Оглавление

[Введение 2](#_Toc21306681)

[Основная часть 3](#_Toc21306682)

[1 Теоретическое введение 3](#_Toc21306683)

[1.1 Объектно-ориентированное программирование. 3](#_Toc21306684)

[1.2 Описание CLR 3](#_Toc21306685)

[1.3 Методы создания интерфейса 4](#_Toc21306686)

[2 Проектная часть 5](#_Toc21306687)

[2.1 Постановка задачи 5](#_Toc21306688)

[2.2 Проектное решение 5](#_Toc21306689)

[3 Экспериментальная часть 14](#_Toc21306690)

[3.1 описание подхода к тестированию разработанной программы 14](#_Toc21306691)

[3.2 Тестирование 14](#_Toc21306692)

[3.3 Инструкция пользователя 19](#_Toc21306693)

[Заключение 22](#_Toc21306694)

[Список используемых источников 23](#_Toc21306695)

[ПРИЛОЖЕНИЕ 24](#_Toc21306696)

Введение

Целью курсового проекта является проектирование и реализация программной системы моделирования движения рыб в аквариуме с использованием объектного подхода. Предметная область представляет собой моделирование движения в аквариуме двух вида рыб: щук и карпов.

Основные задачи, требующие выполнения для достижения цели:

1) анализ предметной области;

2) формирование функциональных требований к системе;

3) проектирование компонентов системы;

4) реализация компонентов системы;

5) тестирование системы;

6) подготовка документации.

Основная часть

1 Теоретическое введение

1.1 Объектно-ориентированное программирование.

У каждого программирования есть своя концепция. Объектно-ориентированное, которое основано на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного класса. Также программирование имеет несколько основных понятий таких как:

* Абстракция – набор важных характеристик для объекта.
* Инкапсуляция – свойство, объединяющее данные и методы в классе, и позволяет скрыть реализацию от пользователя.
* Полиморфизм – одна функция может иметь множество различных действий, которые зависят от типа объекта.
* Наследование – свойство, которое позволяет описать новый класс на основе существующего, и у которого частично или полностью будет такая же функциональность.

1.2 Описание CLR

Общеязыковая среда выполнения (Common Language Runtime, CLR) – это среда выполнения, которая подходит для разных языков программирования. Основная возможность CLR (управление памятью, загрузка сборок, безопасность, обработка исключений, синхронизация) доступны в любых языках программирования, использующих эту среду. Например, при обработке ошибок среда выполнения опирается на исключения, а значит, во всех языках программирования, использующих эту среду выполнения, сообщения об ошибках передаются при помощи механизма исключений. Или, например, среда выполнения позволяет создавать программные потоки, а значит, во всех языках программирования, использующих эту среду, тоже могут создаваться потоки.

Фактически во время выполнения программы в среде CLR неизвестно, на каком языке программирования разработчик написал исходный код. А это значит, что можно выбрать любой язык программирования, который позволяет проще всего решить конкретную задачу. Разрабатывать программное обеспечение можно на любом языке программирования, если только используемый компилятор этого языка поддерживает CLR.

1.3 Методы создания интерфейса

Интерфейс – это важная составляющая любой программной системы. Для того чтобы пользователю было удобно пользоваться программой мы создаем форму, на которой создаем кнопки, поля для ввода, название полей и т. д. Для того, чтобы создать пользовательский интерфейс следует придерживаться некоторых правил, таких как:

* Должен быть удобен для чтения пользователя.

Также существуют различные виды пользовательского интерфейса такие как:

* Визуальный
* Текстовый
* Графический
* Оконный
* WIMP
* Web-ориентированный
* Индуктивный
* Масштабируемый
* Тактильный
* Жестовый
* Голосовой
* Материальный (осязательный)

Таким образом, разработчик, опираясь на условие задачи разрабатывает наиболее удобный и правильный интерфейс.

2 Проектная часть

2.1 Постановка задачи

Цель данной курсовой работы состоит в том, чтобы разработать программное обеспечение системы учёта товаров в магазине с использованием объектного подхода.

Для этого необходимо создать три класса:

* Класс Магазин
* Класс Продукт
* Класс Поставщик
* Класс Поступление Товара
* Класс Продажи Товара

Для удобства предоставить методы:

* Добавления
* Удаления
* Подсчета остававшихся и проданных товаров

2.2 Проектное решение

Система учета товаров будет разработана как настольное приложение. Данные будут храниться в классе-массиве и файле в формате JSON.

Основным инструментом реализации продукта является язык программирования С++ и CLR (Common language runtime).

2.2.1 Анализ исходных данных, постановки задачи, исследование предложенной вариантом задачи

Проанализировав работу розничного магазина были обнаружены следующие исходные данные: товары, поставщики, документы, информация о поступившем и проданном товаре. Следовательно нужно разработать пять классов для обработки и хранении данных.

2.2.2 Список понятий, претендующих на класс

В качестве классов следует сделать следующие понятия:

* Магазин
* Товар
* Поставщик
* Документы
* Приход
* Расход

2.2.3 Диаграмма объектов с указанием воздействия объектов друг на друга и передачей сообщения

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 1 – Диаграмма классов

2.2.4 Определение структуры класса-варианта

Информация о поступлении и продаже товара хранится непосредственно в родительском классе, а взаимодействие с данными и их добавление осуществляется при помощи поля id реализованного в классах товар, документ, провайдер.

2.2.5 Определение атрибутов и методов классов

Из пункта 2.2.3 было выяснено, какие понятия следует реализовать в виде классов, следовательно теперь необходимо определить их атрибуты и методы для работы с ними.

Описание классов приведено в таблице

|  |  |  |  |
| --- | --- | --- | --- |
| Класс | Описание | Атрибуты | Методы |
| Magazin | Содержит данные массивов: Товаров, Поставщиков, Документов, Прихода, Расхода и пути к базе данных. | - массив Товаров  - массив Поставщиков  - массив Документов  - массив Поставок  - массив Расходов  - путь к базе данныъ | - запись в файл  - 3 метода получения Имени по id  - 3 метода получения id по Имени |
| Product | Содержит информацию о товаре | - Идентификатор  - Название |  |
| Provider | Содержит информацию о поставщике | - Идентификатор  - Название  - Адрес  - Комментарий |  |
| Document | Содержит данные для всех рыб | - Идентификатор  - Номер договора  - Номер накладной |  |
| Coming | Содержит информацию о поступлении товара | - Идентификатор  - Идентификатор продукта  - Идентификатор поставщика  - Идентификатор документа  - Количество  - стоимость |  |
| Consumption | Содержит информацию о расходе товара | - Идентификатор  - Идентификатор продукта  - Количество |  |

2.2.6 Разработка алгоритмов методов класса-варианта

2.2.6.1 Алгоритм добавления товара

Алгоритм:

При добавлении нового элемента создается экземпляр класса Product, ему присваивается уникальный идентификатор в поле id и так же заполняются остальные поля. Затем этот объект помещается в конец списка ArrayProducts.

2.2.6.2 Алгоритм редактирования товара

Алгоритм:

С помощью Идентификатора находим нужный нам экземпляр класса в списке ArrayProduct, вносим изменения в нужные поля класса и заново заносим его в список ArrayProduct

2.2.6.2 Алгоритм удаления товара

Алгоритм:

По идентификатору находим нужный нам экземпляр класса и удаляем его из ArrayProduct.

2.2.6.1 Алгоритм добавления поставщиков

Алгоритм:

При добавлении нового элемента создается экземпляр класса Provider, ему присваивается уникальный идентификатор в поле id и так же заполняются остальные поля. Затем этот объект помещается в конец списка ArrayProvider.

2.2.6.2 Алгоритм редактирования поставщиков

Алгоритм:

С помощью Идентификатора находим нужный нам экземпляр класса в списке ArrayProvider, вносим изменения в нужные поля класса и заново заносим его в список ArrayProvider

2.2.6.2 Алгоритм удаления поставщиков

Алгоритм:

По идентификатору находим нужный нам экземпляр класса и удаляем его из ArrayProvider.

2.2.6.1 Алгоритм добавления документов

Алгоритм:

При добавлении нового элемента создается экземпляр класса Document, ему присваивается уникальный идентификатор в поле id и так же заполняются остальные поля. Затем этот объект помещается в конец списка ArrayDocument.

2.2.6.2 Алгоритм редактирования документов

Алгоритм:

С помощью Идентификатора находим нужный нам экземпляр класса в списке ArrayDocument, вносим изменения в нужные поля класса и заново заносим его в список ArrayDocument

2.2.6.2 Алгоритм удаления документов

Алгоритм:

По идентификатору находим нужный нам экземпляр класса и удаляем его из ArrayDocument.

2.2.6.1 Алгоритм добавления поступившего товара

Алгоритм:

При добавлении нового элемента создается экземпляр класса Coming, ему присваивается уникальный идентификатор в поле id и так же заполняются остальные поля. Затем этот объект помещается в конец списка ArrayComing.

2.2.6.2 Алгоритм редактирования поступления товара

Алгоритм:

С помощью Идентификатора находим нужный нам экземпляр класса в списке ArrayComing, вносим изменения в нужные поля класса и заново заносим его в список ArrayComing

2.2.6.2 Алгоритм удаления поступления товара

Алгоритм:

По идентификатору находим нужный нам экземпляр класса и удаляем его из ArrayComing.

2.2.6.1 Алгоритм добавления расхода товаров

Алгоритм:

При добавлении нового элемента создается экземпляр класса Coming, ему присваивается уникальный идентификатор в поле id и так же заполняются остальные поля. Затем этот объект помещается в конец списка ArrayComing.

2.2.6.2 Алгоритм редактирования расхода товаров

Алгоритм:

С помощью Идентификатора находим нужный нам экземпляр класса в списке ArrayConsumption, вносим изменения в нужные поля класса и заново заносим его в список ArrayConsumption.

2.2.6.2 Алгоритм удаления расхода товаров

Алгоритм:

По идентификатору находим нужный нам экземпляр класса и удаляем его из ArrayConsumption.

2.2.6.3 Алгоритм подсчета остатков товаров.

Алгоритм:

Проходимся по спискам ArrayComing и ArrayConsumption собираем в отдельные переменные количество поступившего и проданного товара, затем вычитаем из суммы поступившего сумму проданного.

2.3 Описание структуры интерфейса пользователя

Графический интерфейс пользователя проектировался с целью удобства и интуитивности использования. Отсюда вытекает, что кол-во окон должно быть как можно меньше, но при этом они должны быть содержательны и понятны.

После проведённого анализа были спроектированы следующие формы:

1. Форма с аквариумом и кнопками взаимодействия

Форма изображена на рисунке ниже.

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 2 – Интерфейс программы

Таблица 1 – Употребление видимых элементов формы.

|  |  |  |
| --- | --- | --- |
| Название элемента | Количество элементов | Семантика |
| Form | 16 | Представляет окно или диалоговое окно, которое составляет пользовательский интерфейс приложения. |
| MenuStrip | 6 | MenuStrip выступает своего рода контейнером для отдельных пунктов меню, которые представлены объектом ToolStripMenuItem. |
| ToolStripMenuItem | 21 | Представляет доступный для выбора параметр, отображаемый в объекте MenuStrip или ContextMenuStrip. |
| ListView | 6 | Элемент ListView представляет список. |
| Button | 10 | Представляет элемент управления Windows "Кнопка". |
| ComboBox | 6 | предоставляется пользователям в список параметров. Список отображения или скрытия как элемент управления развернут и свернут. |

3 Экспериментальная часть

3.1 описание подхода к тестированию разработанной программы

Тестирование программного обеспечения (software testing) – деятельность, выполняемая для оценки и улучшения качества программного обеспечения. Эта деятельность в общем случае базируется на обнаружении дефектов и проблем в программных системах.

Процесс тестирования является неотъемлемой частью любого процесса разработки программного обеспечения и играет важную роль на всех этапах его создания. Обнаружения ошибок на ранних стадиях жизненного цикла разработки программного обеспечения позволяет снизить затраты на его создание в целом и уменьшить вероятность появления сбоев в дальнейшем.

Тесты строятся с ориентацией на использование тех величин, которые входят и выходят за рамки специфицированных пределов значений.

3.2 Тестирование

Тесты, проведенные над приложением, проверили работу каждой функции, ошибок не было выявлено, примеры-подтверждения представлены ниже.

Изображение выглядит как снимок экрана

Автоматически созданное описаниеИзображение выглядит как снимок экрана

Автоматически созданное описание

Рисунки 3.1 и 3.2 – Добавление товара

Изображение выглядит как снимок экрана

Автоматически созданное описаниеИзображение выглядит как снимок экрана

Автоматически созданное описание

Изображение выглядит как снимок экрана

Автоматически созданное описаниеИзображение выглядит как снимок экрана

Автоматически созданное описаниеРисунки 4.1 и 4.2 – Редактирование товара

Изображение выглядит как снимок экрана

Автоматически созданное описаниеИзображение выглядит как снимок экрана

Автоматически созданное описаниеРисунки 5.1 и 5.2 – Удаление товара

Рисунки 6.1 и 6.2 – Добавление поставщиков

Изображение выглядит как снимок экрана

Автоматически созданное описаниеИзображение выглядит как снимок экрана

Автоматически созданное описание

Изображение выглядит как снимок экрана

Автоматически созданное описаниеИзображение выглядит как снимок экрана

Автоматически созданное описаниеРисунок 7.1 и 7.2 – Редактирование поставщиков

Рисунок 8.1 и 8.2 – Удаление поставщиков

Изображение выглядит как снимок экрана

Автоматически созданное описание

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 9.1 и 9.2 – Добавление документов

Изображение выглядит как снимок экрана

Автоматически созданное описаниеИзображение выглядит как снимок экрана

Автоматически созданное описание

Изображение выглядит как снимок экрана

Автоматически созданное описаниеИзображение выглядит как снимок экрана

Автоматически созданное описаниеРисунок 10.1 и 10.2 – Редактирование документов

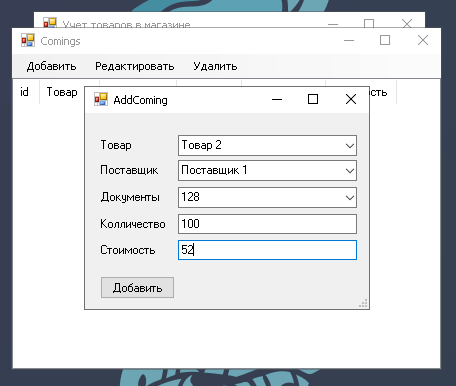
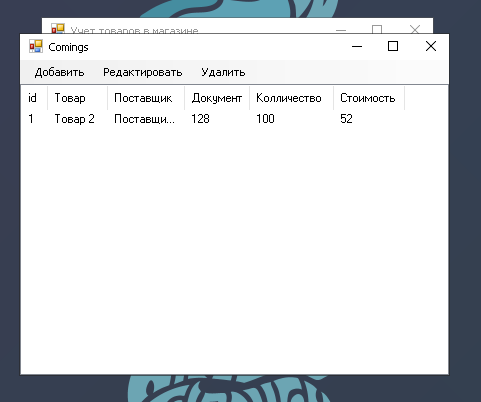
Рисунок 11.1 и 11.2 – Удаление документов

