**Содержание**

[ВВЕДЕНИЕ 3](#_Toc514084631)

[1 ПОСТАНОВКА ЗАДАЧИ 4](#_Toc514084632)

[2 ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ АНАЛИЗ ЗАДАЧИ 5](#_Toc514084633)

[3 ОПИСАНИЕ ПРОГРАММЫ 8](#_Toc514084634)

[4 ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ 9](#_Toc514084635)

[5 ОПИСАНИЕ ПРИМЕНЕНИЯ 14](#_Toc514084636)

[ЗАКЛЮЧЕНИЕ 16](#_Toc514084637)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИТОЧНИКОВ 17](#_Toc514084638)

[ПРИЛОЖЕНИЕ A 18](#_Toc514084639)

[ПРИЛОЖЕНИЕ Б 84](#_Toc514084640)

**ВВЕДЕНИЕ**

В курсовом проекте решается задача построения БД для учёта лекарственных средств, которая является хранилищем информации содержащим данные.

Для автоматизации системы "Учет лекарственных средств" была выбрана система управления базами данных MS Access, которая является общепризнанным лидером среди инструментов для создания приложений и систем, функционирующих на платформе Windows, одной из самых распространённых операционных систем на рынке и в мире.

В курсовом проекте рассматривается применение информационной системы, обеспечивающей полный спектр информационной поддержки учёта лекарственных средств.

Предметная область использования данной информационной системы *–* аптеки, больницы.

Объектом данного исследования является *–* учет прихода и расхода лекарственных средств.

Цели создания системы – автоматизация процесса ведения учета прихода и расхода лекарственных средств, фирм-поставщиков и фирм-изготовителей.

Данная система позволила решать следующие задачи:

1. Поиск данных.
2. Сортировка данных по данному критерию.
3. Навигация по таблицам.
4. Добавление записей.
5. Редактирование записей.
6. Удаление записей.
7. Просмотр информации.
8. Создание отчёта.

# 1 ПОСТАНОВКА ЗАДАЧИ

Основанием для разработки программы является задание, выданное учащемуся по разработке и сопровождению программного обеспечения.

Задача – написание качественного, отлаженного и работоспособного продукта для автоматизации учета лекарственных средств.

Метод исследования – изучение литературы, написание и отладка программ на компьютере.

Данная программа написана на языке программировании C# в среде Microsoft Visual Studio 2013.

Программное обеспечение «Учёт лекарственных средств» должно обеспечивать выполнение следующих функций:

* предоставление пользователю всех данных о приходе лекарственных средств;
* предоставление пользователю всех данных о расходе лекарственных средств;
* предоставление пользователю всех данных о препаратах;
* предоставление пользователю всех данных о фирмах-изготовителях;
* предоставление пользователю всех данных о фирмах-поставщиках;
* предоставление пользователю возможность добавления, удаления, редактирования информации;
* предоставление пользователю формировать и выводить отчеты по всем данным.

Данные операции были автоматизированы для быстрого и более простого учета лекарственных средств.

# 2 ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ АНАЛИЗ ЗАДАЧИ

В программе «Учет лекарственных средств» используются следующие классы:

1. Form1 – класс главной формы, он позволяет выводить на форму таблицу «Приход», создавать отчет «О наличии», на основе данных таблицы «Приход», и выводить в Excel, а также содержит в себе все элементы управления программой.
2. Form2\_Prihod – класс, позволяющий добавить запись в таблицу «Приход» и сохранить изменения в базе даных.
3. Form3\_VibPreparat – класс служащий для выбора препарата из таблицы «Препараты» и передачи значений на другие формы.
4. Form4\_Rashod – класс, позволяющий добавить запись в таблицу «Расход» и сохранить изменения в базе даных.
5. Form5\_VibDvij – класс служащий для выбора движения из таблицы «Движения» и передачи значений на другие формы.
6. Form6\_Edit – класс, который позволяет редактировать выбраную запись в таблице «Приход» и сохранить изменения в базе даных.
7. Form7\_DeleteZapis – класс, который позволяет удалить выбраную запись в таблице «Приход» и сохранить изменения в базе даных.
8. Form8\_AddPreparat – класс, позволяющий добавить запись в таблицу «Препараты» и сохранить изменения в базе даных.
9. Form9\_DeletePrepapat – класс, который позволяет удалить выбраную запись в таблице «Препараты» и сохранить изменения в базе даных.
10. Form10\_AddPostavschik – класс, позволяющий добавить запись в таблицу «Поставщики» и сохранить изменения в базе даных.
11. Form11\_DeletePostavschik – класс, который позволяет удалить выбраную запись в таблице «Поставщики» и сохранить изменения в базе даных.
12. Form12\_AddIzgotovitel – класс, позволяющий добавить запись в таблицу «Изготовители» и сохранить изменения в базе даных.
13. Form13\_DeleteIzgotovitel – класс, который позволяет удалить выбраную запись в таблице «Изготовители» и сохранить изменения в базе даных.
14. Form14\_AddPeremeschenie – класс, позволяющий добавить запись в таблицу «Перемещение» и сохранить изменения в базе даных.
15. Form15\_DeletePeremeschenie – класс, который позволяет удалить выбраную запись в таблице «Перемещение» и сохранить изменения в базе даных.
16. Form16\_OtOSrokeGodnosti – класс, служащий для формирования отчета «О сроке годности» на основе данных таблицы «Приход» и вывода в Excel.
17. Form17\_OtORashodMed – класс, служащий для формирования отчета «О расходе» на основе данных таблиц «Приход» и «Расход», а также вывода этого отчета в Excel.
18. Form18\_OtoRashodeMedSDate – класс, служащий для формирования отчета «О расходе с датой» на основе данных таблиц «Приход» и «Расход», а также вывода этого отчета в Excel.
19. Form19\_OtOPriobretenii – класс, служащий для формирования отчета «О приобретении» на основе данных таблиц «Приход» и вывода этого отчета в Excel.
20. Form20OtPoVsemOtdeleniam – класс, служащий для формирования отчета «По всем отделениям» на основе данных таблиц «Приход», «Расход» и «Отделения», а также вывода этого отчета в Excel.
21. Form21OtPoOdnomuOtdeleniu – класс, служащий для формирования отчета «По одному отделению» на основе данных таблиц «Приход», «Расход» и «Отделения», а также вывода этого отчета в Excel.
22. Form22\_OtODvij – класс, служащий для формирования отчета «О движении» на основе данных таблиц «Приход» и «Расход», а также вывода этого отчета в Excel.
23. Form23\_OProgramme – класс, который позволяет отоборажает дополнительную информацию о программе на форме.
24. Form24\_VibIzgotovit – класс служащий для выбора фирмы-изготовителя из таблицы «Изготовители» и передачи значений на другие формы.
25. Form25\_VibFirmPost – класс служащий для выбора фирмы-поставщика из таблицы «Поставщики» и передачи значений на другие формы.

