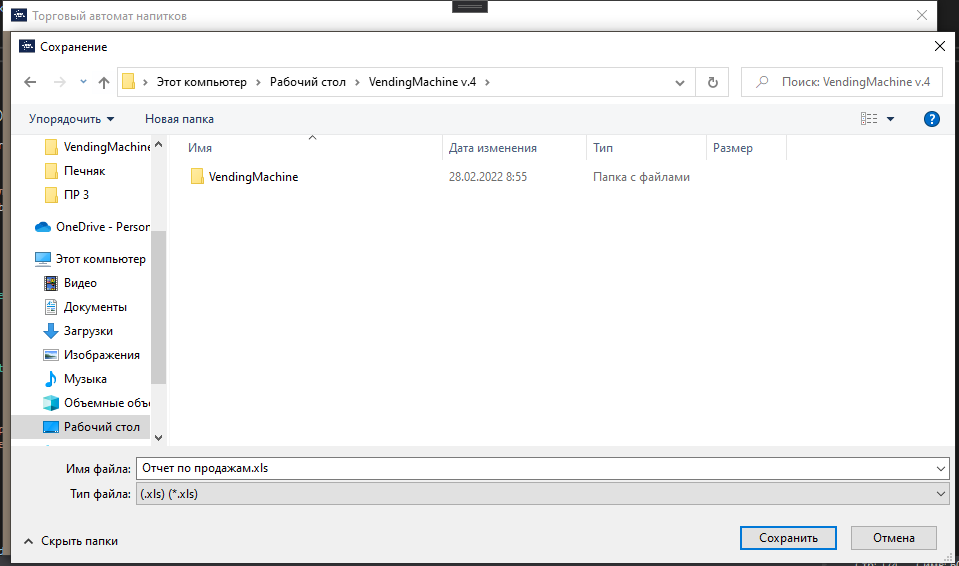
1. «Страница напитков»

На странице «Отчёт» можно просмотреть отчёт и вывести его в Excel. В отчёте отображается название напитков, количество напитков на текущий момент, количество монет на момент запуска приложения и заработанная по каждому напитку прибыль.



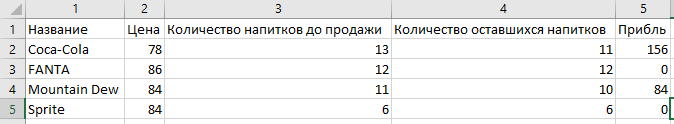
1. «Отчёт о покупке напитков в автомате»

Диалоговое окно для сохранения отчета



1. «Диалоговое окно»

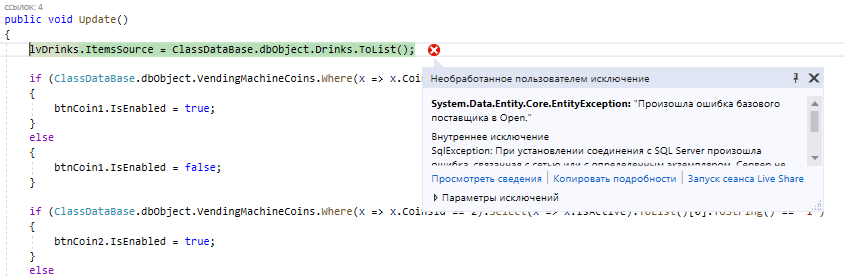
Итоговый отчет по продажам в Excel



1. «Отчет в Excel»

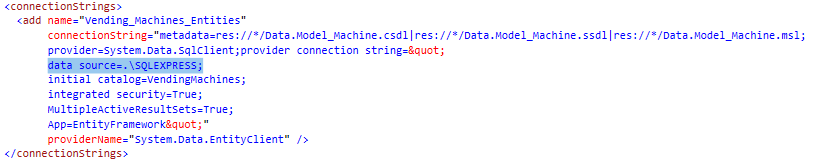
# Отладка и тестирование

Ошибка при запуске приложения из-за невозможности подключения к базе данных, связанная с переносом приложения с одного компьютера на другой и изменением имени сервера в Microsoft SQL Server.