Рисунок 12.1 и 12.2 – Добавление прихода

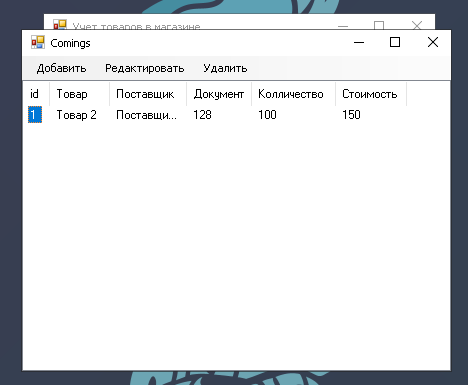
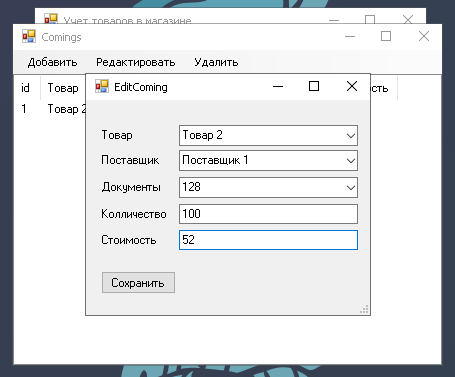
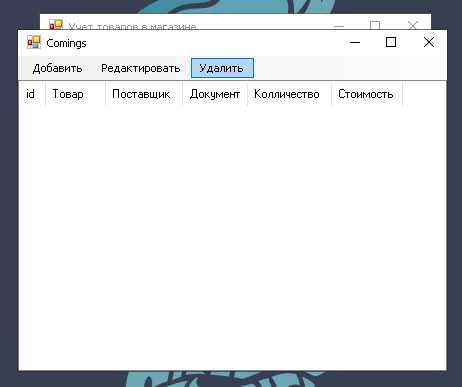
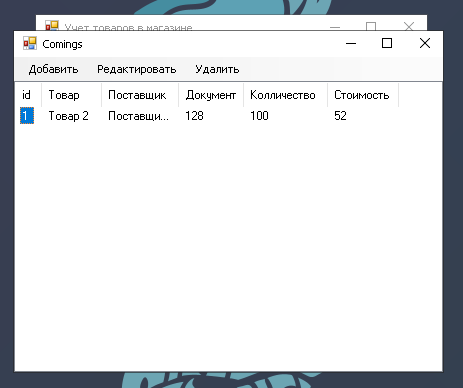


Рисунок 13.1 и 13.2 – Редактирование прихода



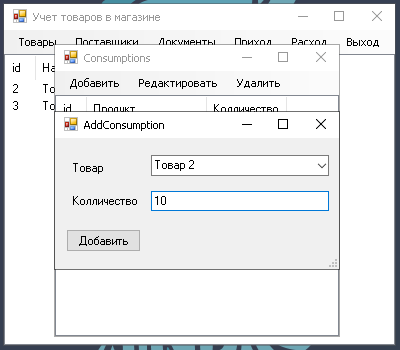
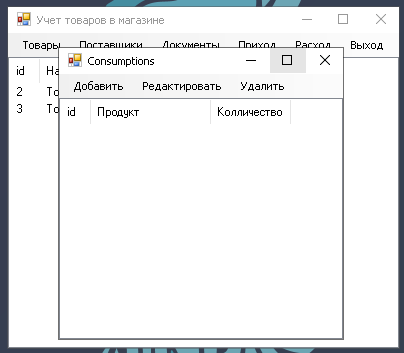
Рисунок 14.1 и 14.2 – Удаление прихода

Рисунок 15.1 и 15.2 – Добавление расхода

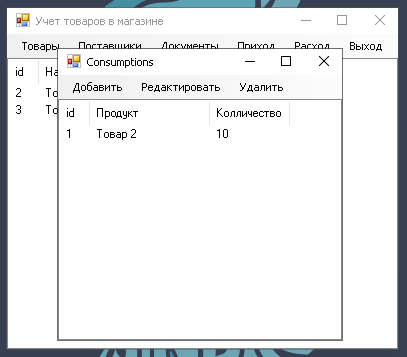
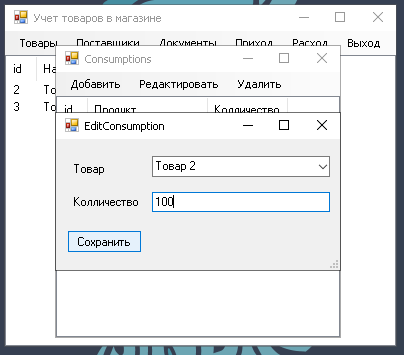


Рисунок 16.1 и 16.2 – Редактирование расхода

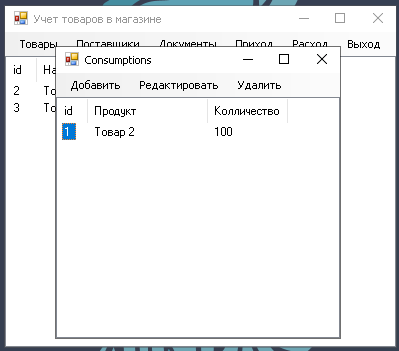
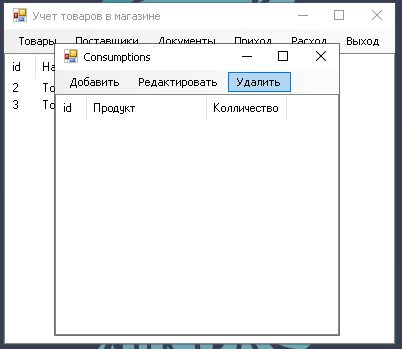


Рисунок 17.1 и 17.2 – Удаление расхода

3.3 Инструкция пользователя

3.3.1 Запуск

При открытии приложения открывается главное окно со статистикой по всем оставшимся товарам.

3.3.2 Функция «Добавления товара»

Для того, чтобы добавить товар перейдите в раздел «Товары», нажмите «Добавить», заполните нужные поля, нажмите кнопку «Добавить».

3.3.3 Функция «Редактирование товара»

Для того, чтобы отредактировать товар перейдите в раздел «Товары», выберите нужный вам товар по полю «ID», нажмите «Редактировать», измените нужные поля, нажмите кнопку «Сохранить».

3.3.4 Функция «Удаление товара»

Для того, чтобы удалить ненужный товар перейдите в раздел «Товары», выберите нужный вам товар по полю «ID», нажмите «Удалить».

3.3.5 Функция «Добавления поставщиков»

Для того, чтобы добавить товар перейдите в раздел «Поставщики», нажмите «Добавить», заполните нужные поля, нажмите кнопку «Добавить».

3.3.3 Функция «Редактирование поставщиков»

Для того, чтобы отредактировать товар перейдите в раздел «Поставщики», выберите нужного вам поставщика по полю «ID», нажмите «Редактировать», измените нужные поля, нажмите кнопку «Сохранить».

3.3.4 Функция «Удаление поставщиков»

Для того, чтобы удалить ненужный товар перейдите в раздел «Поставщики», выберите товар по полю «ID», нажмите «Удалить».

3.3.2 Функция «Добавления документов»

Для того, чтобы добавить товар перейдите в раздел «Документы», нажмите «Добавить», заполните нужные поля, нажмите кнопку «Добавить».

3.3.3 Функция «Редактирование документов»

Для того, чтобы отредактировать товар перейдите в раздел «Документы», выберите нужный вам документ по полю «ID», нажмите «Редактировать», измените нужные поля, нажмите кнопку «Сохранить».

3.3.4 Функция «Удаление документов»

Для того, чтобы удалить ненужный товар перейдите в раздел «Документы», выберите нужный вам документ по полю «ID», нажмите «Удалить.

3.3.2 Функция «Добавления прихода»

Для того, чтобы добавить товар перейдите в раздел «Приход», нажмите «Добавить», заполните нужные поля, нажмите кнопку «Добавить».

3.3.3 Функция «Редактирование прихода»

Для того, чтобы отредактировать товар перейдите в раздел «Приход», выберите нужный вам приход по полю «ID», нажмите «Редактировать», измените нужные поля, нажмите кнопку «Сохранить».

3.3.4 Функция «Удаление прихода»

Для того, чтобы удалить ненужный товар перейдите в раздел «Приход», выберите нужный вам приход по полю «ID», нажмите «Удалить».

3.3.2 Функция «Добавления расхода»

Для того, чтобы добавить товар перейдите в раздел «Расход», нажмите «Добавить», заполните нужные поля, нажмите кнопку «Добавить».

3.3.3 Функция «Редактирование расхода»

Для того, чтобы отредактировать товар перейдите в раздел «Расход», выберите нужный вам расход по полю «ID», нажмите «Редактировать», измените нужные поля, нажмите кнопку «Сохранить».

3.3.4 Функция «Удаление расхода»

Для того, чтобы удалить ненужный товар перейдите в раздел «Расход», выберите нужный вам расход по полю «ID», нажмите «Удалить».

Заключение

В результате выполнения курсового проекта, была разработана программная система моделирования движения рыб в аквариуме.

По итогу выполненного курсового проекта были изучены

* объектно-ориентированный анализ и проектирование
* технологии структурного программирования с использованием современных средств: модуль и объект
* элементы технологии создания программного продукта
* разработка на языке С++ в среде MS Visual Studio 2017 с использованием Windows Forms
* проектирование графического интерфейса пользователя
* тестирование программных компонентов
* качественное оформление документации на программное обеспечение

Список используемых источников

1. Джеффри Рихтер CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# 4-е издание / Пер. с англ – М. : Издательство «Питер»; СПб. : СПб, 2013. – 896 стр. :ил
2. Прата Стивен. Язык программирования C++. Лекции и упражнения. 6-е издание. Издание: Отдельное издание;
3. [https://docs.microsoft.com/](https://docs.microsoft.com/ru-ru/dotnet/csharp/) [Электронный ресурс]. Официальная документация по языку C# от корпорации Microsoft (Дата обращения: 18.06.2019).

ПРИЛОЖЕНИЕ

MyForm.cpp

#include "MyForm.h"

using namespace System;

using namespace System::Windows::Forms;

[STAThreadAttribute]

void main(cli::array<String^>^ args) {

Application::EnableVisualStyles();

Application::SetCompatibleTextRenderingDefault(false);

Project1::MyForm form;

Application::Run(%form);

}

MyForm.h

#pragma once

#include <windows.h>

#include "Magazin.h"

#include "Providers.h"

#include "Documents.h"

#include "Product.h"

#include "Products.h"

#include "Comings.h"

#include "Consumption.h"

#include "Consumptions.h"

namespace Project1 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace Newtonsoft::Json;

using namespace Newtonsoft::Json::Linq;

using namespace System::IO;

/// <summary>

/// Сводка для MyForm

/// </summary>

public ref class MyForm : public System::Windows::Forms::Form

{

public:

MyForm()

{

InitializeComponent();

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~MyForm()

{

if (components)

{

delete components;

}

}

protected:

private: System::Windows::Forms::MenuStrip^ menuStrip1;

private: System::Windows::Forms::ListView^ listView1;

private: System::Windows::Forms::ColumnHeader^ columnName;

private: System::Windows::Forms::ToolStripMenuItem^ ProvidersToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ documentToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ ExitToolStripMenuItem;

private: System::Windows::Forms::ColumnHeader^ columnHeader7;

private: System::Windows::Forms::ToolStripMenuItem^ товарToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ приходToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ расходToolStripMenuItem;

private: System::Windows::Forms::ContextMenuStrip^ contextMenuStrip1;

private: System::Windows::Forms::ColumnHeader^ columnHeader1;

private: System::ComponentModel::IContainer^ components;

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->components = (gcnew System::ComponentModel::Container());

this->menuStrip1 = (gcnew System::Windows::Forms::MenuStrip());

this->товарToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->ProvidersToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->documentToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->приходToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->расходToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->ExitToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->listView1 = (gcnew System::Windows::Forms::ListView());

this->columnHeader1 = (gcnew System::Windows::Forms::ColumnHeader());

this->columnName = (gcnew System::Windows::Forms::ColumnHeader());

this->columnHeader7 = (gcnew System::Windows::Forms::ColumnHeader());

this->contextMenuStrip1 = (gcnew System::Windows::Forms::ContextMenuStrip(this->components));

this->menuStrip1->SuspendLayout();

this->SuspendLayout();

//

// menuStrip1

//

this->menuStrip1->Items->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(6) {

this->товарToolStripMenuItem,

this->ProvidersToolStripMenuItem, this->documentToolStripMenuItem, this->приходToolStripMenuItem, this->расходToolStripMenuItem,

this->ExitToolStripMenuItem

});

this->menuStrip1->Location = System::Drawing::Point(0, 0);

this->menuStrip1->Name = L"menuStrip1";

this->menuStrip1->Size = System::Drawing::Size(391, 24);

this->menuStrip1->TabIndex = 2;

this->menuStrip1->Text = L"menuStrip1";

//

// товарToolStripMenuItem

//

this->товарToolStripMenuItem->Name = L"товарToolStripMenuItem";

this->товарToolStripMenuItem->Size = System::Drawing::Size(57, 20);

this->товарToolStripMenuItem->Text = L"Товары";

this->товарToolStripMenuItem->Click += gcnew System::EventHandler(this, &MyForm::ТоварToolStripMenuItem\_Click);

//

// ProvidersToolStripMenuItem

//

this->ProvidersToolStripMenuItem->Name = L"ProvidersToolStripMenuItem";

this->ProvidersToolStripMenuItem->Size = System::Drawing::Size(82, 20);

this->ProvidersToolStripMenuItem->Text = L"Поставщики";

this->ProvidersToolStripMenuItem->Click += gcnew System::EventHandler(this, &MyForm::ProvidersToolStripMenuItem\_Click);

//

// documentToolStripMenuItem

//

this->documentToolStripMenuItem->Name = L"documentToolStripMenuItem";

this->documentToolStripMenuItem->Size = System::Drawing::Size(77, 20);

this->documentToolStripMenuItem->Text = L"Документы";

this->documentToolStripMenuItem->Click += gcnew System::EventHandler(this, &MyForm::DocumentToolStripMenuItem\_Click);