Диаграмма иерархии классов показана на рисунке 2.1.

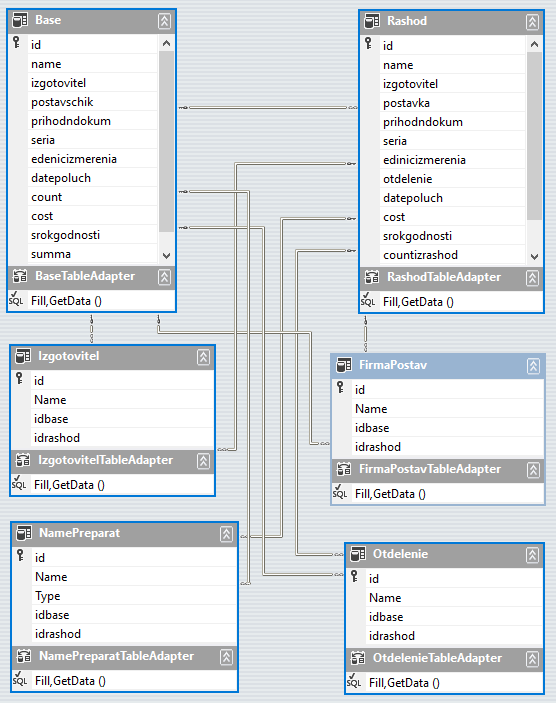


Рисунок 2.1 – Диаграмма иерархии классов

# 3 ОПИСАНИЕ ПРОГРАММЫ

Наименование программы – «Учет лекарственных средств».

Автор программы – Васюхневич П.В., дата создания – 14.05.2018, версия 1.0.

Программная среда Microsoft Visual Studio 2013. Программное обеспечение, необходимое для функционирования программы – операционная система Windows 7 и выше, пакет Microsoft Office, Microsoft.NET FrameWork 4 и выше.

Программа написана на языке C# (CSharp) с использованием Microsoft.NETWindowsForms.

Этапрограмма служит для автоматизации учета лекарственных средств.

Программа взаимодействует с Microsoft Excel при создании отчета, а также с файлом расширения MDB – это файл Microsoft Access.

Минимальные требования для работы программы:

1. Процессор Intel Core I3.
2. Наличие не менее 1024Mб ОЗУ.
3. Наличие не менее 3Мб свободного дискового пространства.
4. SVGA – графический адаптер.
5. Минимальный набор периферийного оборудования.

Для начала работы программы необходимо запустить файл ULS.exe.

Входными данными являются: приход препаратов, расход препаратов, фирмы-поставщики, фирмы-изготовители, отделения, препараты

Выходными данными являются информация о работе аптеки, которая выводится в таблицах на формах, а также отчеты в Excel. Есть восемь вариантов создания отчета, с различной выборкой данных.

# 4 ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ

В данном случае предметом испытаний будет являться программа «Учет лекарственных средств». Данная программа обеспечивает добавление новой записи в базу данных, редактирование ранее созданной записи в базе данных, сортировку записей в базе данных по необходимому полю, поиск по всем полям одновременно и создание отчета с различной выборкой данных.

В ходе тестирования программа вела себя стабильно и без сбоев, после завершения редактирования данных введенная информация отображалась в соответствующих таблицах. На примере показаны тесты главной таблицы.

Главное окно программы показано на рисунке 4.1.

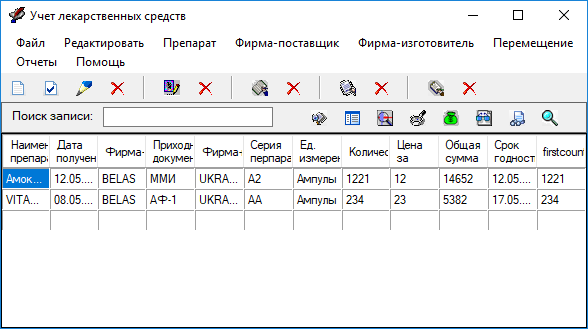


Рисунок 4.1 – Главное окно программы

Добавление новой записи о приходе показано на рисунке 4.2.

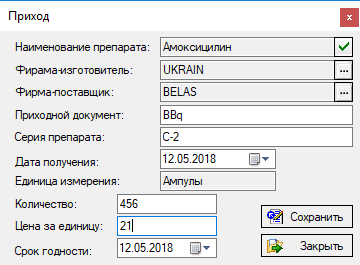


Рисунок 4.2 - Добавление новой записи о приходе

Удаление данных из главной таблицы показано на рисунке 4.3.

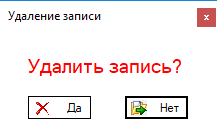


Рисунок 4.3 – Удаление данных из главной таблицы

Поиск данных в главной таблице показан на рисунке 4.4.

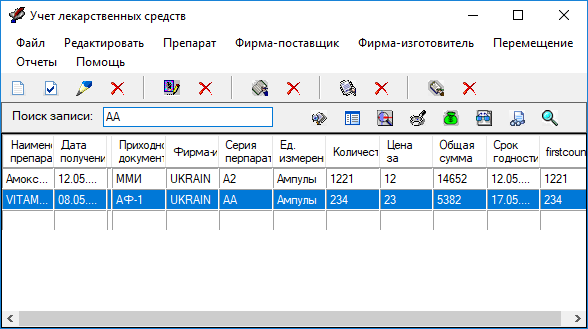


Рисунок 4.4 – Поиск данных в главной таблице

Сортировка данных в главной таблице показана на рисунке 4.5.