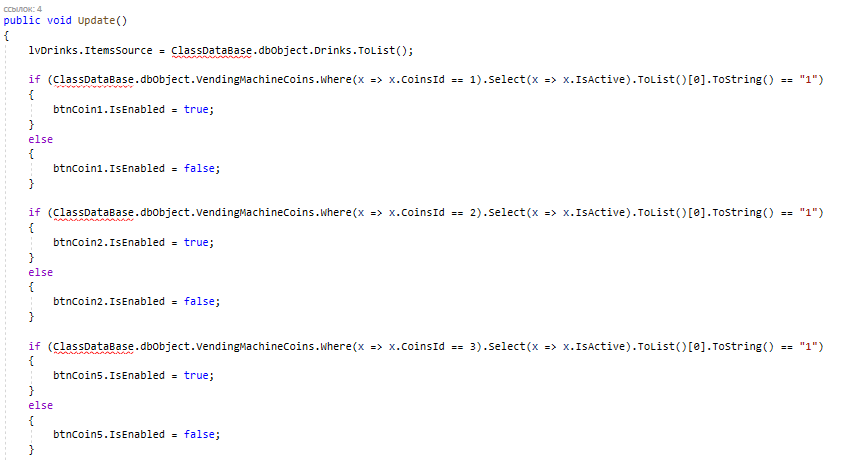
1. «Ошибка при подключении к базе данных»

Исправлено



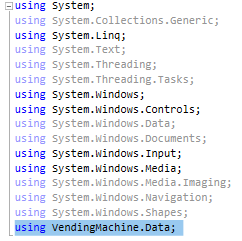
1. «Изменение имени сервера на универсальное»

Ошибка при попытке инициализации объектов из-за отсутствия ссылки на местонахождение объектов и подключения необходимой части проекта.



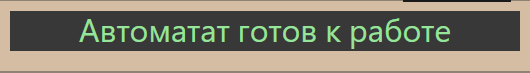
1. «Ошибка видимости объектов в проекте»

Исправлено



1. «Подключение ссылки на папку Data в проекте»

Ошибка в написании слова «Автомат» на главной странице автомата.



1. «Грамматическая ошибка»

Исправлено



1. «Исправление слова “Автоматат” на “Автомат”»

# Текст программы

Функция обновления актуальной информации на главной панели торгового автомата после совершения торговых манипуляций.

public void Update()

{

lvDrinks.ItemsSource = ClassDataBase.dbObject.Drinks.ToList();

if (ClassDataBase.dbObject.VendingMachineCoins.Where(x => x.CoinsId == 1).Select(x => x.IsActive).ToList()[0].ToString() == "1")

{

btnCoin1.IsEnabled = true;

}

else

{

btnCoin1.IsEnabled = false;

}

if (ClassDataBase.dbObject.VendingMachineCoins.Where(x => x.CoinsId == 2).Select(x => x.IsActive).ToList()[0].ToString() == "1")

{

btnCoin2.IsEnabled = true;

}

else

{

btnCoin2.IsEnabled = false;

}

if (ClassDataBase.dbObject.VendingMachineCoins.Where(x => x.CoinsId == 3).Select(x => x.IsActive).ToList()[0].ToString() == "1")

{

btnCoin5.IsEnabled = true;

}

else

{

btnCoin5.IsEnabled = false;

}

if (ClassDataBase.dbObject.VendingMachineCoins.Where(x => x.CoinsId == 4).Select(x => x.IsActive).ToList()[0].ToString() == "1")

{

btnCoin10.IsEnabled = true;

}

else

{

btnCoin10.IsEnabled = false;