//

// приходToolStripMenuItem

//

this->приходToolStripMenuItem->Name = L"приходToolStripMenuItem";

this->приходToolStripMenuItem->Size = System::Drawing::Size(57, 20);

this->приходToolStripMenuItem->Text = L"Приход";

this->приходToolStripMenuItem->Click += gcnew System::EventHandler(this, &MyForm::ComingToolStripMenuItem\_Click);

//

// расходToolStripMenuItem

//

this->расходToolStripMenuItem->Name = L"расходToolStripMenuItem";

this->расходToolStripMenuItem->Size = System::Drawing::Size(55, 20);

this->расходToolStripMenuItem->Text = L"Расход";

this->расходToolStripMenuItem->Click += gcnew System::EventHandler(this, &MyForm::ConsumptionToolStripMenuItem\_Click);

//

// ExitToolStripMenuItem

//

this->ExitToolStripMenuItem->Name = L"ExitToolStripMenuItem";

this->ExitToolStripMenuItem->Size = System::Drawing::Size(52, 20);

this->ExitToolStripMenuItem->Text = L"Выход";

this->ExitToolStripMenuItem->Click += gcnew System::EventHandler(this, &MyForm::ExitToolStripMenuItem\_Click);

//

// listView1

//

this->listView1->Columns->AddRange(gcnew cli::array< System::Windows::Forms::ColumnHeader^ >(3) {

this->columnHeader1, this->columnName,

this->columnHeader7

});

this->listView1->Dock = System::Windows::Forms::DockStyle::Fill;

this->listView1->Location = System::Drawing::Point(0, 24);

this->listView1->Name = L"listView1";

this->listView1->Size = System::Drawing::Size(391, 291);

this->listView1->TabIndex = 4;

this->listView1->UseCompatibleStateImageBehavior = false;

this->listView1->View = System::Windows::Forms::View::Details;

//

// columnHeader1

//

this->columnHeader1->Text = L"id";

this->columnHeader1->Width = 30;

//

// columnName

//

this->columnName->Text = L"Название";

this->columnName->Width = 118;

//

// columnHeader7

//

this->columnHeader7->Text = L"Количество";

this->columnHeader7->Width = 90;

//

// contextMenuStrip1

//

this->contextMenuStrip1->Name = L"contextMenuStrip1";

this->contextMenuStrip1->Size = System::Drawing::Size(61, 4);

//

// MyForm

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(391, 315);

this->Controls->Add(this->listView1);

this->Controls->Add(this->menuStrip1);

this->MainMenuStrip = this->menuStrip1;

this->Name = L"MyForm";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"Учет товаров в магазине";

this->Load += gcnew System::EventHandler(this, &MyForm::MyForm\_Load);

this->menuStrip1->ResumeLayout(false);

this->menuStrip1->PerformLayout();

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

private: Magazin^ mgz;

private: String^ pathDB = Application::StartupPath + "/db/";

private: String^ fileDBName = "magazins.json";

private: System::Void MyForm\_Load(System::Object^ sender, System::EventArgs^ e) {

if (File::Exists(pathDB + fileDBName)) {

try

{

mgz = JsonConvert::DeserializeObject<Magazin^>(File::ReadAllText(pathDB + fileDBName));

}

catch (Exception^ e)

{

MessageBox::Show("Поврежден файл базы!!!", "Ошибка!", MessageBoxButtons::OK, MessageBoxIcon::Error);

Environment::Exit(0);

}

}

else {

if (!Directory::Exists(pathDB)) {

Directory::CreateDirectory(pathDB);

}

FileStream^ fs = File::Create(pathDB + fileDBName);

fs->Close();

InitMagazin();

}

this->UpdateListView();

}

private: System::Void ProvidersToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

Providers^ form = gcnew Providers();

mgz = form->Provider\_Shown(mgz);

this->UpdateListView();

}

private: System::Void DocumentToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

Documents^ form = gcnew Documents();

mgz = form->Document\_Shown(mgz);

this->UpdateListView();

}

private: System::Void ExitToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

Application::Exit();

}

// column click

//private: System::Void ListView1\_ColumnClick(System::Object^ sender, System::Windows::Forms::ColumnClickEventArgs^ e) {

// Console::WriteLine("click column" + " " + e->Column);//вывод номер колонки

//}

private: System::Void ТоварToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

Products^ form = gcnew Products();

mgz = form->Products\_Shown(mgz);

this->UpdateListView();

}

private: System::Void ComingToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

Comings^ form = gcnew Comings();

mgz = form->Coming\_Shown(mgz);

this->UpdateListView();

}

private: System::Void ConsumptionToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

Consumptions^ form = gcnew Consumptions();

mgz = form->Consumption\_Shown(mgz);

this->UpdateListView();

}

private: System::Void UpdateListView() {

listView1->Items->Clear();

for (int i = 0; i < this->mgz->ArrayProduct->Count; i++) {

ListViewItem^ Id = gcnew ListViewItem();

Id->Text = this->mgz->ArrayProduct[i]->id.ToString();

ListViewItem::ListViewSubItem^ Product = gcnew ListViewItem::ListViewSubItem();

Product->Text = this->mgz->ArrayProduct[i]->Name;

Id->SubItems->Add(Product);

ListViewItem::ListViewSubItem^ Provider = gcnew ListViewItem::ListViewSubItem();

Provider->Text = this->ResidualGoods(this->mgz->ArrayProduct[i]).ToString();

Id->SubItems->Add(Provider);

listView1->Items->Add(Id);

}

}

private: int ResidualGoods(Product^ product) {

int countComing = 0;

int countConsumption = 0;

for each (Coming ^ item in this->mgz->ArrayComing)

{

if (item->productId == product->id)

countComing += item->count;

}

for each (Consumption ^ item in this->mgz->ArrayConsumption)

{

if (item->productId == product->id)

countConsumption += item->count;

}

return countComing - countConsumption;

}

private: System::Void InitMagazin() {

mgz = gcnew Magazin();

mgz->ArrayProvider = gcnew List<Provider^>();

mgz->ArrayDocument = gcnew List<Document^>();

mgz->ArrayProduct = gcnew List<Product^>();

mgz->ArrayComing = gcnew List<Coming^>();

mgz->ArrayConsumption = gcnew List<Consumption^>();

}

};

}

Magazin.h

#pragma once

#include "Product.h"

#include "Provider.h"

#include "Document.h"

#include "Coming.h"

#include "Consumption.h"

using namespace System;

using namespace System::Collections;

using namespace System::Collections::Generic;

using namespace System::Windows::Forms;

using namespace Newtonsoft::Json;

using namespace Newtonsoft::Json::Linq;

using namespace System::IO;

public ref class Magazin

{

private: String^ pathDB = Application::StartupPath + "/db/";

private: String^ fileDBName = "magazins.json";

public: List<Provider^>^ ArrayProvider;

public: List<Product^>^ ArrayProduct;

public: List<Document^>^ ArrayDocument;

public: List<Coming^>^ ArrayComing;

public: List<Consumption^>^ ArrayConsumption;

public: unsigned int getIdProductByName(String^ name) {

for each (Product ^ item in this->ArrayProduct)

{

if (item->Name == name)

return item->id;

}

return 0;

}

public: String^ getNameProductById(unsigned int id) {

for each (Product ^ item in this->ArrayProduct)

{

if (item->id == id)

return item->Name;

}

return "";

}

public: unsigned int getIdProviderByName(String^ name) {

for each (Provider ^ item in this->ArrayProvider)

{

if (item->Name == name)

return item->id;

}

return 0;

}

public: String^ getNameProviderById(unsigned int id) {

for each (Provider ^ item in this->ArrayProvider)

{

if (item->id == id)

return item->Name;

}

return "";

}

public: unsigned int getIdDocumentByNumberDogovor(String^ ndog) {

for each (Document ^ item in this->ArrayDocument)

{

if (item->NumberDogovor == ndog)

return item->id;

}

return 0;

}

public: String^ getNumberDogovorById(unsigned int id) {

for each (Document ^ item in this->ArrayDocument)

{

if (item->id == id)

return item->NumberDogovor;

}

return "";

}

public: System::Void WriteToFile() {

File::WriteAllText(pathDB + fileDBName, JsonConvert::SerializeObject(this));

}

};

Provider.h

#pragma once

using namespace System;

ref class Provider

{

public: int id;

public: String^ Name;

public: String^ Addres;

public: String^ Comment;

};

Product.h

#pragma once

using namespace std;

using namespace System;

ref class Product

{

public: int id;

public: String^ Name;

};

Document.h

#pragma once

using namespace System;

ref class Document

{

public: int id;

public: String^ NumberDogovor;

public: String^ Naklodnaya;

};

Coming.h

#pragma once

#include "Product.h"

#include "Provider.h"

#include "Document.h"

using namespace System;

ref class Coming

{

public: unsigned int id;

public: unsigned int productId;

public: unsigned int providerId;

public: unsigned int documentId;

public: int count;

public: long price;

};

Consumption.h

#pragma once

#include "Product.h"

using namespace std;

using namespace System;

ref class Consumption

{

public: unsigned int id;

public: unsigned int productId;

public: int count;

};

Products.h

#pragma once

#include "Magazin.h"

#include "Product.h"

#include "AddProduct.h"

#include "EditProduct.h"

namespace Project1 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace Newtonsoft::Json;

using namespace Newtonsoft::Json::Linq;

using namespace System::IO;

/// <summary>

/// Сводка для Products

/// </summary>

public ref class Products : public System::Windows::Forms::Form

{

public:

Products(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~Products()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::ListView^ listView1;

private: System::Windows::Forms::ColumnHeader^ columnHeader1;

private: System::Windows::Forms::ColumnHeader^ columnHeader2;

private: System::Windows::Forms::MenuStrip^ menuStrip1;

private: System::Windows::Forms::ToolStripMenuItem^ продуктыToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ редактироватьToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ удалитьToolStripMenuItem;

protected:

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->listView1 = (gcnew System::Windows::Forms::ListView());

this->columnHeader1 = (gcnew System::Windows::Forms::ColumnHeader());

this->columnHeader2 = (gcnew System::Windows::Forms::ColumnHeader());

this->menuStrip1 = (gcnew System::Windows::Forms::MenuStrip());

this->продуктыToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->редактироватьToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->удалитьToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->menuStrip1->SuspendLayout();

this->SuspendLayout();

//

// listView1

//

this->listView1->Columns->AddRange(gcnew cli::array< System::Windows::Forms::ColumnHeader^ >(2) { this->columnHeader1, this->columnHeader2 });

this->listView1->Dock = System::Windows::Forms::DockStyle::Fill;

this->listView1->Location = System::Drawing::Point(0, 24);

this->listView1->Name = L"listView1";

this->listView1->Size = System::Drawing::Size(384, 291);

this->listView1->TabIndex = 0;

this->listView1->UseCompatibleStateImageBehavior = false;

this->listView1->View = System::Windows::Forms::View::Details;

//

// columnHeader1

//

this->columnHeader1->Text = L"id";

//

// columnHeader2

//

this->columnHeader2->Text = L"Название";

this->columnHeader2->Width = 80;

//

// menuStrip1

//

this->menuStrip1->Items->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(3) {

this->продуктыToolStripMenuItem,

this->редактироватьToolStripMenuItem, this->удалитьToolStripMenuItem

});

this->menuStrip1->Location = System::Drawing::Point(0, 0);

this->menuStrip1->Name = L"menuStrip1";

this->menuStrip1->Size = System::Drawing::Size(384, 24);

this->menuStrip1->TabIndex = 1;

this->menuStrip1->Text = L"menuStrip1";

//

// продуктыToolStripMenuItem

//

this->продуктыToolStripMenuItem->Name = L"продуктыToolStripMenuItem";

this->продуктыToolStripMenuItem->Size = System::Drawing::Size(69, 20);

this->продуктыToolStripMenuItem->Text = L"Добавить";

this->продуктыToolStripMenuItem->Click += gcnew System::EventHandler(this, &Products::CreateToolStripMenuItem\_Click);

//

// редактироватьToolStripMenuItem

//

this->редактироватьToolStripMenuItem->Name = L"редактироватьToolStripMenuItem";

this->редактироватьToolStripMenuItem->Size = System::Drawing::Size(98, 20);

this->редактироватьToolStripMenuItem->Text = L"Редактировать";

this->редактироватьToolStripMenuItem->Click += gcnew System::EventHandler(this, &Products::EditToolStripMenuItem\_Click);

//

// удалитьToolStripMenuItem

//

this->удалитьToolStripMenuItem->Name = L"удалитьToolStripMenuItem";

this->удалитьToolStripMenuItem->Size = System::Drawing::Size(63, 20);

this->удалитьToolStripMenuItem->Text = L"Удалить";

this->удалитьToolStripMenuItem->Click += gcnew System::EventHandler(this, &Products::RemoveToolStripMenuItem\_Click);

//

// Products

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(384, 315);

this->Controls->Add(this->listView1);

this->Controls->Add(this->menuStrip1);

this->Name = L"Products";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"Products";

this->Load += gcnew System::EventHandler(this, &Products::Products\_Load);

this->menuStrip1->ResumeLayout(false);

this->menuStrip1->PerformLayout();

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

public: Product^ pr;

public: Magazin^ magz;

public: Magazin^ Products\_Shown(Magazin^ m) {

magz = m;

listView1->Items->Clear();

this->ShowDialog();

return magz;

}

private: System::Void Products\_Load(System::Object^ sender, System::EventArgs^ e) {

this->UpdateListView();

}

private: System::Void UpdateListView() {

listView1->Items->Clear();

for (int i = 0; i < this->magz->ArrayProduct->Count; i++) {

ListViewItem^ Id = gcnew ListViewItem();

Id->Text = this->magz->ArrayProduct[i]->id.ToString();

ListViewItem::ListViewSubItem^ Name = gcnew ListViewItem::ListViewSubItem();

Name->Text = this->magz->ArrayProduct[i]->Name;

Id->SubItems->Add(Name);

listView1->Items->Add(Id);

}

}

private: System::Void CreateToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

AddProduct^ form = gcnew AddProduct();

magz = form->AddProduct\_Shown(magz);

this->UpdateListView();

magz->WriteToFile();

}

private: System::Void EditToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

if (listView1->SelectedItems->Count > 0 &&

listView1->SelectedItems->Count < 2) {

EditProduct^ form = gcnew EditProduct();

magz = form->EditProduct\_Shown(this->magz, (int)Convert::ToInt32(listView1->SelectedItems[0]->SubItems[0]->Text));

this->UpdateListView();

magz->WriteToFile();

}

else if (listView1->SelectedItems->Count >= 2) {

MessageBox::Show("Выбрано много элементов для редактирования!!!", "Ошибка!", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

else {

MessageBox::Show("Не выбран элемент для редактирования!!!", "Ошибка!", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

private: System::Void RemoveToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

for (int i = 0; i < listView1->SelectedItems->Count; i++) {

int id = Convert::ToInt32(listView1->SelectedItems[i]->SubItems[0]->Text);

for (int i = 0; i < magz->ArrayProduct->Count; i++)