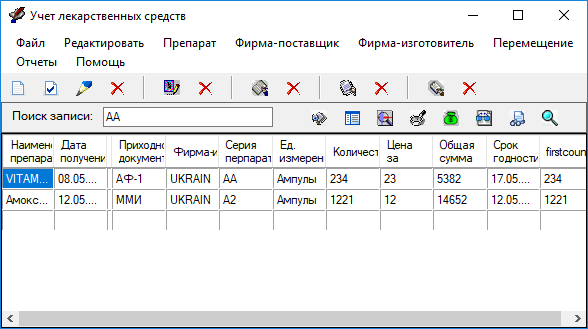


Рисунок 4.5 – Сортировка данных в главной таблице

Создание отчета «О наличии» в Excel показано на рисунке 4.6.

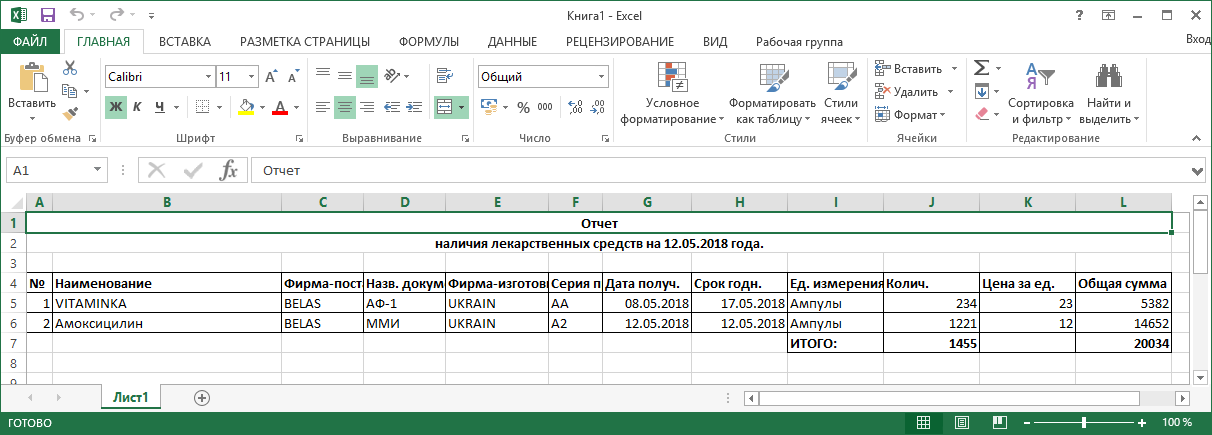


Рисунок 4.6 – Создание отчета «О наличии» в Excel

Добавление новой записи о препарате показано на рисунке 4.7.

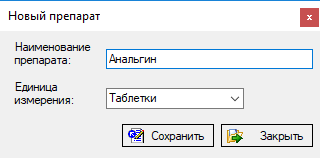


Рисунок 4.7 - Добавление новой записи о препарате

Добавление новой записи о фирме-поставщике показано на рисунке 4.8.

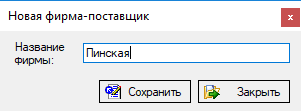


Рисунок 4.8 - Добавление новой записи о фирме-поставщике

Добавление новой записи о фирме-изготовителе показано на рисунке 4.9.

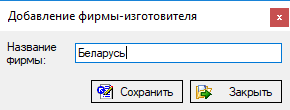


Рисунок 4.9 - Добавление новой записи о фирме-изготовителе

Добавление новой записи о перемещении показано на рисунке 4.10.

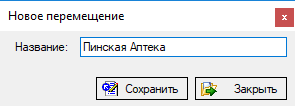


Рисунок 4.10 - Добавление новой записи о перемещении

Удаление записи о препарате показано на рисунке 4.11.

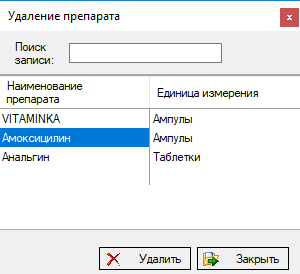


Рисунок 4.11 – Удаление записи о препарате

Удаление записи о фирме-поставщике показано на рисунке 4.12.

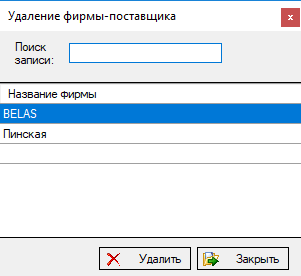


Рисунок 4.12 – Удаление записи о фирме-поставщике

Удаление записи о фирме-изготовителе показано на рисунке 4.13.

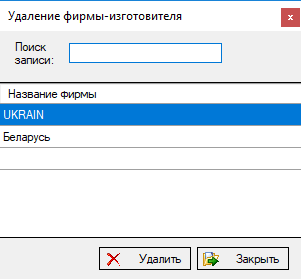


Рисунок 4.13 – Удаление записи о фирме-изготовителе

Удаление записи о перемещении показано на рисунке 4.14.

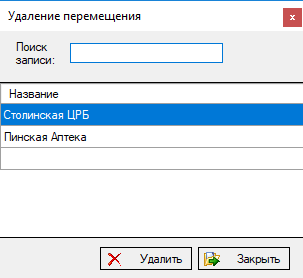


Рисунок 4.14 – Удаление записи о перемещении

Добавление новой записи о расходе показано на рисунке 4.15.

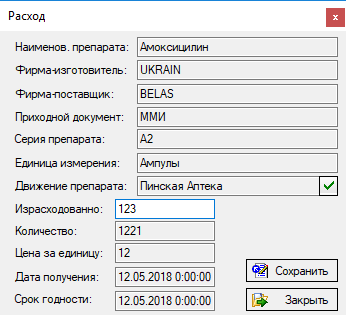


Рисунок 4.15 - Добавление новой записи о расходе

В процессе тестирования программы было проверено:

* Добавление данных во все таблицы
* Удаление данных из всех таблиц
* Поиск данных в главной таблице
* Сортировка данных главной таблице
* Создание отчета в Excel

В процессе тестирования программы не было выявлено ошибок. На данный момент программа функционирует без ошибок.

# 5 ОПИСАНИЕ ПРИМЕНЕНИЯ

Целью создания программы является автоматизация процесса ведения учета препаратов, отделений для перемещения, фирм-поставщиков и фирм-изготовителей, прихода и расхода лекарственных средств.

Предметная область использования данной программы *–* аптеки, больницы.

В программе используются методы добавления, удаления и редактирования базы данных, реализованных через SQL-запросы.