}

if (int.Parse(ClassDataBase.dbObject.VendingMachineCoins.Where(x => x.CoinsId == 1).Select(x => x.Count).ToList()[0].ToString()) < 10 ||

int.Parse(ClassDataBase.dbObject.VendingMachineCoins.Where(x => x.CoinsId == 2).Select(x => x.Count).ToList()[0].ToString()) < 5 ||

int.Parse(ClassDataBase.dbObject.VendingMachineCoins.Where(x => x.CoinsId == 3).Select(x => x.Count).ToList()[0].ToString()) < 2 ||

int.Parse(ClassDataBase.dbObject.VendingMachineCoins.Where(x => x.CoinsId == 1).Select(x => x.Count).ToList()[0].ToString()) < 1)

{

tblShopper.Foreground = new SolidColorBrush(Color.FromRgb(0xfa, 0x4b, 0x4b));

tblShopper.Text = "Автомат сдачи не выдаёт";

machineActive = false;

}

else

{

if (btnCoin1.IsEnabled == false && btnCoin2.IsEnabled == false && btnCoin5.IsEnabled == false && btnCoin10.IsEnabled == false)

{

lvDrinks.IsEnabled = false;

tblShopper.Foreground = new SolidColorBrush(Color.FromRgb(0xfa, 0x4b, 0x4b));

tblShopper.Text = "! Автомат не работает !";

}

else

{

lvDrinks.IsEnabled = true;

tblShopper.Foreground = new SolidColorBrush(Color.FromRgb(0x97, 0xe5, 0x9a));

tblShopper.Text = "Автомат готов к работе";

btnChange.IsEnabled = true;

machineActive = true;

}

}

}

Функционал кнопки «Сдача» на главной панели торгового автомата.

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

{

if (int.Parse(tblSumma.Text) == 0)

{

clickCount += 1;

}

else

{

clickCount = 0;

if (machineActive == false)

{

MessageBox.Show("Автомат сдачи не выдаёт", "Предупреждение", MessageBoxButton.OK, MessageBoxImage.Warning);

tblSumma.Text = "0";

return;

}

}

if (int.Parse(tblSumma.Text) == 0 && clickCount == 10)

{

CodeWindow window = new CodeWindow();

clickCount = 0;

window.ShowDialog();

}

else

{

if (int.Parse(tblSumma.Text) > 0)

{

while (int.Parse(tblSumma.Text) / 10 >= 1)

{

tblSumma.Text = (int.Parse(tblSumma.Text) - 10).ToString();

ClassDataBase.dbObject.CoinsDrop(10);

}

while (int.Parse(tblSumma.Text) / 5 >= 1)

{

tblSumma.Text = (int.Parse(tblSumma.Text) - 5).ToString();

ClassDataBase.dbObject.CoinsDrop(5);

}

while (int.Parse(tblSumma.Text) / 2 >= 1)

{

tblSumma.Text = (int.Parse(tblSumma.Text) - 2).ToString();

ClassDataBase.dbObject.CoinsDrop(2);

}

while (int.Parse(tblSumma.Text) > 0)

{

tblSumma.Text = (int.Parse(tblSumma.Text) - 1).ToString();

ClassDataBase.dbObject.CoinsDrop(1);

}

Update();

}

}

}

Событие при нажатии на напиток в списке при покупке

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

{

Drinks drinks = lvDrinks.SelectedItem as Drinks;

if (drinks.Cost > int.Parse(tblSumma.Text))

{

tblShopper.Foreground = new SolidColorBrush(Color.FromRgb(0xfa, 0x4b, 0x4b));

tblShopper.Text = "Недостаточно средств";

}

else

{

ClassDataBase.dbObject.DrinksDrop(drinks.Id);

tblSumma.Text = (int.Parse(tblSumma.Text) - Convert.ToInt32(drinks.Cost)).ToString();

tblShopper.Foreground = new SolidColorBrush(Color.FromRgb(0x97, 0xe5, 0x9a));

Update();

}

}

Код проверки и шифрования секретного ключа администратора.