{

if (magz->ArrayProduct[i]->id == id)

{

magz->ArrayProduct->Remove(magz->ArrayProduct[i]);

}

}

}

this->UpdateListView();

magz->WriteToFile();

}

};

}

Providers.h

#pragma once

#include "AddProvider.h"

#include "EditProvider.h"

namespace Project1 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Сводка для Providers

/// </summary>

public ref class Providers : public System::Windows::Forms::Form

{

public:

Providers(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~Providers()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::MenuStrip^ menuStrip1;

protected:

private: System::Windows::Forms::ToolStripMenuItem^ создатьToolStripMenuItem;

private: System::Windows::Forms::ListView^ listView1;

private: System::Windows::Forms::ColumnHeader^ columnHeader1;

private: System::Windows::Forms::ColumnHeader^ columnHeader2;

private: System::Windows::Forms::ToolStripMenuItem^ редактироватьToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ удалитьToolStripMenuItem;

private: System::Windows::Forms::ColumnHeader^ columnHeader3;

private: System::Windows::Forms::ColumnHeader^ columnHeader4;

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container^ components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->menuStrip1 = (gcnew System::Windows::Forms::MenuStrip());

this->создатьToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->редактироватьToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->удалитьToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->listView1 = (gcnew System::Windows::Forms::ListView());

this->columnHeader1 = (gcnew System::Windows::Forms::ColumnHeader());

this->columnHeader2 = (gcnew System::Windows::Forms::ColumnHeader());

this->columnHeader3 = (gcnew System::Windows::Forms::ColumnHeader());

this->columnHeader4 = (gcnew System::Windows::Forms::ColumnHeader());

this->menuStrip1->SuspendLayout();

this->SuspendLayout();

//

// menuStrip1

//

this->menuStrip1->Items->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(3) {

this->создатьToolStripMenuItem,

this->редактироватьToolStripMenuItem, this->удалитьToolStripMenuItem

});

this->menuStrip1->Location = System::Drawing::Point(0, 0);

this->menuStrip1->Name = L"menuStrip1";

this->menuStrip1->Size = System::Drawing::Size(418, 24);

this->menuStrip1->TabIndex = 0;

this->menuStrip1->Text = L"menuStrip1";

//

// создатьToolStripMenuItem

//

this->создатьToolStripMenuItem->Name = L"создатьToolStripMenuItem";

this->создатьToolStripMenuItem->Size = System::Drawing::Size(69, 20);

this->создатьToolStripMenuItem->Text = L"Добавить";

this->создатьToolStripMenuItem->Click += gcnew System::EventHandler(this, &Providers::CreateToolStripMenuItem\_Click);

//

// редактироватьToolStripMenuItem

//

this->редактироватьToolStripMenuItem->Name = L"редактироватьToolStripMenuItem";

this->редактироватьToolStripMenuItem->Size = System::Drawing::Size(98, 20);

this->редактироватьToolStripMenuItem->Text = L"Редактировать";

this->редактироватьToolStripMenuItem->Click += gcnew System::EventHandler(this, &Providers::EditToolStripMenuItem\_Click);

//

// удалитьToolStripMenuItem

//

this->удалитьToolStripMenuItem->Name = L"удалитьToolStripMenuItem";

this->удалитьToolStripMenuItem->Size = System::Drawing::Size(66, 20);

this->удалитьToolStripMenuItem->Text = L" Удалить";

this->удалитьToolStripMenuItem->Click += gcnew System::EventHandler(this, &Providers::RemoveToolStripMenuItem\_Click);

//

// listView1

//

this->listView1->Columns->AddRange(gcnew cli::array< System::Windows::Forms::ColumnHeader^ >(4) {

this->columnHeader1, this->columnHeader2,

this->columnHeader3, this->columnHeader4

});

this->listView1->Dock = System::Windows::Forms::DockStyle::Fill;

this->listView1->Location = System::Drawing::Point(0, 24);

this->listView1->Name = L"listView1";

this->listView1->Size = System::Drawing::Size(418, 241);

this->listView1->TabIndex = 1;

this->listView1->UseCompatibleStateImageBehavior = false;

this->listView1->View = System::Windows::Forms::View::Details;

//

// columnHeader1

//

this->columnHeader1->Text = L"№";

this->columnHeader1->Width = 27;

//

// columnHeader2

//

this->columnHeader2->Text = L"Название";

this->columnHeader2->Width = 117;

//

// columnHeader3

//

this->columnHeader3->Text = L" Адрес";

this->columnHeader3->Width = 151;

//

// columnHeader4

//

this->columnHeader4->Text = L"Комментарий";

this->columnHeader4->Width = 103;

//

// Providers

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(418, 265);

this->Controls->Add(this->listView1);

this->Controls->Add(this->menuStrip1);

this->MainMenuStrip = this->menuStrip1;

this->Name = L"Providers";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"Providers";

this->Load += gcnew System::EventHandler(this, &Providers::Providers\_Load);

this->menuStrip1->ResumeLayout(false);

this->menuStrip1->PerformLayout();

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

public: Provider^ provider;

public: Magazin^ magz;

public: Magazin^ Provider\_Shown(Magazin^ marr) {

magz = marr;

listView1->Items->Clear();

this->ShowDialog();

return magz;

}

private: System::Void CreateToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

AddProvider^ form = gcnew AddProvider();

magz = form->AddProvider\_Shown(this->magz);

this->UpdateListView();

magz->WriteToFile();

}

private: System::Void Providers\_Load(System::Object^ sender, System::EventArgs^ e) {

this->UpdateListView();

}

private: System::Void EditToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

if (listView1->SelectedItems->Count > 0 &&

listView1->SelectedItems->Count < 2) {

EditProvider^ form = gcnew EditProvider();

magz = form->EditProvider\_Shown(this->magz, (int)Convert::ToInt32(listView1->SelectedItems[0]->SubItems[0]->Text));

this->UpdateListView();

magz->WriteToFile();

}

else if (listView1->SelectedItems->Count >= 2) {

MessageBox::Show("Выбрано много элементов для редактирования!!!", "Ошибка!", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

else {

MessageBox::Show("Не выбран элемент для редактирования!!!", "Ошибка!", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

private: System::Void RemoveToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

for (int i = 0; i < listView1->SelectedItems->Count; i++) {

int id = Convert::ToInt32(listView1->SelectedItems[i]->SubItems[0]->Text);

for (int i = 0; i < magz->ArrayProvider->Count; i++)

{

if (magz->ArrayProvider[i]->id == id)

{

magz->ArrayProvider->Remove(magz->ArrayProvider[i]);

}

}

}

this->UpdateListView();

magz->WriteToFile();

}

private: System::Void UpdateListView() {

listView1->Items->Clear();

for (int i = 0; i < this->magz->ArrayProvider->Count; i++) {

ListViewItem^ Id = gcnew ListViewItem();

Id->Text = this->magz->ArrayProvider[i]->id.ToString();

ListViewItem::ListViewSubItem^ Name = gcnew ListViewItem::ListViewSubItem();

Name->Text = this->magz->ArrayProvider[i]->Name;

Id->SubItems->Add(Name);

ListViewItem::ListViewSubItem^ Adress = gcnew ListViewItem::ListViewSubItem();

Adress->Text = this->magz->ArrayProvider[i]->Addres;

Id->SubItems->Add(Adress);

ListViewItem::ListViewSubItem^ Comment = gcnew ListViewItem::ListViewSubItem();

Comment->Text = this->magz->ArrayProvider[i]->Comment;

Id->SubItems->Add(Comment);

listView1->Items->Add(Id);

}

}

};

}

Documents.h

#pragma once

#include "Magazin.h"

#include "Document.h"

#include "AddDocument.h"

#include "EditdDocument.h"

namespace Project1 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Сводка для Documents

/// </summary>

public ref class Documents : public System::Windows::Forms::Form

{

public:

Documents(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~Documents()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::MenuStrip^ menuStrip1;

private: System::Windows::Forms::ContextMenuStrip^ contextMenuStrip1;

private: System::Windows::Forms::ListView^ listView1;

private: System::Windows::Forms::ColumnHeader^ columnHeader1;

private: System::Windows::Forms::ColumnHeader^ columnHeader2;

private: System::Windows::Forms::ColumnHeader^ columnHeader3;

private: System::Windows::Forms::ToolStripMenuItem^ добавитьToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ редактироватьToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ удалитьToolStripMenuItem;

private: System::ComponentModel::IContainer^ components;

protected:

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->components = (gcnew System::ComponentModel::Container());

this->menuStrip1 = (gcnew System::Windows::Forms::MenuStrip());

this->добавитьToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->редактироватьToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->удалитьToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->contextMenuStrip1 = (gcnew System::Windows::Forms::ContextMenuStrip(this->components));

this->listView1 = (gcnew System::Windows::Forms::ListView());

this->columnHeader1 = (gcnew System::Windows::Forms::ColumnHeader());

this->columnHeader2 = (gcnew System::Windows::Forms::ColumnHeader());

this->columnHeader3 = (gcnew System::Windows::Forms::ColumnHeader());

this->menuStrip1->SuspendLayout();

this->SuspendLayout();

//

// menuStrip1

//

this->menuStrip1->Items->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(3) {

this->добавитьToolStripMenuItem,

this->редактироватьToolStripMenuItem, this->удалитьToolStripMenuItem

});

this->menuStrip1->Location = System::Drawing::Point(0, 0);

this->menuStrip1->Name = L"menuStrip1";

this->menuStrip1->Size = System::Drawing::Size(384, 24);

this->menuStrip1->TabIndex = 0;

this->menuStrip1->Text = L"menuStrip1";

//

// добавитьToolStripMenuItem

//

this->добавитьToolStripMenuItem->Name = L"добавитьToolStripMenuItem";

this->добавитьToolStripMenuItem->Size = System::Drawing::Size(69, 20);

this->добавитьToolStripMenuItem->Text = L"Добавить";

this->добавитьToolStripMenuItem->Click += gcnew System::EventHandler(this, &Documents::CreateToolStripMenuItem\_Click);

//

// редактироватьToolStripMenuItem

//

this->редактироватьToolStripMenuItem->Name = L"редактироватьToolStripMenuItem";

this->редактироватьToolStripMenuItem->Size = System::Drawing::Size(98, 20);

this->редактироватьToolStripMenuItem->Text = L"Редактировать";

this->редактироватьToolStripMenuItem->Click += gcnew System::EventHandler(this, &Documents::EditToolStripMenuItem\_Click);

//

// удалитьToolStripMenuItem

//

this->удалитьToolStripMenuItem->Name = L"удалитьToolStripMenuItem";

this->удалитьToolStripMenuItem->Size = System::Drawing::Size(63, 20);

this->удалитьToolStripMenuItem->Text = L"Удалить";

this->удалитьToolStripMenuItem->Click += gcnew System::EventHandler(this, &Documents::RemoveToolStripMenuItem\_Click);

//

// contextMenuStrip1

//

this->contextMenuStrip1->Name = L"contextMenuStrip1";

this->contextMenuStrip1->Size = System::Drawing::Size(61, 4);

//

// listView1

//

this->listView1->Columns->AddRange(gcnew cli::array< System::Windows::Forms::ColumnHeader^ >(3) {

this->columnHeader1, this->columnHeader2,

this->columnHeader3

});

this->listView1->Dock = System::Windows::Forms::DockStyle::Fill;

this->listView1->Location = System::Drawing::Point(0, 24);

this->listView1->Name = L"listView1";

this->listView1->Size = System::Drawing::Size(384, 291);

this->listView1->TabIndex = 2;

this->listView1->UseCompatibleStateImageBehavior = false;

this->listView1->View = System::Windows::Forms::View::Details;

//

// columnHeader1

//

this->columnHeader1->Text = L"id";

//

// columnHeader2

//

this->columnHeader2->Text = L"Номер документа";

this->columnHeader2->Width = 108;

//

// columnHeader3

//

this->columnHeader3->Text = L"Наклодная";

this->columnHeader3->Width = 99;

//

// Documents

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(384, 315);

this->Controls->Add(this->listView1);

this->Controls->Add(this->menuStrip1);

this->MainMenuStrip = this->menuStrip1;

this->Name = L"Documents";

this->Text = L"Документы";

this->Load += gcnew System::EventHandler(this, &Documents::Documents\_Load);

this->menuStrip1->ResumeLayout(false);

this->menuStrip1->PerformLayout();

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

public: Document^ doc;

public: Magazin^ magz;

public: Magazin^ Document\_Shown(Magazin^ m) {

magz = m;

listView1->Items->Clear();

this->ShowDialog();

return magz;

}

private: System::Void CreateToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

AddDocument^ form = gcnew AddDocument();

magz = form->AddDocument\_Shown(magz);

this->UpdateListView();

magz->WriteToFile();

}

private: System::Void UpdateListView() {

listView1->Items->Clear();

for (int i = 0; i < this->magz->ArrayDocument->Count; i++) {

ListViewItem^ Id = gcnew ListViewItem();

Id->Text = this->magz->ArrayDocument[i]->id.ToString();

ListViewItem::ListViewSubItem^ Name = gcnew ListViewItem::ListViewSubItem();

Name->Text = this->magz->ArrayDocument[i]->NumberDogovor;

Id->SubItems->Add(Name);

ListViewItem::ListViewSubItem^ Adress = gcnew ListViewItem::ListViewSubItem();

Adress->Text = this->magz->ArrayDocument[i]->Naklodnaya;

Id->SubItems->Add(Adress);

listView1->Items->Add(Id);

}

}

private: System::Void Documents\_Load(System::Object^ sender, System::EventArgs^ e) {

this->UpdateListView();

}

private: System::Void EditToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

if (listView1->SelectedItems->Count > 0 &&

listView1->SelectedItems->Count < 2) {

EditdDocument^ form = gcnew EditdDocument();

magz = form->EditDocument\_Shown(this->magz, (int)Convert::ToInt32(listView1->SelectedItems[0]->SubItems[0]->Text));

this->UpdateListView();

magz->WriteToFile();

}

else if (listView1->SelectedItems->Count >= 2) {

MessageBox::Show("Выбрано много элементов для редактирования!!!", "Ошибка!", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

else {

MessageBox::Show("Не выбран элемент для редактирования!!!", "Ошибка!", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

private: System::Void RemoveToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

for (int i = 0; i < listView1->SelectedItems->Count; i++) {

int id = Convert::ToInt32(listView1->SelectedItems[i]->SubItems[0]->Text);

for (int i = 0; i < magz->ArrayDocument->Count; i++)

{

if (magz->ArrayDocument[i]->id == id)