Программа эксплуатируется в условиях индивидуального рабочего места на персональном компьютере. Для работы в диалоговом режиме используется экран дисплея, клавиатура и манипулятор типа "мышь". Занесённые данные хранятся на жёстком диске.

Чтобы установить программу на компьютер, следует запустить установщик «Setup.exe».

Для запуска программы необходимо запустить «ULS.exe». После данной операции откроется главное окно программы (рис. 5.1).

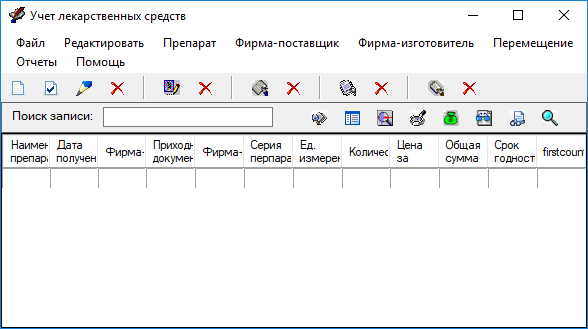


Рисунок 5.1 – Главное окно программы

Для добавления новой записи необходимо нажать кнопку «Редактировать → Приход», или «Редактировать → Расход», или «Препарат → Новый» , или «Фирма-поставщик → Новая» , или «Фирма-изготовитель → Новая» , или «Перемещение → Новое».

Для того, чтобы отредактировать запись, следует курсором мыши выбрать нужную запись в таблице, после этого нажать кнопку «Редактировать запись».

Если при вводе информации в поле для цифр ввести текст, то будет выведено соответсвующиее сообщение об ошибке (рис. 5.2).

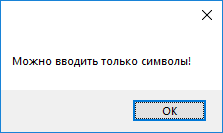


Рисунок 5.2 – Окно поиска по записям в таблице

Для того, чтобы удалить запись, следует курсором мыши выбрать нужную запись в таблице, после этого нажать кнопку «Удалить запись».

Для того, чтобы отсортировать необходимые данные, следует нажать на название нужного столбца прямо в таблице, после чего все записи будут отсортированы.

Для того, чтобы найти запись, достаточно ввести поисковый запрос в специальное окно, после чего из всех записей в таблице будет выбрана та запись, которая максимально подходит введенному запросу (рис. 5.3).



Рисунок 5.3 – Окно поиска по записям в таблице

Чтобы создать отчет, достаточно нажать «Отчеты» и выбрать интересующий вас тип отчета (рис. 5.4).

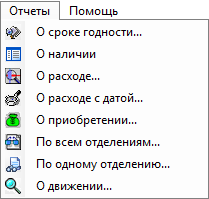


Рисунок 5.4 – Окно поиска по записям в таблице

Закрыть программу возможно путем нажатия «Файл → Выход».

# ЗАКЛЮЧЕНИЕ

В ходе работе была успешно выполнена поставленная задача написать качественный, отлаженный и работоспособный продукт для автоматизации учета лекарственных средств.

Так же было разработано приложение с использованием базы данных, а также обследована предметная область логического и физического проектирования базы данных.

Программный продукт был разработан с удобным и понятным интерфейсом, проблем с работой с базой данных не возникает.

Подходящей средой проектирования подобного приложения является Microsoft Visual Studio 2013, оптимальным средой для хранения данных – СУБД Access.

После неких манипуляций с перемещением программы на разные носители проблем не возникает. Важным условием корректной работы является целостность всех файлов и запуск программы от имени администратора. Удаление, перемещение файлов из корневой папки программы приведет к ошибкам работы программы или вовсе к ее неработоспособности.

Поставленная цель была достигнута, посредством создания программного обеспечения, в котором были реализованы такие функции, как добавление, удаление, редактирование, поиск и сортировка данных, а также создание восьми вариантов отчета на основе всех данных.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИТОЧНИКОВ

1. Информатика: Учебник / Под ред. проф. Макаровой Н. В. – М.: Финансы и статистика, 2000. –768 с.
2. Хомоненко А. Д., Цыганков В. М., Мальцев М.Г. Базы данных. Учебник для ВУЗов / Под ред. проф. А. Д. Хомоненко. - СПб.: КОРОНА-принт, 2000. - 416 с.
3. Черемных С.В. и др. Моделирование и анализ систем. IDEF-технологии: практикум / С.В. Черемных, И.О. Семенов, В.С. Ручкин. – М.: Финансы и статистика, 2005. – 192 с.
4. Карпова Т. С. Базы данных. Модели, разработка, реализация. – СПб.: Питер, 2002. 304 с.
5. Малыхина М.П. Базы данных: основы, проектирование, использование. – СПб.: БХВ – Петербург, 2004. – 512 с.
6. Практикум по информатике / А.А. Землянский, Г.А. Кретова, Ю.Р. Стратонович, Е.А. Яшкова; Под ред. А.А. Землянского. – М.: Колос, 2003. – 384 с.
7. Модели, разработка, реализация. Землянского. – М.: Колос, 2003. – 384 с.
8. Моделирование и анализ систем. IDEF-технологии: практикум / С.В. Черемных, Финансы и статистика, 2005. – 192 с.
9. Информатика: Учебник / Под ред. проф. Издательский дом «Вильямс», 2001. - 1026 с.

# ПРИЛОЖЕНИЕ A

**Листинг программы**

**Листинг 1 – Модуль класса Form1.cs**

*{=========================================================================}*

*{ класс* Form1*.cs* ***}***

*{ гр. П-10 };*

*{ Разработчик: Васюхневич П.В. }*

*{ Модифицирован 8 мая 2018г }*

**{-------------------------------------------------------------------------}**

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using Microsoft.Office.Interop.Excel;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Diagnostics;

using System.Drawing;

using System.Linq;

using System.Reflection;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Excel = Microsoft.Office.Interop.Excel;

namespace ULS

{

public partial class Form1 : Form

{

Object Range;

Excel.\_Worksheet workSheet;

public Form2\_Prihod f2;

public Form4\_Rashod f4;

public Form6\_Edit f6;

Form7\_DeleteZapis f7;

Form8\_AddPreparat f8 ;

Form9\_DeletePrepapat f9 ;

Form10\_AddPostavschik f10 ;

Form11\_DeletePostavschik f11;

Form12\_AddIzgotovitel f12 ;