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

{

ushort secretKey = 0x0040;

string code = tbCode.Text;

code = EncodeDecrypt(code, secretKey);

var admin = ClassDataBase.dbObject.VendingMachines.FirstOrDefault(

x => x.SecretCode == code);

if (admin == null)

{

MessageBox.Show("Код введён неверно",

"Ошибка",

MessageBoxButton.OK,

MessageBoxImage.Error);

}

else

{

ClassFrame.frameObj.Navigate(new PageAdmin());

Close();

}

}

public static string EncodeDecrypt(string str, ushort secretKey)

{

var ch = str.ToArray();

string newStr = "";

foreach (var c in ch)

newStr += TopSecret(c, secretKey);

return newStr;

}

public static char TopSecret(char character, ushort secretKey)

{

character = (char)(character ^ secretKey);

return character;

}

Код импорта данных из отчёта в приложении в Excel.

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

{

dgOtchet.SelectAllCells();

dgOtchet.ClipboardCopyMode = DataGridClipboardCopyMode.IncludeHeader;

ApplicationCommands.Copy.Execute(null, dgOtchet);

String resultat = (string)Clipboard.GetData(DataFormats.CommaSeparatedValue);

String result = (string)Clipboard.GetData(DataFormats.Text);

dgOtchet.UnselectAllCells();

Microsoft.Win32.SaveFileDialog dlg = new Microsoft.Win32.SaveFileDialog();

dlg.FileName = "Export";

dlg.DefaultExt = ".text";

dlg.Filter = "(.xls)|\*.xls";

Nullable<bool> result1 = dlg.ShowDialog();

if (result1 == true)

{

string filename = dlg.FileName;

System.IO.StreamWriter file = new System.IO.StreamWriter(filename, false, Encoding.Default);

file.WriteLine(result);

file.Close();

MessageBox.Show("Экспорт данных успешно завершен");}}

Код удаления напитка из торгового автомата.

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

{

Drinks drink = lvDrinks.SelectedItem as Drinks;

if (drink == null) MessageBox.Show("Запись не выбрана!", "Ошибка удаления", MessageBoxButton.OK, MessageBoxImage.Error);

else if (MessageBox.Show("Вы точно хотите удалить запись?\n" + drink.Name,

"Удаление записи", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

ClassDataBase.dbObject.Drinks.Remove(drink);

ClassDataBase.dbObject.SaveChanges();

MessageBox.Show("Запись удалена", "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information);

lvDrinks.ItemsSource = ClassDataBase.dbObject.Drinks.ToList();}}

Код сохранения внесённых в приложении изменений в базе данных.

var entObj = ClassDataBase.dbObject.Drinks.FirstOrDefault(x => x.Name.ToLower() == txbName.Text.ToLower());

if (txbName.Text.Length == 0 && txbPriceAdd.Text.Length == 0 && txbCountAdd.Text.Length == 0)

MessageBox.Show("Поля не могут быть пустыми!",

"Критическая ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

else if (txbName.Text.Length == 0)

{

MessageBox.Show("Поле: 'Название' не может быть пустым!",

"Критическая ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

txbName.Focus();

}

else if (txbPriceAdd.Text.Length == 0)

{

MessageBox.Show("Поле: 'Цена' не может быть пустым!",

"Критическая ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

txbPriceAdd.Focus();

}

else if (txbCountAdd.Text.Length == 0)

{

MessageBox.Show("Поле: 'Количество' не может быть пустым!",

"Критическая ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

txbCountAdd.Focus();

}

else if (entObj != null)

MessageBox.Show("Напиток с таким именем уже есть!",

"Критическая ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

else

{

if (imagePath == null)

{

Drinks drink = new Drinks()

{

Name = txbName.Text,

Image = null,

Cost = Convert.ToDecimal(txbPriceAdd.Text)

};

ClassDataBase.dbObject.Drinks.Add(drink);

ClassDataBase.dbObject.SaveChanges();

VendingMachineDrinks countDrinks = new VendingMachineDrinks()

{

VendingMachineId = 1,

DrinksId = drink.Id,

Count = int.Parse(txbCountAdd.Text)

};

