МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Ижевский государственный технический университет имени М. Т. Калашникова»

(ФГБОУ ВО «ИжГТУ имени М. Т. Калашникова»)

Институт непрерывного профессионального образования

Кафедра «Вычислительная техника»

Отчёт

по лабораторной работе 1 «Консольное приложение учета»

по дисциплине «Прикладное программирование»

Выполнил работу: студент гр. Д19-191-1 Шкляев Кирилл

Принял: Медякова Д.А

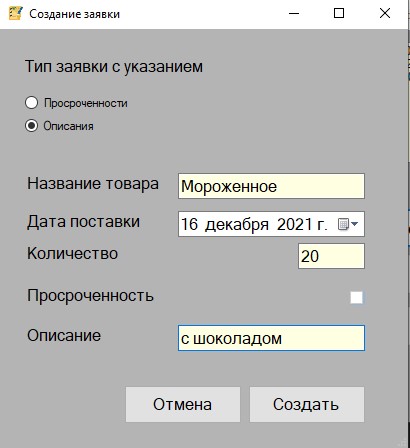
Ижевск 2021

Задача:

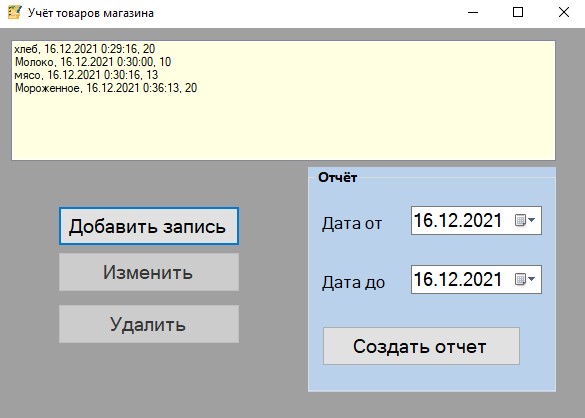
Необходимо реализовать оконный интерфейс к программе, разработанной на  
предыдущих лабораторных работах и написать автоматические тесты  
Реализация оконного приложения  
К моменту 3-ей лабораторной работы у вас уже должно быть:  
• Класс FileDataSource, который отвечает за хранение внесенных в программу  
данных в файле  
• Классы бизнес-логики, которые отвечают за выполнение основных функций  
программы без привязки к конкретному пользовательскому интерфейсу  
Вы должны скопировать эти модули без изменений и заменить лишь модуль, отвечавший  
за ввод/вывод в консоль.  
В программе у вас будет две формы:  
• Основное окно  
• Форма добавления/изменения записи