Form13\_DeleteIzgotovitel f13 ;

Form14\_AddPeremeschenie f14;

Form15\_DeletePeremeschenie f15;

Form16\_OtOSrokeGodnosti f16;

Form17\_OtORashodMed f17;

Form18\_OtoRashodeMedSDate f18 ;

Form19\_OtOPriobretenii f19;

Form20OtPoVsemOtdeleniam f20;

Form21OtPoOdnomuOtdeleniu f21 ;

Form22\_OtODvij f22 ;

Form23\_OProgramme f23 = new Form23\_OProgramme();

public Form1()

{

InitializeComponent();

}

public int newcount = 0;

public int lastfocus = 0;

public int newsum = 0;

//Процедура для обновления таблиц на форме

public void upd()

{

if (newcount >= 0)

{

string connStringAcc = string.Format("Provider=Microsoft.Jet.OLEDB.4.0;Data Source={0}", System.Windows.Forms.Application.StartupPath + "\\BD.mdb");

OleDbConnection connection = new OleDbConnection(connStringAcc);

connection.Open();

OleDbCommand command = new OleDbCommand();

command.Connection = connection;

command.CommandText = string.Format("UPDATE [Base] SET [count]="+ newcount +",[summa]="+newsum+" WHERE (([id]="+lastfocus+"))");

command.ExecuteNonQuery();

connection.Close();

this.baseTableAdapter.Update(bDDataSet);

this.baseTableAdapter.Fill(this.bDDataSet.Base);

}

}

//Процедура выхода из программы

private void выходToolStripMenuItem\_Click(object sender, EventArgs e)

{

Close();

}

//Приход

private void приходToolStripMenuItem\_Click(object sender, EventArgs e)

{

f2 = new Form2\_Prihod(this);

f2.ShowDialog();

}

//Расход

private void расходToolStripMenuItem\_Click(object sender, EventArgs e)

{

try

{

f4 = new Form4\_Rashod(this, dataGridView1, dataGridView1.SelectedCells[0].RowIndex,f21);

f4.ShowDialog();

}

catch (Exception)

{

MessageBox.Show("Нет записей!");

}

}

//Редактирование записи

private void редактироватьЗаписьToolStripMenuItem\_Click(object sender, EventArgs e)

{

try

{

if (dataGridView1.Rows[dataGridView1.SelectedCells[0].RowIndex].Cells[12].Value.ToString() == dataGridView1.Rows[dataGridView1.SelectedCells[0].RowIndex].Cells[7].Value.ToString())

{

f6 = new Form6\_Edit(this, dataGridView1, dataGridView1.SelectedCells[0].RowIndex);

f6.ShowDialog();

}

else

MessageBox.Show("Эту запись редактировать нельзя!");

}

catch (Exception)

{

MessageBox.Show("Нет записей!");

}

}

//Удаление записи

private void удалитьToolStripMenuItem\_Click(object sender, EventArgs e)

{

try

{

f7 = new Form7\_DeleteZapis(this, dataGridView1, dataGridView1.SelectedCells[0].RowIndex);

f7.ShowDialog();

}

catch (Exception)

{

MessageBox.Show("Нет записей!");

}

}

//Новый препарат

private void новыйToolStripMenuItem\_Click(object sender, EventArgs e)

{

f8 = new Form8\_AddPreparat(this);

f8.ShowDialog();

}

//Удалить препарат

private void удалитьToolStripMenuItem1\_Click(object sender, EventArgs e)

{

f9 = new Form9\_DeletePrepapat(this);

f9.ShowDialog();

}

//Новый поставщик

private void новаяToolStripMenuItem\_Click(object sender, EventArgs e)

{

f10 = new Form10\_AddPostavschik(this);

f10.ShowDialog();

}

//Удалить поставщика

private void удалитьToolStripMenuItem2\_Click(object sender, EventArgs e)

{

f11 = new Form11\_DeletePostavschik(this);

f11.ShowDialog();

}

//Новый изготовитель

private void новаяToolStripMenuItem1\_Click(object sender, EventArgs e)

{

f12 = new Form12\_AddIzgotovitel(this);

f12.ShowDialog();

}

//Удалить изготовителя

private void удалитьToolStripMenuItem3\_Click(object sender, EventArgs e)

{

f13 = new Form13\_DeleteIzgotovitel(this);

f13.ShowDialog();

}

//Новое перемещение

private void новоеToolStripMenuItem\_Click(object sender, EventArgs e)

{

f14 = new Form14\_AddPeremeschenie(this);

f14.ShowDialog();

}

//Удалить перемещение

private void удалитьToolStripMenuItem4\_Click(object sender, EventArgs e)

{

f15 = new Form15\_DeletePeremeschenie(this);

f15.ShowDialog();

}

//Отчет о сроке годности

private void оСрокеГодностиToolStripMenuItem\_Click(object sender, EventArgs e)

{

f16 = new Form16\_OtOSrokeGodnosti(this);

f16.ShowDialog();

}

//Отчет о расходе

private void оРасходеToolStripMenuItem\_Click(object sender, EventArgs e)

{

f17 = new Form17\_OtORashodMed(this);

f17.ShowDialog();

}

//Отчет о расходе с датой

private void оРасходеСДатойToolStripMenuItem\_Click(object sender, EventArgs e)

{

f18 = new Form18\_OtoRashodeMedSDate(this);

f18.ShowDialog();

}

//Отчет о приобретении

private void оПриобретенииToolStripMenuItem\_Click(object sender, EventArgs e)

{

f19 = new Form19\_OtOPriobretenii(this);

f19.ShowDialog();

}

//Отчет по всем отделениям

private void поВсемОтделениямToolStripMenuItem\_Click(object sender, EventArgs e)

{

f20 = new Form20OtPoVsemOtdeleniam(this);

f20.ShowDialog();

}

//Отчет по одному отделению

private void поОдномуОтделениюToolStripMenuItem\_Click(object sender, EventArgs e)

{

f21 = new Form21OtPoOdnomuOtdeleniu(this,f4);

f21.ShowDialog();

}

//Отчет о движении

private void оДвиженииToolStripMenuItem\_Click(object sender, EventArgs e)

{

f22 = new Form22\_OtODvij(this);

f22.ShowDialog();

}

//Вызов окно "О программе"

private void оПрограммеToolStripMenuItem\_Click(object sender, EventArgs e)