ClassDataBase.dbObject.VendingMachineDrinks.Add(countDrinks);

ClassDataBase.dbObject.SaveChanges();

MessageBox.Show("Напиток успешно добавлен", "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information);

lvDrinks.ItemsSource = ClassDataBase.dbObject.Drinks.ToList();

}

else

{

Drinks drink = new Drinks()

{

Name = txbName.Text,

Image = File.ReadAllBytes(imagePath),

Cost = Convert.ToDecimal(txbPriceAdd.Text)

};

ClassDataBase.dbObject.Drinks.Add(drink);

ClassDataBase.dbObject.SaveChanges();

VendingMachineDrinks countDrinks = new VendingMachineDrinks()

{

VendingMachineId = 1,

DrinksId = drink.Id,

Count = int.Parse(txbCountAdd.Text)

};

ClassDataBase.dbObject.VendingMachineDrinks.Add(countDrinks);

ClassDataBase.dbObject.SaveChanges();

MessageBox.Show("Напиток успешно добавлен", "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information);

lvDrinks.ItemsSource = ClassDataBase.dbObject.Drinks.ToList();}}

# Список литературы

1. Илюшечкин, В. М. Основы использования и проектирования баз данных: учебник для среднего профессионального образования /В. М. Илюшечкин. — испр. и доп. — Москва: Издательство Юрайт, 2019. — 213 с. — (Профессиональное образование). — ISBN 978-5-534-01283-5. — Текст: электронный //ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/437670>
2. Нестеров, С. А. Базы данных: учебник и практикум для среднего профессионального образования / С. А. Нестеров. — Москва: Издательство Юрайт, 2020. — 230 с. — (Профессиональное образование). — ISBN 978-5-534-11629-8. — URL: <https://urait.ru/bcode/457142>
3. Федорова Г.Н. Основы проектирования баз данных: учебник для студентов СПО /Г.Н. Федорова. - 2-е изд., стер. – Москва: Академия, 2018. – 219с.
4. Федорова Г.Н. Разработка, администрирование и защита баз данных.: учебник для среднего профессионального образования / Федорова Г.Н — Москва: Издательство Академия, 2020. — 288 с. — (Профессиональное образование).
5. Маркин, А. В. Программирование на SQL: учебное пособие для среднего профессионального образования / А. В. Маркин. — Москва: Издательство Юрайт, 2020. — 435 с. — (Профессиональное образование). — ISBN 978-5-534-11093-7. — Текст: электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/456926>
6. Стружкин, Н. П. Базы данных: проектирование: учебник для среднего профессионального образования / Н. П. Стружкин, В. В. Годин. — Москва: Издательство Юрайт, 2020. — 477 с. — (Профессиональное образование). — ISBN 978-5-534-11635-9. — URL: <https://urait.ru/bcode/457135>
7. Гордеев, С. И. Организация баз данных в 2 ч. Часть 1: учебник для среднего профессионального образования / С. И. Гордеев, В. Н. Волошин
8. а. — 2-е изд., испр. и доп. — Москва: Издательство Юрайт, 2020. — 310 с. — (Профессиональное образование). — ISBN 978-5-534-11626-7. — URL: <https://urait.ru/bcode/457145>
9. Гордеев, С. И. Организация баз данных в 2 ч. Часть 2: учебник для среднего профессионального образования / С. И. Гордеев, В. Н. Волошина. — 2-е изд., испр. и доп. — Москва: Издательство Юрайт, 2020. — 513 с. — (Профессиональное образование). — ISBN 978-5-534-11625-0. — URL: <https://urait.ru/bcode/457146>
10. Стасышин, В. М. Базы данных: технологии доступа: учебное пособие для среднего профессионального образования / В. М. Стасышин, Т. Л. Стасышина. — 2-е изд., испр. и доп. — Москва: Издательство Юрайт, 2020. — 164 с. — (Профессиональное образование). — ISBN 978-5-534-09888-4. — URL: <https://urait.ru/bcode/45586>