{

magz->ArrayDocument->Remove(magz->ArrayDocument[i]);

}

}

}

this->UpdateListView();

magz->WriteToFile();

}

};

}

Consumptions.h

#pragma once

#include "AddConsumption.h"

#include "EditConsumption.h"

#include "Magazin.h"

#include "Consumption.h"

namespace Project1 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Сводка для Consumptions

/// </summary>

public ref class Consumptions : public System::Windows::Forms::Form

{

public:

Consumptions(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~Consumptions()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::ListView^ listView1;

protected:

private: System::Windows::Forms::ColumnHeader^ columnHeader1;

private: System::Windows::Forms::ColumnHeader^ columnHeader2;

private: System::Windows::Forms::ColumnHeader^ columnHeader3;

private: System::Windows::Forms::MenuStrip^ menuStrip1;

private: System::Windows::Forms::ToolStripMenuItem^ добавитьToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ редактироватьToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ удалитьToolStripMenuItem;

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->listView1 = (gcnew System::Windows::Forms::ListView());

this->columnHeader1 = (gcnew System::Windows::Forms::ColumnHeader());

this->columnHeader2 = (gcnew System::Windows::Forms::ColumnHeader());

this->columnHeader3 = (gcnew System::Windows::Forms::ColumnHeader());

this->menuStrip1 = (gcnew System::Windows::Forms::MenuStrip());

this->добавитьToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->редактироватьToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->удалитьToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->menuStrip1->SuspendLayout();

this->SuspendLayout();

//

// listView1

//

this->listView1->Columns->AddRange(gcnew cli::array< System::Windows::Forms::ColumnHeader^ >(3) {

this->columnHeader1, this->columnHeader2,

this->columnHeader3

});

this->listView1->Dock = System::Windows::Forms::DockStyle::Fill;

this->listView1->Location = System::Drawing::Point(0, 24);

this->listView1->Name = L"listView1";

this->listView1->Size = System::Drawing::Size(284, 241);

this->listView1->TabIndex = 3;

this->listView1->UseCompatibleStateImageBehavior = false;

this->listView1->View = System::Windows::Forms::View::Details;

//

// columnHeader1

//

this->columnHeader1->Text = L"id";

this->columnHeader1->Width = 30;

//

// columnHeader2

//

this->columnHeader2->Text = L"Продукт";

this->columnHeader2->Width = 120;

//

// columnHeader3

//

this->columnHeader3->Text = L"Колличество";

this->columnHeader3->Width = 80;

//

// menuStrip1

//

this->menuStrip1->Items->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(3) {

this->добавитьToolStripMenuItem,

this->редактироватьToolStripMenuItem, this->удалитьToolStripMenuItem

});

this->menuStrip1->Location = System::Drawing::Point(0, 0);

this->menuStrip1->Name = L"menuStrip1";

this->menuStrip1->Size = System::Drawing::Size(284, 24);

this->menuStrip1->TabIndex = 2;

this->menuStrip1->Text = L"menuStrip1";

//

// добавитьToolStripMenuItem

//

this->добавитьToolStripMenuItem->Name = L"добавитьToolStripMenuItem";

this->добавитьToolStripMenuItem->Size = System::Drawing::Size(69, 20);

this->добавитьToolStripMenuItem->Text = L"Добавить";

this->добавитьToolStripMenuItem->Click += gcnew System::EventHandler(this, &Consumptions::CreateToolStripMenuItem\_Click);

//

// редактироватьToolStripMenuItem

//

this->редактироватьToolStripMenuItem->Name = L"редактироватьToolStripMenuItem";

this->редактироватьToolStripMenuItem->Size = System::Drawing::Size(98, 20);

this->редактироватьToolStripMenuItem->Text = L"Редактировать";

this->редактироватьToolStripMenuItem->Click += gcnew System::EventHandler(this, &Consumptions::EditToolStripMenuItem\_Click);

//

// удалитьToolStripMenuItem

//

this->удалитьToolStripMenuItem->Name = L"удалитьToolStripMenuItem";

this->удалитьToolStripMenuItem->Size = System::Drawing::Size(63, 20);

this->удалитьToolStripMenuItem->Text = L"Удалить";

this->удалитьToolStripMenuItem->Click += gcnew System::EventHandler(this, &Consumptions::RemoveToolStripMenuItem\_Click);

//

// Consumptions

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(284, 265);

this->Controls->Add(this->listView1);

this->Controls->Add(this->menuStrip1);

this->Name = L"Consumptions";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"Consumptions";

this->Load += gcnew System::EventHandler(this, &Consumptions::Consumptions\_Load);

this->menuStrip1->ResumeLayout(false);

this->menuStrip1->PerformLayout();

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

private: Consumption^ consumption;

private: Magazin^ magaz;

public: Magazin^ Consumption\_Shown(Magazin^ magaz) {

if (magaz->ArrayComing->Count == 0) {

MessageBox::Show("Таблица прихода не должна быть пустой!", "Ошибка!", MessageBoxButtons::OK, MessageBoxIcon::Error);

return magaz;

}

this->magaz = magaz;

this->ShowDialog();

return this->magaz;

}

private: System::Void UpdateListView() {

listView1->Items->Clear();

for (int i = 0; i < this->magaz->ArrayConsumption->Count; i++) {

ListViewItem^ Id = gcnew ListViewItem();

Id->Text = this->magaz->ArrayConsumption[i]->id.ToString();

ListViewItem::ListViewSubItem^ Product = gcnew ListViewItem::ListViewSubItem();

Product->Text = this->magaz->getNameProductById(this->magaz->ArrayConsumption[i]->productId);

Id->SubItems->Add(Product);

ListViewItem::ListViewSubItem^ Count = gcnew ListViewItem::ListViewSubItem();

Count->Text = this->magaz->ArrayConsumption[i]->count.ToString();

Id->SubItems->Add(Count);

listView1->Items->Add(Id);

}

}

private: System::Void Consumptions\_Load(System::Object^ sender, System::EventArgs^ e) {

this->UpdateListView();

}

private: System::Void CreateToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

AddConsumption^ form = gcnew AddConsumption();

magaz = form->AddConsumption\_Shown(magaz);

this->UpdateListView();

magaz->WriteToFile();

}

private: System::Void EditToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

if (listView1->SelectedItems->Count > 0 &&

listView1->SelectedItems->Count < 2) {

EditConsumption^ form = gcnew EditConsumption();

magaz = form->EditConsumption\_Shown(this->magaz, (int)Convert::ToInt32(listView1->SelectedItems[0]->SubItems[0]->Text));

this->UpdateListView();

magaz->WriteToFile();

}

else if (listView1->SelectedItems->Count >= 2) {

MessageBox::Show("Выбрано много элементов для редактирования!!!", "Ошибка!", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

else {

MessageBox::Show("Не выбран элемент для редактирования!!!", "Ошибка!", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

private: System::Void RemoveToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

for (int i = 0; i < listView1->SelectedItems->Count; i++) {

int id = Convert::ToInt32(listView1->SelectedItems[i]->SubItems[0]->Text);

for (int i = 0; i < magaz->ArrayConsumption->Count; i++)

{

if (magaz->ArrayConsumption[i]->id == id)

{

magaz->ArrayConsumption->Remove(magaz->ArrayConsumption[i]);

}

}

}

this->UpdateListView();

magaz->WriteToFile();

}

};

}

Comings.h

#pragma once

#include "EditComing.h"

#include "AddComing.h"

#include "Coming.h"

#include "Magazin.h"

namespace Project1 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace System::IO;

/// <summary>

/// Сводка для Comings

/// </summary>

public ref class Comings : public System::Windows::Forms::Form

{

public:

Comings(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~Comings()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::MenuStrip^ menuStrip1;

private: System::Windows::Forms::ToolStripMenuItem^ добавитьToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ редактироватьToolStripMenuItem;

private: System::Windows::Forms::ContextMenuStrip^ contextMenuStrip1;

private: System::Windows::Forms::ToolStripMenuItem^ удалитьToolStripMenuItem;

private: System::Windows::Forms::ListView^ listView1;

private: System::Windows::Forms::ColumnHeader^ columnHeader1;

private: System::Windows::Forms::ColumnHeader^ columnHeader2;

private: System::Windows::Forms::ColumnHeader^ columnHeader3;

private: System::Windows::Forms::ColumnHeader^ columnHeader4;

private: System::Windows::Forms::ColumnHeader^ columnHeader5;

private: System::Windows::Forms::ColumnHeader^ columnHeader7;

private: System::ComponentModel::IContainer^ components;

protected:

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->components = (gcnew System::ComponentModel::Container());

this->menuStrip1 = (gcnew System::Windows::Forms::MenuStrip());

this->добавитьToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->редактироватьToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->удалитьToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->contextMenuStrip1 = (gcnew System::Windows::Forms::ContextMenuStrip(this->components));

this->listView1 = (gcnew System::Windows::Forms::ListView());

this->columnHeader1 = (gcnew System::Windows::Forms::ColumnHeader());

this->columnHeader2 = (gcnew System::Windows::Forms::ColumnHeader());

this->columnHeader3 = (gcnew System::Windows::Forms::ColumnHeader());

this->columnHeader4 = (gcnew System::Windows::Forms::ColumnHeader());

this->columnHeader5 = (gcnew System::Windows::Forms::ColumnHeader());

this->columnHeader7 = (gcnew System::Windows::Forms::ColumnHeader());

this->menuStrip1->SuspendLayout();

this->SuspendLayout();

//

// menuStrip1

//

this->menuStrip1->Items->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(3) {

this->добавитьToolStripMenuItem,

this->редактироватьToolStripMenuItem, this->удалитьToolStripMenuItem

});

this->menuStrip1->Location = System::Drawing::Point(0, 0);

this->menuStrip1->Name = L"menuStrip1";

this->menuStrip1->Size = System::Drawing::Size(429, 24);

this->menuStrip1->TabIndex = 0;

this->menuStrip1->Text = L"menuStrip1";

//

// добавитьToolStripMenuItem

//

this->добавитьToolStripMenuItem->Name = L"добавитьToolStripMenuItem";

this->добавитьToolStripMenuItem->Size = System::Drawing::Size(69, 20);

this->добавитьToolStripMenuItem->Text = L"Добавить";

this->добавитьToolStripMenuItem->Click += gcnew System::EventHandler(this, &Comings::CreateToolStripMenuItem\_Click);

//

// редактироватьToolStripMenuItem

//

this->редактироватьToolStripMenuItem->Name = L"редактироватьToolStripMenuItem";

this->редактироватьToolStripMenuItem->Size = System::Drawing::Size(98, 20);

this->редактироватьToolStripMenuItem->Text = L"Редактировать";

this->редактироватьToolStripMenuItem->Click += gcnew System::EventHandler(this, &Comings::EditToolStripMenuItem\_Click);

//

// удалитьToolStripMenuItem

//

this->удалитьToolStripMenuItem->Name = L"удалитьToolStripMenuItem";

this->удалитьToolStripMenuItem->Size = System::Drawing::Size(63, 20);

this->удалитьToolStripMenuItem->Text = L"Удалить";

this->удалитьToolStripMenuItem->Click += gcnew System::EventHandler(this, &Comings::RemoveToolStripMenuItem\_Click);

//

// contextMenuStrip1

//

this->contextMenuStrip1->Name = L"contextMenuStrip1";

this->contextMenuStrip1->Size = System::Drawing::Size(61, 4);

//

// listView1

//

this->listView1->Columns->AddRange(gcnew cli::array< System::Windows::Forms::ColumnHeader^ >(6) {

this->columnHeader1, this->columnHeader2,

this->columnHeader3, this->columnHeader4, this->columnHeader5, this->columnHeader7

});

this->listView1->Dock = System::Windows::Forms::DockStyle::Fill;

this->listView1->Location = System::Drawing::Point(0, 24);

this->listView1->Name = L"listView1";

this->listView1->Size = System::Drawing::Size(429, 291);

this->listView1->TabIndex = 2;

this->listView1->UseCompatibleStateImageBehavior = false;

this->listView1->View = System::Windows::Forms::View::Details;

//

// columnHeader1

//

this->columnHeader1->Text = L"id";

this->columnHeader1->Width = 26;

//

// columnHeader2

//

this->columnHeader2->Text = L"Товар";

//

// columnHeader3

//

this->columnHeader3->Text = L"Поставщик";

this->columnHeader3->Width = 77;

//

// columnHeader4

//

this->columnHeader4->Text = L"Документ";

this->columnHeader4->Width = 65;

//

// columnHeader5

//

this->columnHeader5->Text = L"Колличество";

this->columnHeader5->Width = 84;

//

// columnHeader7

//

this->columnHeader7->Text = L"Стоимость";

this->columnHeader7->Width = 71;

//

// Comings

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(429, 315);

this->Controls->Add(this->listView1);

this->Controls->Add(this->menuStrip1);

this->MainMenuStrip = this->menuStrip1;

this->Name = L"Comings";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"Comings";

this->Load += gcnew System::EventHandler(this, &Comings::Comings\_Load);

this->menuStrip1->ResumeLayout(false);

this->menuStrip1->PerformLayout();

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

private: Coming^ cm;

public: Magazin^ magz;

private: String^ pathDB = Application::StartupPath + "/db/";

private: String^ fileDBName = "magazins.json";

public: Magazin^ Coming\_Shown(Magazin^ magaz) {

if (magaz->ArrayDocument->Count == 0 ||

magaz->ArrayProduct->Count == 0 ||

magaz->ArrayProvider->Count == 0

) {

MessageBox::Show("Товары, Документы, Постащики не должны быть пустыми!", "Ошибка!", MessageBoxButtons::OK, MessageBoxIcon::Error);

return magaz;

}

magz = magaz;

this->ShowDialog();

return magz;

}

private: System::Void CreateToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

AddComing ^form = gcnew AddComing();

magz = form->AddComing\_Shown(magz);

this->UpdateListView();

magz->WriteToFile();

}

private: System::Void EditToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

if (listView1->SelectedItems->Count > 0 &&

listView1->SelectedItems->Count < 2) {

EditComing^ form = gcnew EditComing();

magz = form->EditComing\_Shown(this->magz, (int)Convert::ToInt32(listView1->SelectedItems[0]->SubItems[0]->Text));

this->UpdateListView();

magz->WriteToFile();

}

else if (listView1->SelectedItems->Count >= 2) {

MessageBox::Show("Выбрано много элементов для редактирования!!!", "Ошибка!", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

else {

MessageBox::Show("Не выбран элемент для редактирования!!!", "Ошибка!", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

private: System::Void RemoveToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

for (int i = 0; i < listView1->SelectedItems->Count; i++) {

int id = Convert::ToInt32(listView1->SelectedItems[i]->SubItems[0]->Text);

for (int i = 0; i < magz->ArrayComing->Count; i++)