{

f23 = new Form23\_OProgramme();

f23.ShowDialog();

}

private void Form1\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "bDDataSet.Base". При необходимости она может быть перемещена или удалена.

this.baseTableAdapter.Fill(this.bDDataSet.Base);

}

private void Form1\_Activated(object sender, EventArgs e)

{

upd();

this.baseTableAdapter.Fill(this.bDDataSet.Base);

}

//Поиск по главной таблице

private void textBox1\_TextChanged(object sender, EventArgs e)

{

for (int i = 0; i < dataGridView1.RowCount; i++)

{

dataGridView1.Rows[i].Selected = false;

for (int j = 0; j < dataGridView1.ColumnCount; j++)

if (dataGridView1.Rows[i].Cells[j].Value != null)

if (dataGridView1.Rows[i].Cells[j].Value.ToString().Contains(textBox1.Text))

{

dataGridView1.Rows[i].Selected = true;

break;

}

}

if (textBox1.Text=="")

dataGridView1.ClearSelection();

}

private void button1\_Click(object sender, EventArgs e)

{

f2 = new Form2\_Prihod(this);

f2.ShowDialog();

}

private void button2\_Click(object sender, EventArgs e)

{

try

{

f4 = new Form4\_Rashod(this, dataGridView1, dataGridView1.SelectedCells[0].RowIndex, f21);

f4.ShowDialog();

}

catch (Exception)

{

MessageBox.Show("Нет записей!");

}

}

private void button3\_Click(object sender, EventArgs e)

{

try

{

if (dataGridView1.Rows[dataGridView1.SelectedCells[0].RowIndex].Cells[12].Value.ToString() == dataGridView1.Rows[dataGridView1.SelectedCells[0].RowIndex].Cells[7].Value.ToString())

{

f6 = new Form6\_Edit(this, dataGridView1, dataGridView1.SelectedCells[0].RowIndex);

f6.ShowDialog();

}

else

MessageBox.Show("Эту запись редактировать нельзя!");

}

catch (Exception)

{

MessageBox.Show("Нет записей!");

}

}

private void button4\_Click(object sender, EventArgs e)

{

try

{

f7 = new Form7\_DeleteZapis(this, dataGridView1, dataGridView1.SelectedCells[0].RowIndex);

f7.ShowDialog();

}

catch (Exception)

{

MessageBox.Show("Нет записей!");

}

}

private void button5\_Click(object sender, EventArgs e)

{

f8 = new Form8\_AddPreparat(this);

f8.ShowDialog();

}

private void button6\_Click(object sender, EventArgs e)

{

f9 = new Form9\_DeletePrepapat(this);

f9.ShowDialog();

}

private void button7\_Click(object sender, EventArgs e)

{

f10 = new Form10\_AddPostavschik(this);

f10.ShowDialog();

}

private void button8\_Click(object sender, EventArgs e)

{

f11 = new Form11\_DeletePostavschik(this);

f11.ShowDialog();

}

private void button9\_Click(object sender, EventArgs e)

{

f12 = new Form12\_AddIzgotovitel(this);

f12.ShowDialog();

}

private void button10\_Click(object sender, EventArgs e)

{

f13 = new Form13\_DeleteIzgotovitel(this);

f13.ShowDialog();

}

private void button11\_Click(object sender, EventArgs e)

{

f14 = new Form14\_AddPeremeschenie(this);

f14.ShowDialog();

}

private void button12\_Click(object sender, EventArgs e)

{

f15 = new Form15\_DeletePeremeschenie(this);

f15.ShowDialog();

}

private void справкаToolStripMenuItem\_Click(object sender, EventArgs e)

{

try

{

Process SysInfo = new Process();

SysInfo.StartInfo.ErrorDialog = true;

SysInfo.StartInfo.FileName = "help.chm";

SysInfo.Start();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void button13\_Click(object sender, EventArgs e)

{

f16 = new Form16\_OtOSrokeGodnosti(this);

f16.ShowDialog();

}

private void оНаличииToolStripMenuItem\_Click(object sender, EventArgs e)

{

ExportToExcel();

}

//Процедура для создания, прорисовки и вывода данных в Excel

private void ExportToExcel()

{

int count = 0;

for (int i = 0; i < dataGridView1.Rows.Count - 1; i++)

if (int.Parse(dataGridView1.Rows[i].Cells[7].Value.ToString())>0)

{ count++; }

if (count != 0)

{

Excel.Application exApp = new Excel.Application();

exApp.Visible = true;

exApp.Workbooks.Add();

workSheet = (Worksheet)exApp.ActiveSheet;

SetMerge("A1:L1", 3);

SetMerge("A2:L2", 3);

workSheet.Cells[1, 1].EntireRow.Font.Bold = true;

workSheet.Cells[2, 1].EntireRow.Font.Bold = true;

workSheet.Cells[4, 1].EntireRow.Font.Bold = true;

workSheet.Cells[4, 2].EntireRow.Font.Bold = true;

workSheet.Cells[4, 3].EntireRow.Font.Bold = true;

workSheet.Cells[4, 4].EntireRow.Font.Bold = true;

workSheet.Cells[4, 5].EntireRow.Font.Bold = true;

workSheet.Cells[4, 6].EntireRow.Font.Bold = true;

workSheet.Cells[4, 7].EntireRow.Font.Bold = true;

workSheet.Cells[4, 8].EntireRow.Font.Bold = true;

SetBorderStyle("A4:L4", 1);

workSheet.Cells[1, 1] = "Отчет";

workSheet.Cells[2, 1] = "наличия лекарственных средств на " + DateTime.Now.ToShortDateString() + " года.";

workSheet.Cells[4, 1] = "№";

workSheet.Cells[4, 2] = "Наименование";

workSheet.Cells[4, 3] = "Фирма-поставщик";

workSheet.Cells[4, 4] = "Назв. документа";

workSheet.Cells[4, 5] = "Фирма-изготовитель";

workSheet.Cells[4, 6] = "Серия препарата";

workSheet.Cells[4, 7] = "Дата получ.";

workSheet.Cells[4, 8] = "Срок годн.";

workSheet.Cells[4, 9] = "Ед. измерения";

workSheet.Cells[4, 10] = "Колич.";

workSheet.Cells[4, 11] = "Цена за ед.";

workSheet.Cells[4, 12] = "Общая сумма";

workSheet.Cells[count + 5, 9] = "ИТОГО:";

workSheet.Cells[count + 5, 9].EntireRow.Font.Bold = true;