**Демонстрация решения**

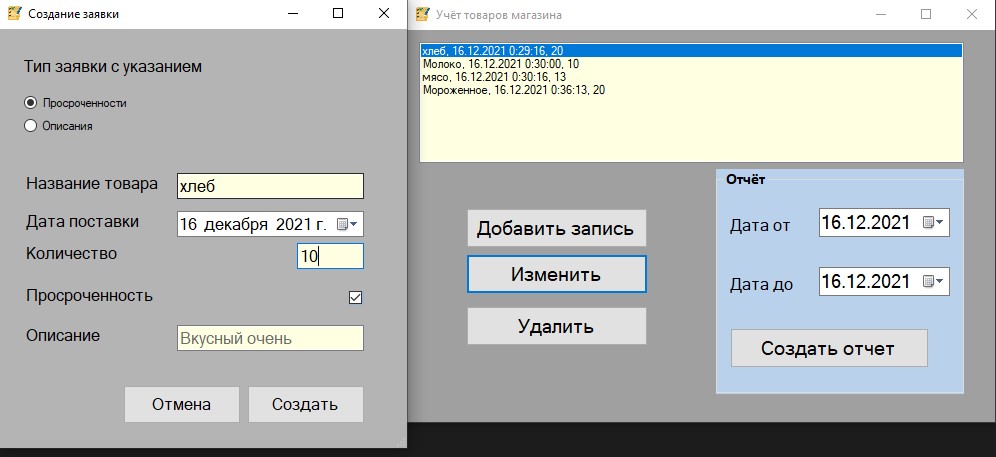
Создаю запись



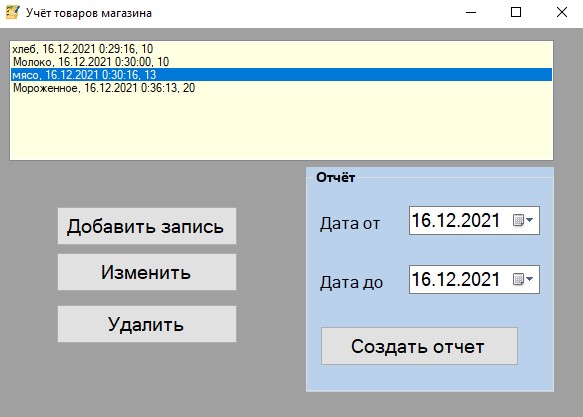
Просматриваю записи



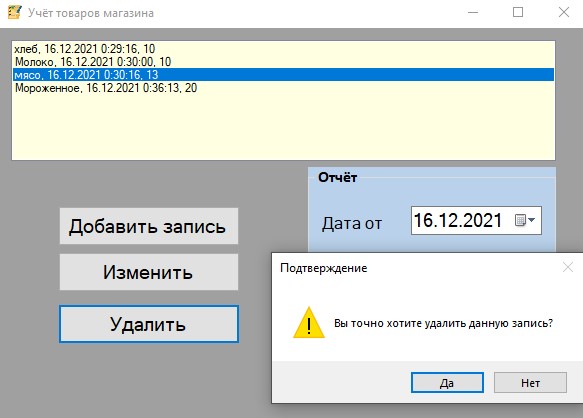
Изменяю 1 запись



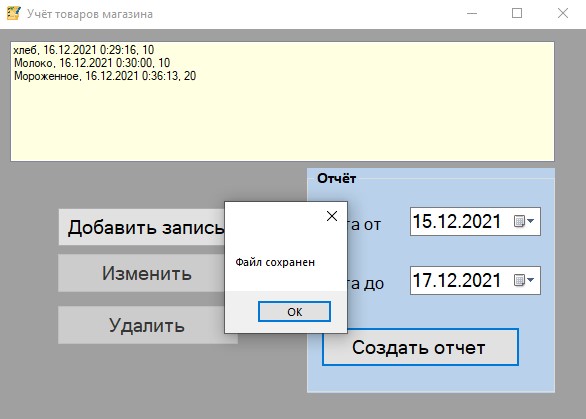
Просматриваю изменённую запись



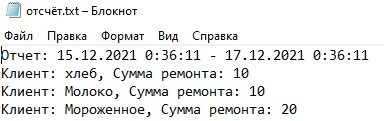
Удаляю 3 запись



Создаю отчёт



Проверяю отчёт



**Автотесты**

using Microsoft.VisualStudio.TestTools.UnitTesting;

using ShklyaevK\_Lab1;

using System;

using System.Collections.Generic;

namespace UnitTestProject1

{

[TestClass]

public class UnitTest

{

BusinessLogic logic = new BusinessLogic(new MemoryDataSource());

[TestMethod]

// Добавляем элемент с правильными данными

public void TestingAdd()

{

Tovar ord = new Description(0, "хлеб", DateTime.Parse("2019.05.05"), 30, "вкусный");

logic.Save(ord);

}

[TestMethod]

// Добавляем элемент с неправильными данными

public void TestingAdd2()

{

Assert.ThrowsException<FormatException>(

() => new Description(0, "хлеб", DateTime.Parse("0000.05.05"), 30, "вкусный"));

}

[TestMethod]

// Изменяем запись с правильными данными

public void TestingReplace()

{

// Добавляем

Tovar ord = new Description(0, "хлеб", DateTime.Parse("2019.05.05"), 30, "вкусный");

logic.Save(ord);

// Изменяем

Tovar ord2 = new Description(1, "хлеб", DateTime.Parse("2019.05.05"), 30, "вкусный");

if (logic.Save(ord2).id != 1) throw new Exception();

}

[TestMethod]

// Изменяем запись с неправильными данными (несуществующий id)

public void TestingReplace2()

{

// Добавляем

Tovar ord = new Description(0, "хлеб", DateTime.Parse("2019.05.05"), 30, "вкусный");

logic.Save(ord);

// Изменяем

Tovar ord2 = new Description(99999, "хлеб", DateTime.Parse("2019.05.05"), 30, "вкусный");

Assert.ThrowsException<Exception>(

() => logic.Save(ord2));

}

[TestMethod]

// Вводим неправильные данные

public void TestingError()

{

Tovar ord = new Description(9999, "хлеб", DateTime.Parse("2019.05.05"), -1, "вкусный");

Assert.ThrowsException<Exception>(

() => logic.Save(ord));

}

[TestMethod]

// Получаем отсортированный список

public void TestingGetList()

{

List<Tovar> list = logic.GetList();

}

[TestMethod]

// Удаляем первую запись

public void TestingDelete()

{

Tovar ord = new Description(0, "хлеб", DateTime.Parse("2019.05.05"), -1, "вкусный");

ord = logic.Save(ord);

logic.Delete(ord.id);

}

[TestMethod]

// Удаляем несуществующую запись

public void TestingDelete2()

{

Tovar ord = new Description(0, "хлеб", DateTime.Parse("2019.05.05"), -12, "вкусный");

logic.Save(ord);

if (!logic.Delete(32623272))

{

Assert.ThrowsException<Exception>(

() => throw new Exception());

}

}

}

}