{

if (magz->ArrayComing[i]->id == id)

{

magz->ArrayComing->Remove(magz->ArrayComing[i]);

}

}

}

this->UpdateListView();

magz->WriteToFile();

}

private: System::Void Comings\_Load(System::Object^ sender, System::EventArgs^ e) {

this->UpdateListView();

}

private: System::Void UpdateListView() {

listView1->Items->Clear();

for (int i = 0; i < this->magz->ArrayComing->Count; i++) {

ListViewItem^ Id = gcnew ListViewItem();

Id->Text = this->magz->ArrayComing[i]->id.ToString();

ListViewItem::ListViewSubItem^ Product = gcnew ListViewItem::ListViewSubItem();

Product->Text = this->magz->getNameProductById(this->magz->ArrayComing[i]->productId);

Id->SubItems->Add(Product);

ListViewItem::ListViewSubItem^ Provider = gcnew ListViewItem::ListViewSubItem();

Provider->Text = this->magz->getNameProviderById(this->magz->ArrayComing[i]->providerId);

Id->SubItems->Add(Provider);

ListViewItem::ListViewSubItem^ Document = gcnew ListViewItem::ListViewSubItem();

Document->Text = this->magz->getNumberDogovorById(this->magz->ArrayComing[i]->documentId);

Id->SubItems->Add(Document);

ListViewItem::ListViewSubItem^ Count = gcnew ListViewItem::ListViewSubItem();

Count->Text = this->magz->ArrayComing[i]->count.ToString();

Id->SubItems->Add(Count);

ListViewItem::ListViewSubItem^ Price = gcnew ListViewItem::ListViewSubItem();

Price->Text = this->magz->ArrayComing[i]->price.ToString();

Id->SubItems->Add(Price);

listView1->Items->Add(Id);

}

}

};

}

AddComing.h

#pragma once

#include "Magazin.h"

#include "Product.h"

#include "Provider.h"

#include "Document.h"

namespace Project1 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace Newtonsoft::Json;

using namespace Newtonsoft::Json::Linq;

/// <summary>

/// Сводка для AddComing

/// </summary>

public ref class AddComing : public System::Windows::Forms::Form

{

public:

AddComing(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~AddComing()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Button^ button1;

protected:

private: System::Windows::Forms::TextBox^ textBox3;

private: System::Windows::Forms::TextBox^ textBox1;

private: System::Windows::Forms::ComboBox^ comboBox3;

private: System::Windows::Forms::ComboBox^ comboBox2;

private: System::Windows::Forms::ComboBox^ comboBox1;

private: System::Windows::Forms::Label^ label6;

private: System::Windows::Forms::Label^ label4;

private: System::Windows::Forms::Label^ label3;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::Label^ label1;

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->button1 = (gcnew System::Windows::Forms::Button());

this->textBox3 = (gcnew System::Windows::Forms::TextBox());

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->comboBox3 = (gcnew System::Windows::Forms::ComboBox());

this->comboBox2 = (gcnew System::Windows::Forms::ComboBox());

this->comboBox1 = (gcnew System::Windows::Forms::ComboBox());

this->label6 = (gcnew System::Windows::Forms::Label());

this->label4 = (gcnew System::Windows::Forms::Label());

this->label3 = (gcnew System::Windows::Forms::Label());

this->label2 = (gcnew System::Windows::Forms::Label());

this->label1 = (gcnew System::Windows::Forms::Label());

this->SuspendLayout();

//

// button1

//

this->button1->Location = System::Drawing::Point(15, 163);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(75, 23);

this->button1->TabIndex = 25;

this->button1->Text = L"Добавить";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &AddComing::Button1\_Click);

//

// textBox3

//

this->textBox3->Location = System::Drawing::Point(93, 127);

this->textBox3->Name = L"textBox3";

this->textBox3->Size = System::Drawing::Size(179, 20);

this->textBox3->TabIndex = 24;

this->textBox3->TextChanged += gcnew System::EventHandler(this, &AddComing::TextBox3\_TextChanged);

//

// textBox1

//

this->textBox1->Location = System::Drawing::Point(93, 101);

this->textBox1->Name = L"textBox1";

this->textBox1->Size = System::Drawing::Size(179, 20);

this->textBox1->TabIndex = 22;

this->textBox1->TextChanged += gcnew System::EventHandler(this, &AddComing::TextBox1\_TextChanged);

//

// comboBox3

//

this->comboBox3->FormattingEnabled = true;

this->comboBox3->Location = System::Drawing::Point(93, 74);

this->comboBox3->Name = L"comboBox3";

this->comboBox3->Size = System::Drawing::Size(179, 21);

this->comboBox3->TabIndex = 21;

//

// comboBox2

//

this->comboBox2->FormattingEnabled = true;

this->comboBox2->Location = System::Drawing::Point(93, 47);

this->comboBox2->Name = L"comboBox2";

this->comboBox2->Size = System::Drawing::Size(179, 21);

this->comboBox2->TabIndex = 20;

//

// comboBox1

//

this->comboBox1->FormattingEnabled = true;

this->comboBox1->Location = System::Drawing::Point(93, 22);

this->comboBox1->Name = L"comboBox1";

this->comboBox1->Size = System::Drawing::Size(179, 21);

this->comboBox1->TabIndex = 19;

//

// label6

//

this->label6->AutoSize = true;

this->label6->Location = System::Drawing::Point(12, 130);

this->label6->Name = L"label6";

this->label6->Size = System::Drawing::Size(62, 13);

this->label6->TabIndex = 18;

this->label6->Text = L"Стоимость";

//

// label4

//

this->label4->AutoSize = true;

this->label4->Location = System::Drawing::Point(12, 104);

this->label4->Name = L"label4";

this->label4->Size = System::Drawing::Size(72, 13);

this->label4->TabIndex = 16;

this->label4->Text = L"Колличество";

//

// label3

//

this->label3->AutoSize = true;

this->label3->Location = System::Drawing::Point(12, 77);

this->label3->Name = L"label3";

this->label3->Size = System::Drawing::Size(66, 13);

this->label3->TabIndex = 15;

this->label3->Text = L"Документы";

//

// label2

//

this->label2->AutoSize = true;

this->label2->Location = System::Drawing::Point(12, 50);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(65, 13);

this->label2->TabIndex = 14;

this->label2->Text = L"Поставщик";

//

// label1

//

this->label1->AutoSize = true;

this->label1->Location = System::Drawing::Point(12, 25);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(38, 13);

this->label1->TabIndex = 13;

this->label1->Text = L"Товар";

//

// AddComing

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(284, 196);

this->Controls->Add(this->button1);

this->Controls->Add(this->textBox3);

this->Controls->Add(this->textBox1);

this->Controls->Add(this->comboBox3);

this->Controls->Add(this->comboBox2);

this->Controls->Add(this->comboBox1);

this->Controls->Add(this->label6);

this->Controls->Add(this->label4);

this->Controls->Add(this->label3);

this->Controls->Add(this->label2);

this->Controls->Add(this->label1);

this->Name = L"AddComing";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"AddComing";

this->Load += gcnew System::EventHandler(this, &AddComing::AddComing\_Load);

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

public: Coming^ coming;

public: Magazin^ magaz;

public: Magazin^ AddComing\_Shown(Magazin^ m) {

this->magaz = m;

//добавление элементов в combobox

this->ShowDialog();

return this->magaz;

}

private: System::Void Button1\_Click(System::Object^ sender, System::EventArgs^ e) {

if (textBox1->Text != "" &&

textBox3->Text != "")

{

int id = 1;

if (magaz->ArrayComing->Count != 0) {

id = (int)magaz->ArrayComing[magaz->ArrayComing->Count - 1]->id;

id = id + 1;

}

coming = gcnew Coming();

coming->id = id;

coming->productId = this->magaz->getIdProductByName(comboBox1->SelectedItem->ToString());

coming->providerId = this->magaz->getIdProviderByName(comboBox2->SelectedItem->ToString());

coming->documentId = this->magaz->getIdDocumentByNumberDogovor(comboBox3->SelectedItem->ToString());

coming->count = Convert::ToInt32(textBox1->Text);

coming->price = Convert::ToInt32(textBox3->Text);

magaz->ArrayComing->Add(coming);

Console::WriteLine(JsonConvert::SerializeObject(magaz));

this->Close();

}

else {

MessageBox::Show("Неправильно введены данные", "Ошибка!", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

private: System::Void TextBox3\_TextChanged(System::Object^ sender, System::EventArgs^ e) {

try

{

unsigned long number = Convert::ToInt32(textBox3 ->Text);

}

catch (FormatException^ e)

{

MessageBox::Show("В поле стоимость допусщена ошибка", "Ошибка!", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

private: System::Void TextBox1\_TextChanged(System::Object^ sender, System::EventArgs^ e) {

try

{

unsigned long number = Convert::ToInt32(textBox1->Text);

}

catch (FormatException^ e)

{

MessageBox::Show("В поле колличество допусщена ошибка", "Ошибка!", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

private: System::Void AddComing\_Load(System::Object^ sender, System::EventArgs^ e) {

for each (Product ^ item in this->magaz->ArrayProduct)

{

comboBox1->Items->Add(item->Name);

comboBox1->SelectedIndex = 0;

}

for each (Provider ^ item in this->magaz->ArrayProvider)

{

comboBox2->Items->Add(item->Name);

comboBox2->SelectedIndex = 0;

}

for each (Document ^ item in this->magaz->ArrayDocument)

{

comboBox3->Items->Add(item->NumberDogovor);

comboBox3->SelectedIndex = 0;

}

}

};

}

AddConsumption.h

#pragma once

#include "Magazin.h"

#include "Consumption.h"

namespace Project1 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Сводка для AddConsumption

/// </summary>

public ref class AddConsumption : public System::Windows::Forms::Form

{

public:

AddConsumption(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~AddConsumption()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Button^ button1;

protected:

private: System::Windows::Forms::TextBox^ textBox1;

private: System::Windows::Forms::ComboBox^ comboBox1;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::Label^ label1;

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->button1 = (gcnew System::Windows::Forms::Button());

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->comboBox1 = (gcnew System::Windows::Forms::ComboBox());

this->label2 = (gcnew System::Windows::Forms::Label());

this->label1 = (gcnew System::Windows::Forms::Label());

this->SuspendLayout();

//

// button1

//

this->button1->Location = System::Drawing::Point(11, 91);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(75, 23);

this->button1->TabIndex = 9;

this->button1->Text = L"Добавить";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &AddConsumption::Button1\_Click);

//

// textBox1

//

this->textBox1->Location = System::Drawing::Point(96, 53);

this->textBox1->Name = L"textBox1";

this->textBox1->Size = System::Drawing::Size(178, 20);

this->textBox1->TabIndex = 8;

this->textBox1->TextChanged += gcnew System::EventHandler(this, &AddConsumption::TextBox1\_TextChanged);

//

// comboBox1

//

this->comboBox1->FormattingEnabled = true;

this->comboBox1->Location = System::Drawing::Point(96, 17);

this->comboBox1->Name = L"comboBox1";

this->comboBox1->Size = System::Drawing::Size(178, 21);

this->comboBox1->TabIndex = 7;

//

// label2

//

this->label2->AutoSize = true;

this->label2->Location = System::Drawing::Point(14, 56);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(72, 13);

this->label2->TabIndex = 6;

this->label2->Text = L"Колличество";

//

// label1

//

this->label1->AutoSize = true;

this->label1->Location = System::Drawing::Point(14, 23);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(38, 13);

this->label1->TabIndex = 5;

this->label1->Text = L"Товар";

//

// AddConsumption

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(284, 131);

this->Controls->Add(this->button1);

this->Controls->Add(this->textBox1);

this->Controls->Add(this->comboBox1);

this->Controls->Add(this->label2);

this->Controls->Add(this->label1);

this->MaximumSize = System::Drawing::Size(300, 166);

this->MinimumSize = System::Drawing::Size(300, 166);

this->Name = L"AddConsumption";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"AddConsumption";

this->Load += gcnew System::EventHandler(this, &AddConsumption::AddConsumption\_Load);

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

private: Magazin^ magaz;

private: Consumption^ consumption;

public: Magazin^ AddConsumption\_Shown(Magazin^ magaz) {

this->magaz = magaz;

this->ShowDialog();

return this->magaz;

}

private: System::Void AddConsumption\_Load(System::Object^ sender, System::EventArgs^ e) {

for each (Product ^ item in this->magaz->ArrayProduct)

{

comboBox1->Items->Add(item->Name);

comboBox1->SelectedIndex = 0;

}

}

private: System::Void Button1\_Click(System::Object^ sender, System::EventArgs^ e) {

if (textBox1->Text != "")

{

int id = 1;

if (magaz->ArrayConsumption->Count != 0) {

id = (int)magaz->ArrayConsumption[magaz->ArrayConsumption->Count - 1]->id;

id = id + 1;

}

consumption = gcnew Consumption();

consumption->id = id;

consumption->productId = magaz->getIdProductByName(comboBox1->SelectedItem->ToString());

consumption->count = Convert::ToInt32(textBox1->Text);

magaz->ArrayConsumption->Add(consumption);

this->Close();

}

else {

MessageBox::Show("Неправильно введены данные", "Ошибка!", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

private: System::Void TextBox1\_TextChanged(System::Object^ sender, System::EventArgs^ e) {

try

{

unsigned long number = Convert::ToInt32(textBox1->Text);

}

catch (FormatException^ e)

{

MessageBox::Show("В поле колличество допусщена ошибка", "Ошибка!", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

};

}

AddDocument.h

#pragma once

#include "Document.h"

#include "Magazin.h"

namespace Project1 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Сводка для AddDocument

/// </summary>

public ref class AddDocument : public System::Windows::Forms::Form

{

public:

AddDocument(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~AddDocument()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Label^ label1;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::TextBox^ textBox1;

private: System::Windows::Forms::TextBox^ textBox2;

private: System::Windows::Forms::Button^ button1;

protected:

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->label1 = (gcnew System::Windows::Forms::Label());

this->label2 = (gcnew System::Windows::Forms::Label());

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->textBox2 = (gcnew System::Windows::Forms::TextBox());

this->button1 = (gcnew System::Windows::Forms::Button());

this->SuspendLayout();

//

// label1

//

this->label1->AutoSize = true;

this->label1->Location = System::Drawing::Point(12, 25);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(91, 13);

this->label1->TabIndex = 0;

this->label1->Text = L"Номер договора";

//

// label2

//

this->label2->AutoSize = true;

this->label2->Location = System::Drawing::Point(12, 51);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(98, 13);

this->label2->TabIndex = 1;

this->label2->Text = L"Номер накладной";

//

// textBox1

//

this->textBox1->Location = System::Drawing::Point(116, 22);

this->textBox1->Name = L"textBox1";

this->textBox1->Size = System::Drawing::Size(156, 20);

this->textBox1->TabIndex = 2;