SetBorderStyle("I" + (count + 5).ToString() + ":L" + (count + 5).ToString(), 1);

SetColumnWidth("A1", 3);

SetColumnWidth("B1", 32);

SetColumnWidth("C1", 11);

SetColumnWidth("D1", 11);

SetColumnWidth("E1", 14);

SetColumnWidth("F1", 7);

SetColumnWidth("G1", 12);

SetColumnWidth("H1", 13);

SetColumnWidth("I1", 13);

SetColumnWidth("J1", 13);

SetColumnWidth("K1", 13);

SetColumnWidth("L1", 13);

int rowExcel = 5;

int col = 0;

int sum = 0;

for (int i = 0; i < dataGridView1.Rows.Count - 1; i++)

{

SetBorderStyle("A" + (i + 5).ToString() + ":L" + (i + 5).ToString(), 1);

workSheet.Cells[rowExcel, "A"] = dataGridView1.Rows[i].Cells[11].Value;

workSheet.Cells[rowExcel, "B"] = dataGridView1.Rows[i].Cells[0].Value;

workSheet.Cells[rowExcel, "C"] = dataGridView1.Rows[i].Cells[2].Value;

workSheet.Cells[rowExcel, "D"] = dataGridView1.Rows[i].Cells[3].Value;

workSheet.Cells[rowExcel, "E"] = dataGridView1.Rows[i].Cells[4].Value;

workSheet.Cells[rowExcel, "F"] = dataGridView1.Rows[i].Cells[5].Value;

workSheet.Cells[rowExcel, "G"] = dataGridView1.Rows[i].Cells[1].Value;

workSheet.Cells[rowExcel, "H"] = dataGridView1.Rows[i].Cells[10].Value;

workSheet.Cells[rowExcel, "I"] = dataGridView1.Rows[i].Cells[6].Value;

workSheet.Cells[rowExcel, "J"] = dataGridView1.Rows[i].Cells[7].Value;

workSheet.Cells[rowExcel, "K"] = dataGridView1.Rows[i].Cells[8].Value;

workSheet.Cells[rowExcel, "L"] = dataGridView1.Rows[i].Cells[9].Value;

++rowExcel;

col = col + int.Parse(dataGridView1.Rows[i].Cells[7].Value.ToString());

sum = sum + int.Parse(dataGridView1.Rows[i].Cells[9].Value.ToString());

}

workSheet.Cells[count + 5, 10] = col.ToString();

workSheet.Cells[count + 5, 12] = sum.ToString();

workSheet.Cells[count + 5, 10].EntireRow.Font.Bold = true;

workSheet.Cells[count + 5, 12].EntireRow.Font.Bold = true;

}

else

{ MessageBox.Show("Лекарственных средств нет в наличии!"); }

}

private void SetColumnWidth(string range, double Width)//TODO: Ширина столбца

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Width };

Range.GetType().InvokeMember("ColumnWidth", BindingFlags.SetProperty, null, Range, args);

}

private void SetBorderStyle(string range, int Style)//TODO: Стиль границ ячейки

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Style };

object Borders = Range.GetType().InvokeMember("Borders", BindingFlags.GetProperty, null, Range, null);

Borders = Range.GetType().InvokeMember("LineStyle", BindingFlags.SetProperty, null, Borders, args);

}

private void SetWrapText(string range, bool Value)//TODO: Перенос по словам

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Value };

Range.GetType().InvokeMember("WrapText", BindingFlags.SetProperty, null, Range, args);

}

private void SetRowHeight(string range, double Height)//TODO: Высота строк

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty,

null, workSheet, new object[] { range });

object[] args = new object[] { Height };

Range.GetType().InvokeMember("RowHeight", BindingFlags.SetProperty, null, Range, args);

}

private void SetMerge(string range, int Alignment)//TODO: Выравнивание текста в ячейке

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Alignment };

Range.GetType().InvokeMember("MergeCells", BindingFlags.SetProperty, null, Range, new object[] { true });

Range.GetType().InvokeMember("HorizontalAlignment", BindingFlags.SetProperty, null, Range, args);

}

private void SetFont(string range, Microsoft.Office.Interop.Excel.Font font)

{

//Range.Font.Name

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty,

null, workSheet, new object[] { range });

object Font = Range.GetType().InvokeMember("Font", BindingFlags.GetProperty,

null, Range, null);

Range.GetType().InvokeMember("Name", BindingFlags.SetProperty, null,

Font, new object[] { font.Name });

Range.GetType().InvokeMember("Size", BindingFlags.SetProperty, null,

Font, new object[] { font.Size });

}

private void button14\_Click(object sender, EventArgs e)

{

ExportToExcel();

}

private void button15\_Click(object sender, EventArgs e)

{

f17 = new Form17\_OtORashodMed(this);

f17.ShowDialog();

}

private void button16\_Click(object sender, EventArgs e)

{

f18 = new Form18\_OtoRashodeMedSDate(this);

f18.ShowDialog();

}

private void button17\_Click(object sender, EventArgs e)

{

f19 = new Form19\_OtOPriobretenii(this);

f19.ShowDialog();

}

private void button18\_Click(object sender, EventArgs e)

{

f20 = new Form20OtPoVsemOtdeleniam(this);

f20.ShowDialog();

}

private void button19\_Click(object sender, EventArgs e)

{

f21 = new Form21OtPoOdnomuOtdeleniu(this, f4);

f21.ShowDialog();

}

private void button20\_Click(object sender, EventArgs e)

{

f22 = new Form22\_OtODvij(this);

f22.ShowDialog();

}

}

}

**Листинг 2 – Модуль класса Form2\_Prihod.cs**

{======================================================================}

*{ класс Form2\_Prihod.cs }*

*{ гр. П-10 };*

*{ Разработчик: Васюхневич П.В. }*

*{ Модифицирован 8 мая 2018г }*

**{----------------------------------------------------------------------}**

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ULS

{

public partial class Form2\_Prihod : Form

{

Form1 frm1;

Form3\_VibPreparat f3;

Form24\_VibIzgotovit f24;

Form25\_VibFirmPost f25;

public Form2\_Prihod(Form1 f)