//

// textBox2

//

this->textBox2->Location = System::Drawing::Point(116, 48);

this->textBox2->Name = L"textBox2";

this->textBox2->Size = System::Drawing::Size(156, 20);

this->textBox2->TabIndex = 3;

//

// button1

//

this->button1->Location = System::Drawing::Point(101, 92);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(75, 23);

this->button1->TabIndex = 4;

this->button1->Text = L"Добавить";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &AddDocument::Button1\_Click);

//

// AddDocument

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(284, 133);

this->Controls->Add(this->button1);

this->Controls->Add(this->textBox2);

this->Controls->Add(this->textBox1);

this->Controls->Add(this->label2);

this->Controls->Add(this->label1);

this->Name = L"AddDocument";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"Добавление документов";

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

public: Document^ doc;

public: Magazin^ magz;

public: Magazin^ AddDocument\_Shown(Magazin^ m) {

this->magz = m;

this->ShowDialog();

return this->magz;

}

private: System::Void Button1\_Click(System::Object^ sender, System::EventArgs^ e) {

int id = 1;

if (textBox1->Text != "" &&

textBox2->Text != "" ) {

if (magz->ArrayDocument->Count != 0) {

id = (int)magz->ArrayDocument[magz->ArrayDocument->Count - 1]->id;

id = id + 1;

}

doc = gcnew Document();

doc->id = id;

doc->NumberDogovor = textBox1->Text;

doc->Naklodnaya = textBox2->Text;

magz->ArrayDocument->Add(doc);

this->Close();

}

else {

MessageBox::Show("Неправильно введены данные", "Ошибка!", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

};

}

AddProduct.h

#pragma once

#include "Product.h"

#include "Magazin.h"

namespace Project1 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Сводка для AddProduct

/// </summary>

public ref class AddProduct : public System::Windows::Forms::Form

{

public:

AddProduct(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~AddProduct()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::TextBox^ textBox1;

private: System::Windows::Forms::Label^ label1;

private: System::Windows::Forms::Button^ button1;

protected:

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container^ components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->label1 = (gcnew System::Windows::Forms::Label());

this->button1 = (gcnew System::Windows::Forms::Button());

this->SuspendLayout();

//

// textBox1

//

this->textBox1->Location = System::Drawing::Point(126, 12);

this->textBox1->Name = L"textBox1";

this->textBox1->Size = System::Drawing::Size(146, 20);

this->textBox1->TabIndex = 0;

//

// label1

//

this->label1->AutoSize = true;

this->label1->Location = System::Drawing::Point(12, 15);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(83, 13);

this->label1->TabIndex = 4;

this->label1->Text = L"Наименование";

//

// button1

//

this->button1->Location = System::Drawing::Point(12, 48);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(75, 23);

this->button1->TabIndex = 8;

this->button1->Text = L"Добавить";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &AddProduct::Button1\_Click);

//

// AddProduct

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(284, 85);

this->Controls->Add(this->button1);

this->Controls->Add(this->label1);

this->Controls->Add(this->textBox1);

this->Name = L"AddProduct";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"AddProduct";

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

public: Product^ pd;

public: Magazin^ magz;

public: Magazin^ AddProduct\_Shown(Magazin^ arr) {

this->magz = arr;

this->ShowDialog();

//сделать проверку на cуществующий элемент

return this->magz;

}

private: System::Void Button1\_Click(System::Object^ sender, System::EventArgs^ e) {

int id = 1;

if (textBox1->Text != "") {

if (magz->ArrayProduct->Count != 0) {

id = (int)magz->ArrayProduct[magz->ArrayProduct->Count - 1]->id;

id = id + 1;

}

//сделать проверку на введенные данные

pd = gcnew Product();

pd->id = id;

pd->Name = textBox1->Text;

magz->ArrayProduct->Add(pd);

this->Close();

}

else {

MessageBox::Show("Неправильно введены данные", "Ошибка!", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

};

}

AddProvider.h

#pragma once

#include "Provider.h"

#include "Magazin.h"

namespace Project1 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Сводка для AddProvider

/// </summary>

public ref class AddProvider : public System::Windows::Forms::Form

{

public:

AddProvider(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~AddProvider()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Button^ button1;

protected:

private: System::Windows::Forms::Label^ label3;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::Label^ label1;

private: System::Windows::Forms::TextBox^ textBox3;

private: System::Windows::Forms::TextBox^ textBox2;

private: System::Windows::Forms::TextBox^ textBox1;

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->button1 = (gcnew System::Windows::Forms::Button());

this->label3 = (gcnew System::Windows::Forms::Label());

this->label2 = (gcnew System::Windows::Forms::Label());

this->label1 = (gcnew System::Windows::Forms::Label());

this->textBox3 = (gcnew System::Windows::Forms::TextBox());

this->textBox2 = (gcnew System::Windows::Forms::TextBox());

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->SuspendLayout();

//

// button1

//

this->button1->Location = System::Drawing::Point(15, 115);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(75, 23);

this->button1->TabIndex = 13;

this->button1->Text = L" Добавить";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &AddProvider::Button1\_Click);

//

// label3

//

this->label3->AutoSize = true;

this->label3->Location = System::Drawing::Point(12, 84);

this->label3->Name = L"label3";

this->label3->Size = System::Drawing::Size(77, 13);

this->label3->TabIndex = 12;

this->label3->Text = L"Комментарий";

//

// label2

//

this->label2->AutoSize = true;

this->label2->Location = System::Drawing::Point(12, 58);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(38, 13);

this->label2->TabIndex = 11;

this->label2->Text = L"Адрес";

//

// label1

//

this->label1->AutoSize = true;

this->label1->Location = System::Drawing::Point(12, 29);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(83, 13);

this->label1->TabIndex = 10;

this->label1->Text = L"Наименование";

//

// textBox3

//

this->textBox3->Location = System::Drawing::Point(107, 81);

this->textBox3->Name = L"textBox3";

this->textBox3->Size = System::Drawing::Size(147, 20);

this->textBox3->TabIndex = 9;

//

// textBox2

//

this->textBox2->Location = System::Drawing::Point(107, 55);

this->textBox2->Name = L"textBox2";

this->textBox2->Size = System::Drawing::Size(147, 20);

this->textBox2->TabIndex = 8;

//

// textBox1

//

this->textBox1->Location = System::Drawing::Point(107, 29);

this->textBox1->Name = L"textBox1";

this->textBox1->Size = System::Drawing::Size(147, 20);

this->textBox1->TabIndex = 7;

//

// AddProvider

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(284, 161);

this->Controls->Add(this->button1);

this->Controls->Add(this->label3);

this->Controls->Add(this->label2);

this->Controls->Add(this->label1);

this->Controls->Add(this->textBox3);

this->Controls->Add(this->textBox2);

this->Controls->Add(this->textBox1);

this->Name = L"AddProvider";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"AddProvider";

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

public: Provider^ Prov;

public: Magazin^ magz;

public: Magazin^ AddProvider\_Shown(Magazin^ arr) {

this->magz = arr;

this->ShowDialog();

return this->magz;

}

private: System::Void Button1\_Click(System::Object^ sender, System::EventArgs^ e) {

int id = 1;

if (textBox1->Text != "" &&

textBox2->Text != "" &&

textBox3->Text != ""

) {

if (magz->ArrayProvider->Count != 0) {

id = (int)magz->ArrayProvider[magz->ArrayProvider->Count - 1]->id;

id = id + 1;

}

Prov = gcnew Provider();

Prov->id = id;

Prov->Name = textBox1->Text;

Prov->Addres = textBox2->Text;

Prov->Comment = textBox3->Text;

magz->ArrayProvider->Add(Prov);

this->Close();

}

else {

MessageBox::Show("Неправильно введены данные", "Ошибка!", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

};

}

EditComing.h

#pragma once

#include "Coming.h"

#include "Magazin.h"

namespace Project1 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Сводка для EditComing

/// </summary>

public ref class EditComing : public System::Windows::Forms::Form

{

public:

EditComing(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~EditComing()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Label^ label1;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::Label^ label3;

private: System::Windows::Forms::Label^ label4;

private: System::Windows::Forms::Label^ label6;

private: System::Windows::Forms::ComboBox^ comboBox1;

private: System::Windows::Forms::ComboBox^ comboBox2;

private: System::Windows::Forms::ComboBox^ comboBox3;

private: System::Windows::Forms::TextBox^ textBox1;

private: System::Windows::Forms::TextBox^ textBox3;

private: System::Windows::Forms::Button^ button1;

protected:

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->label1 = (gcnew System::Windows::Forms::Label());

this->label2 = (gcnew System::Windows::Forms::Label());

this->label3 = (gcnew System::Windows::Forms::Label());

this->label4 = (gcnew System::Windows::Forms::Label());

this->label6 = (gcnew System::Windows::Forms::Label());

this->comboBox1 = (gcnew System::Windows::Forms::ComboBox());

this->comboBox2 = (gcnew System::Windows::Forms::ComboBox());

this->comboBox3 = (gcnew System::Windows::Forms::ComboBox());

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->textBox3 = (gcnew System::Windows::Forms::TextBox());

this->button1 = (gcnew System::Windows::Forms::Button());

this->SuspendLayout();

//

// label1

//

this->label1->AutoSize = true;

this->label1->Location = System::Drawing::Point(12, 28);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(38, 13);

this->label1->TabIndex = 0;

this->label1->Text = L"Товар";

//

// label2

//

this->label2->AutoSize = true;

this->label2->Location = System::Drawing::Point(12, 53);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(65, 13);

this->label2->TabIndex = 1;

this->label2->Text = L"Поставщик";

//

// label3

//

this->label3->AutoSize = true;

this->label3->Location = System::Drawing::Point(12, 80);

this->label3->Name = L"label3";

this->label3->Size = System::Drawing::Size(66, 13);

this->label3->TabIndex = 2;

this->label3->Text = L"Документы";

//

// label4

//

this->label4->AutoSize = true;

this->label4->Location = System::Drawing::Point(12, 107);

this->label4->Name = L"label4";

this->label4->Size = System::Drawing::Size(72, 13);

this->label4->TabIndex = 3;

this->label4->Text = L"Колличество";

//

// label6

//

this->label6->AutoSize = true;

this->label6->Location = System::Drawing::Point(12, 133);

this->label6->Name = L"label6";

this->label6->Size = System::Drawing::Size(62, 13);

this->label6->TabIndex = 5;

this->label6->Text = L"Стоимость";

//

// comboBox1

//

this->comboBox1->FormattingEnabled = true;

this->comboBox1->Location = System::Drawing::Point(93, 25);

this->comboBox1->Name = L"comboBox1";

this->comboBox1->Size = System::Drawing::Size(179, 21);

this->comboBox1->TabIndex = 6;

//

// comboBox2

//

this->comboBox2->FormattingEnabled = true;

this->comboBox2->Location = System::Drawing::Point(93, 50);

this->comboBox2->Name = L"comboBox2";

this->comboBox2->Size = System::Drawing::Size(179, 21);

this->comboBox2->TabIndex = 7;

//

// comboBox3

//

this->comboBox3->FormattingEnabled = true;

this->comboBox3->Location = System::Drawing::Point(93, 77);

this->comboBox3->Name = L"comboBox3";

this->comboBox3->Size = System::Drawing::Size(179, 21);

this->comboBox3->TabIndex = 8;

//

// textBox1

//

this->textBox1->Location = System::Drawing::Point(93, 104);

this->textBox1->Name = L"textBox1";

this->textBox1->Size = System::Drawing::Size(179, 20);

this->textBox1->TabIndex = 9;

this->textBox1->TextChanged += gcnew System::EventHandler(this, &EditComing::TextBox1\_TextChanged);

//

// textBox3

//

this->textBox3->Location = System::Drawing::Point(93, 130);

this->textBox3->Name = L"textBox3";

this->textBox3->Size = System::Drawing::Size(179, 20);

this->textBox3->TabIndex = 11;

this->textBox3->TextChanged += gcnew System::EventHandler(this, &EditComing::TextBox3\_TextChanged);

//

// button1

//

this->button1->Location = System::Drawing::Point(15, 171);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(75, 23);

this->button1->TabIndex = 12;

this->button1->Text = L"Сохранить";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &EditComing::Button1\_Click);

//

// EditComing

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(284, 215);

this->Controls->Add(this->button1);

this->Controls->Add(this->textBox3);

this->Controls->Add(this->textBox1);

this->Controls->Add(this->comboBox3);

this->Controls->Add(this->comboBox2);

this->Controls->Add(this->comboBox1);

this->Controls->Add(this->label6);

this->Controls->Add(this->label4);

this->Controls->Add(this->label3);

this->Controls->Add(this->label2);

this->Controls->Add(this->label1);

this->Name = L"EditComing";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"EditComing";

this->Load += gcnew System::EventHandler(this, &EditComing::EditComing\_Load);

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

public: Coming^ coming;

public: Magazin^ magaz;

public: Magazin^ EditComing\_Shown(Magazin^ magaz, int id) {

this->magaz = magaz;

for (int i = 0; i < magaz->ArrayComing->Count; i++)

{

if (magaz->ArrayComing[i]->id == id) {

coming = magaz->ArrayComing[i];

break;

}

}

if (coming != nullptr) {

this->ShowDialog();

for (int i = 0; i < magaz->ArrayComing->Count; i++)

{

if (magaz->ArrayComing[i]->id == id) {

magaz->ArrayComing[i] = coming;

break;

}

}

}

return this->magaz;

}

private: System::Void EditComing\_Load(System::Object^ sender, System::EventArgs^ e) {

for each (Product ^ item in this->magaz->ArrayProduct)

{

comboBox1->Items->Add(item->Name);

if (item->id == coming->productId) {

comboBox1->SelectedIndex = this->magaz->ArrayProduct->IndexOf(item);

}

}

for each (Provider ^ item in this->magaz->ArrayProvider)

{

comboBox2->Items->Add(item->Name);

if (item->id == coming->providerId) {

comboBox2->SelectedIndex = this->magaz->ArrayProvider->IndexOf(item);

}

}

for each (Document ^ item in this->magaz->ArrayDocument)

{

comboBox3->Items->Add(item->NumberDogovor);

if (item->id == coming->documentId) {

comboBox3->SelectedIndex = this->magaz->ArrayDocument->IndexOf(item);

}

}

textBox1->Text = coming->count.ToString();

textBox3->Text = coming->price.ToString();

}

private: System::Void Button1\_Click(System::Object^ sender, System::EventArgs^ e) {

if (textBox1->Text != "" &&

textBox3->Text != "")

{

coming->productId = this->magaz->getIdProductByName(comboBox1->SelectedItem->ToString());

coming->providerId = this->magaz->getIdProviderByName(comboBox2->SelectedItem->ToString());

coming->documentId = this->magaz->getIdDocumentByNumberDogovor(comboBox3->SelectedItem->ToString());

coming->count = Convert::ToInt32(textBox1->Text);

coming->price = Convert::ToInt32(textBox3->Text);

}

this->Close();

}

private: System::Void TextBox1\_TextChanged(System::Object^ sender, System::EventArgs^ e) {

try

{

unsigned long number = Convert::ToInt32(textBox1->Text);

}

catch (FormatException^ e)

{

MessageBox::Show("В поле колличество допусщена ошибка", "Ошибка!", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

private: System::Void TextBox3\_TextChanged(System::Object^ sender, System::EventArgs^ e) {

try

{

unsigned long number = Convert::ToInt32(textBox3->Text);

}

catch (FormatException^ e)

{

MessageBox::Show("В поле стоимость допусщена ошибка", "Ошибка!", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

};

}

EditConsumption.h

#pragma once

#include "Magazin.h"

#include "Consumption.h"

namespace Project1 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Сводка для EditConsumption

/// </summary>

public ref class EditConsumption : public System::Windows::Forms::Form

{

public:

EditConsumption(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~EditConsumption()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Button^ button1;

protected:

private: System::Windows::Forms::TextBox^ textBox1;

private: System::Windows::Forms::ComboBox^ comboBox1;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::Label^ label1;

protected:

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->button1 = (gcnew System::Windows::Forms::Button());

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->comboBox1 = (gcnew System::Windows::Forms::ComboBox());

this->label2 = (gcnew System::Windows::Forms::Label());

this->label1 = (gcnew System::Windows::Forms::Label());

this->SuspendLayout();

//

// button1

//

this->button1->Location = System::Drawing::Point(11, 91);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(75, 23);

this->button1->TabIndex = 9;

this->button1->Text = L"Сохранить";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &EditConsumption::Button1\_Click);

//

// textBox1

//

this->textBox1->Location = System::Drawing::Point(96, 53);

this->textBox1->Name = L"textBox1";

this->textBox1->Size = System::Drawing::Size(178, 20);

this->textBox1->TabIndex = 8;

this->textBox1->TextChanged += gcnew System::EventHandler(this, &EditConsumption::TextBox1\_TextChanged);

//

// comboBox1

//

this->comboBox1->FormattingEnabled = true;

this->comboBox1->Location = System::Drawing::Point(96, 17);

this->comboBox1->Name = L"comboBox1";

this->comboBox1->Size = System::Drawing::Size(178, 21);

this->comboBox1->TabIndex = 7;

//

// label2

//

this->label2->AutoSize = true;

this->label2->Location = System::Drawing::Point(14, 56);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(72, 13);

this->label2->TabIndex = 6;

this->label2->Text = L"Колличество";

//

// label1

//

this->label1->AutoSize = true;

this->label1->Location = System::Drawing::Point(14, 23);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(38, 13);

this->label1->TabIndex = 5;

this->label1->Text = L"Товар";

//

// EditConsumption

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(284, 131);

this->Controls->Add(this->button1);

this->Controls->Add(this->textBox1);

this->Controls->Add(this->comboBox1);

this->Controls->Add(this->label2);

this->Controls->Add(this->label1);

this->MaximumSize = System::Drawing::Size(300, 166);

this->MinimumSize = System::Drawing::Size(300, 166);

this->Name = L"EditConsumption";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"EditConsumption";

this->Load += gcnew System::EventHandler(this, &EditConsumption::EditConsumption\_Load);

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

private: Magazin^ magaz;

private: Consumption^ consumption;

public: Magazin^ EditConsumption\_Shown(Magazin^ magaz, int id ) {

this->magaz = magaz;

for (int i = 0; i < magaz->ArrayComing->Count; i++)

{

if (magaz->ArrayComing[i]->id == id) {

consumption = magaz->ArrayConsumption[i];

break;

}

}

if (consumption != nullptr) {

this->ShowDialog();

for (int i = 0; i < magaz->ArrayConsumption->Count; i++)

{

if (magaz->ArrayConsumption[i]->id == id) {

magaz->ArrayConsumption[i] = consumption;

break;

}

}

}

return this->magaz;

}

private: System::Void EditConsumption\_Load(System::Object^ sender, System::EventArgs^ e) {

for each (Product ^ item in this->magaz->ArrayProduct)

{

comboBox1->Items->Add(item->Name);

if (item->id == consumption->productId) {

comboBox1->SelectedIndex = this->magaz->ArrayProduct->IndexOf(item);

}

}

textBox1->Text = consumption->count.ToString();

}

private: System::Void Button1\_Click(System::Object^ sender, System::EventArgs^ e) {

if (textBox1->Text != "")

{

consumption->productId = this->magaz->getIdProductByName(comboBox1->SelectedItem->ToString());

consumption->count = Convert::ToInt32(textBox1->Text);

}

this->Close();

}

private: System::Void TextBox1\_TextChanged(System::Object^ sender, System::EventArgs^ e) {

try

{

unsigned long number = Convert::ToInt32(textBox1->Text);

}

catch (FormatException^ e)

{

MessageBox::Show("В поле колличество допусщена ошибка", "Ошибка!", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

};

}

EditProduct.h

#pragma once

#include "Product.h"

namespace Project1 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Сводка для EditProduct

/// </summary>

public ref class EditProduct : public System::Windows::Forms::Form

{

public:

EditProduct(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~EditProduct()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Button^ button1;

protected:

private: System::Windows::Forms::Label^ label1;

private: System::Windows::Forms::TextBox^ textBox1;

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->button1 = (gcnew System::Windows::Forms::Button());

this->label1 = (gcnew System::Windows::Forms::Label());

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->SuspendLayout();

//

// button1

//

this->button1->Location = System::Drawing::Point(12, 51);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(75, 23);

this->button1->TabIndex = 17;

this->button1->Text = L"Сохранить";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &EditProduct::Button1\_Click);

//

// label1

//

this->label1->AutoSize = true;

this->label1->Location = System::Drawing::Point(12, 21);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(83, 13);

this->label1->TabIndex = 13;

this->label1->Text = L"Наименование";

//

// textBox1

//

this->textBox1->Location = System::Drawing::Point(126, 18);

this->textBox1->Name = L"textBox1";

this->textBox1->Size = System::Drawing::Size(146, 20);

this->textBox1->TabIndex = 9;

//

// EditProduct

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(284, 89);

this->Controls->Add(this->button1);

this->Controls->Add(this->label1);

this->Controls->Add(this->textBox1);

this->Name = L"EditProduct";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"EditProduct";

this->Load += gcnew System::EventHandler(this, &EditProduct::EditProduct\_Load);

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

public: Product^ pd;

public: Magazin^ EditProduct\_Shown(Magazin^ magz, int id) {

for (int i = 0; i < magz->ArrayProduct->Count; i++)

{

if (magz->ArrayProduct[i]->id == id)

{

pd = magz->ArrayProduct[i];

break;

}

}

if (pd != nullptr)

{

this->ShowDialog();

for (int i = 0; i < magz->ArrayProduct->Count; i++)

{

if (magz->ArrayProduct[i]->id == id)

{

magz->ArrayProduct[i] = pd;

break;

}

}

}

return magz;

}

private: System::Void EditProduct\_Load(System::Object^ sender, System::EventArgs^ e) {

textBox1->Text = pd->Name;

}

private: System::Void Button1\_Click(System::Object^ sender, System::EventArgs^ e) {

//сделать проверку введенных данных

pd->Name = textBox1->Text;

this->Close();

}

};

}

EditProvider.h

#pragma once

#include "Provider.h"

#include "Magazin.h"

namespace Project1 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Сводка для EditProvider

/// </summary>

public ref class EditProvider : public System::Windows::Forms::Form

{

public:

EditProvider(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~EditProvider()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Button^ button1;

protected:

private: System::Windows::Forms::Label^ label3;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::Label^ label1;

private: System::Windows::Forms::TextBox^ textBox3;

private: System::Windows::Forms::TextBox^ textBox2;

private: System::Windows::Forms::TextBox^ textBox1;

protected:

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container^ components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->button1 = (gcnew System::Windows::Forms::Button());

this->label3 = (gcnew System::Windows::Forms::Label());

this->label2 = (gcnew System::Windows::Forms::Label());

this->label1 = (gcnew System::Windows::Forms::Label());

this->textBox3 = (gcnew System::Windows::Forms::TextBox());

this->textBox2 = (gcnew System::Windows::Forms::TextBox());

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->SuspendLayout();

//

// button1

//

this->button1->Location = System::Drawing::Point(24, 114);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(75, 23);

this->button1->TabIndex = 20;

this->button1->Text = L"Сохранить";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &EditProvider::Button1\_Click);

//

// label3

//

this->label3->AutoSize = true;

this->label3->Location = System::Drawing::Point(21, 83);

this->label3->Name = L"label3";

this->label3->Size = System::Drawing::Size(77, 13);

this->label3->TabIndex = 19;

this->label3->Text = L"Комментарий";

//

// label2

//

this->label2->AutoSize = true;

this->label2->Location = System::Drawing::Point(21, 57);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(38, 13);

this->label2->TabIndex = 18;

this->label2->Text = L"Адрес";

//

// label1

//

this->label1->AutoSize = true;

this->label1->Location = System::Drawing::Point(21, 28);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(83, 13);

this->label1->TabIndex = 17;

this->label1->Text = L"Наименование";

//

// textBox3

//

this->textBox3->Location = System::Drawing::Point(116, 80);

this->textBox3->Name = L"textBox3";

this->textBox3->Size = System::Drawing::Size(147, 20);

this->textBox3->TabIndex = 16;

//

// textBox2

//

this->textBox2->Location = System::Drawing::Point(116, 54);

this->textBox2->Name = L"textBox2";

this->textBox2->Size = System::Drawing::Size(147, 20);

this->textBox2->TabIndex = 15;

//

// textBox1

//

this->textBox1->Location = System::Drawing::Point(116, 28);

this->textBox1->Name = L"textBox1";

this->textBox1->Size = System::Drawing::Size(147, 20);

this->textBox1->TabIndex = 14;

//

// EditProvider

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(284, 165);

this->Controls->Add(this->button1);

this->Controls->Add(this->label3);

this->Controls->Add(this->label2);

this->Controls->Add(this->label1);

this->Controls->Add(this->textBox3);

this->Controls->Add(this->textBox2);

this->Controls->Add(this->textBox1);

this->Name = L"EditProvider";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"EditProvider";

this->Load += gcnew System::EventHandler(this, &EditProvider::EditProvider\_Load);

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

public: Provider^ Prov;

public: Magazin^ EditProvider\_Shown(Magazin^ magz, int id) {

for (int i = 0; i < magz->ArrayProvider->Count; i++)

{

if (magz->ArrayProvider[i]->id == id)

{

Prov = magz->ArrayProvider[i];

break;

}

}

if (Prov != nullptr)

{

this->ShowDialog();

for (int i = 0; i < magz->ArrayProvider->Count; i++)

{

if (magz->ArrayProvider[i]->id == id)

{

magz->ArrayProvider[i] = Prov;

break;

}

}

}

return magz;

}

private: System::Void EditProvider\_Load(System::Object^ sender, System::EventArgs^ e) {

textBox1->Text = Prov->Name;

textBox2->Text = Prov->Addres;

textBox3->Text = Prov->Comment;

}

private: System::Void Button1\_Click(System::Object^ sender, System::EventArgs^ e) {

Prov->Name = textBox1->Text;

Prov->Addres = textBox2->Text;

Prov->Comment = textBox3->Text;

this->Close();

}

};

}

EditDocument.h

#pragma once

#include "Document.h"

#include "Magazin.h"

namespace Project1 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Сводка для EditdDocument

/// </summary>

public ref class EditdDocument : public System::Windows::Forms::Form

{

public:

EditdDocument(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~EditdDocument()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Button^ button1;

protected:

private: System::Windows::Forms::TextBox^ textBox2;

private: System::Windows::Forms::TextBox^ textBox1;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::Label^ label1;

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->button1 = (gcnew System::Windows::Forms::Button());

this->textBox2 = (gcnew System::Windows::Forms::TextBox());

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->label2 = (gcnew System::Windows::Forms::Label());

this->label1 = (gcnew System::Windows::Forms::Label());

this->SuspendLayout();

//

// button1

//

this->button1->Location = System::Drawing::Point(101, 97);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(75, 23);

this->button1->TabIndex = 9;

this->button1->Text = L"Сохранить";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &EditdDocument::Button1\_Click);

//

// textBox2

//

this->textBox2->Location = System::Drawing::Point(116, 53);

this->textBox2->Name = L"textBox2";

this->textBox2->Size = System::Drawing::Size(156, 20);

this->textBox2->TabIndex = 8;

//

// textBox1

//

this->textBox1->Location = System::Drawing::Point(116, 27);

this->textBox1->Name = L"textBox1";

this->textBox1->Size = System::Drawing::Size(156, 20);

this->textBox1->TabIndex = 7;

//

// label2

//

this->label2->AutoSize = true;

this->label2->Location = System::Drawing::Point(12, 56);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(98, 13);

this->label2->TabIndex = 6;

this->label2->Text = L"Номер накладной";

//

// label1

//

this->label1->AutoSize = true;

this->label1->Location = System::Drawing::Point(12, 30);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(91, 13);

this->label1->TabIndex = 5;

this->label1->Text = L"Номер договора";

//

// EditdDocument

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(284, 138);

this->Controls->Add(this->button1);

this->Controls->Add(this->textBox2);

this->Controls->Add(this->textBox1);

this->Controls->Add(this->label2);

this->Controls->Add(this->label1);

this->Name = L"EditdDocument";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"Редактирование документа";

this->Load += gcnew System::EventHandler(this, &EditdDocument::EditdDocument\_Load);

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

public: Document^ doc;

public: Magazin^ EditDocument\_Shown(Magazin^ magz, int id) {

for (int i = 0; i < magz->ArrayDocument->Count; i++)

{

if (magz->ArrayDocument[i]->id == id) {

doc = magz->ArrayDocument[i];

break;

}

}

if (doc != nullptr) {

this->ShowDialog();

for (int i = 0; i < magz->ArrayDocument->Count; i++)

{

if (magz->ArrayDocument[i]->id == id) {

magz->ArrayDocument[i] = doc;

break;

}

}

}

return magz;

}

private: System::Void EditdDocument\_Load(System::Object^ sender, System::EventArgs^ e) {

textBox1->Text = doc->NumberDogovor;

textBox2->Text = doc->Naklodnaya;

}

private: System::Void Button1\_Click(System::Object^ sender, System::EventArgs^ e) {

doc->NumberDogovor = textBox1->Text;

doc->Naklodnaya = textBox2->Text;

this->Close();

}

};

}