{

InitializeComponent();

f3 = new Form3\_VibPreparat(f,this, f.f6);

frm1 = f;

f24 = new Form24\_VibIzgotovit(this, f.f6);

f25 = new Form25\_VibFirmPost(this, f.f6);

}

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

private void button3\_Click(object sender, EventArgs e)

{

f3.fr2 = true;

f3.fr6 = false;

f3.ShowDialog();

}

private void Form2\_Prihod\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "bDDataSet.FirmaPostav". При необходимости она может быть перемещена или удалена.

this.firmaPostavTableAdapter.Fill(this.bDDataSet.FirmaPostav);

// TODO: данная строка кода позволяет загрузить данные в таблицу "bDDataSet.Izgotovitel". При необходимости она может быть перемещена или удалена.

this.izgotovitelTableAdapter.Fill(this.bDDataSet.Izgotovitel);

}

//Функция для выбора препарата

public void func(string s,string s2)

{

textBox1.Text = s;

textBox4.Text = s2;

}

//Функция для выбора фирмы-изготовителя

public void func2(string s)

{

textBox7.Text = s;

}

//функция для выбора фирмы-поставщика

public void func3(string s)

{

textBox8.Text = s;

}

//Проверка заполнены ли все поля

void en()

{

if (textBox1.Text != "" && textBox2.Text != "" && textBox3.Text != "" && textBox4.Text != "" && textBox5.Text != "" && textBox6.Text != "" && textBox7.Text != "" && textBox8.Text != "")

button1.Enabled = true;

else

button1.Enabled = false;

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

en();

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

en();

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

en();

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

en();

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

en();

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

en();

}

private void textBox5\_TextChanged(object sender, EventArgs e)

{

try

{ if (textBox5.Text != "")

{

int i = int.Parse(textBox5.Text);

}

}

catch(Exception)

{ MessageBox.Show("Можно вводить только символы!"); textBox5.Text = ""; }

en();

}

private void textBox6\_TextChanged(object sender, EventArgs e)

{

try

{ if (textBox6.Text != "")

{ int i = int.Parse(textBox6.Text); }

}

catch (Exception)

{ MessageBox.Show("Можно вводить только символы!"); textBox6.Text = ""; }

en();

}

private void button1\_Click(object sender, EventArgs e)

{

string connStringAcc = string.Format("Provider=Microsoft.Jet.OLEDB.4.0;Data Source={0}", Application.StartupPath + "\\BD.mdb");

OleDbConnection connection = new OleDbConnection(connStringAcc);

connection.Open();

OleDbCommand command = new OleDbCommand();

command.Connection = connection;

int sum = int.Parse(textBox5.Text) \* int.Parse(textBox6.Text);

int cost = int.Parse(textBox6.Text) + 0;

int count = int.Parse(textBox5.Text) + 0;

command.CommandText = "insert into [Base] ( [firstcount], [count], [cost], [name], [izgotovitel], [postavschik], [prihodndokum], [seria], [edenicizmerenia], [datepoluch], [srokgodnosti], [summa]) values (" + count + "," + count + "," + cost + ",'" + textBox1.Text + "','" + textBox7.Text + "','" + textBox8.Text + "','" + textBox2.Text + "','" + textBox3.Text + "','" + textBox4.Text + "','" + dateTimePicker1.Value.ToShortDateString() + "','" + dateTimePicker2.Value.ToShortDateString() + "', " + sum + ")";

command.ExecuteNonQuery();

connection.Close();

Close();

}

private void button4\_Click(object sender, EventArgs e)

{

f24.fr2 = true;

f24.fr6 = false;

f24.ShowDialog();

}

private void button5\_Click(object sender, EventArgs e)

{

f25.fr2 = true;

f25.fr6 = false;

f25.ShowDialog();

}

}

}

**Листинг 3 – Модуль класса Form3\_VibPreparat.cs**

{======================================================================}

*{ класс Form3\_VibPreparat.cs }*

*{ гр. П-10 };*

*{ Разработчик: Васюхневич П.В. }*

*{ Модифицирован 8 мая 2018г }*

**{----------------------------------------------------------------------}**

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ULS

{

public partial class Form3\_VibPreparat : Form

{

Form1 frm1;

Form2\_Prihod frm2;

Form6\_Edit frm6;

public Form3\_VibPreparat(Form1 f, Form2\_Prihod f2, Form6\_Edit f6)

{

InitializeComponent();

frm1 = f;

frm2 = f2;

frm6 = f6;

}

public bool fr2 = false;

public bool fr6 = false;

private void button2\_Click(object sender, EventArgs e)

{

Hide();

}

private void Form3\_VibPreparat\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "bDDataSet.NamePreparat". При необходимости она может быть перемещена или удалена.

this.namePreparatTableAdapter.Fill(this.bDDataSet.NamePreparat);

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

try

{

namePreparatBindingSource.Position = namePreparatBindingSource.Find("Name", textBox1.Text);

}

catch (Exception)

{

}

}

private void button1\_Click(object sender, EventArgs e)

{

if (fr2==true)

frm2.func(dataGridView1.CurrentRow.Cells[0].Value.ToString(), dataGridView1.CurrentRow.Cells[1].Value.ToString());

if (fr6==true)

frm6.func(dataGridView1.CurrentRow.Cells[0].Value.ToString(), dataGridView1.CurrentRow.Cells[1].Value.ToString());

Close();

}

}

}

**Листинг 4 – Модуль класса Form4\_Rashod.cs**

{======================================================================}

*{ класс Form4\_Rashod.cs }*

*{ гр. П-10 };*

*{ Разработчик: Васюхневич П.В. }*

*{ Модифицирован 8 мая 2018г }*

**{----------------------------------------------------------------------}**

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ULS

{

public partial class Form4\_Rashod : Form

{

Form1 frm1;

Form5\_VibDvij f5;

DataGridView dv;

int focus;

public Form4\_Rashod(Form1 f, DataGridView s, int index, Form21OtPoOdnomuOtdeleniu f21)

{

InitializeComponent();

frm1 = f;

f5 = new Form5\_VibDvij(f,this,f21);

dv = s;

focus = index;

}

//Функция для выбора движения

public void func(string s)

{

textBox7.Text = s;

}

//Проверка заполнены ли все поля

public void en()

{

if (textBox7.Text != "" && textBox8.Text != "" )

button1.Enabled = true;

else

button1.Enabled = false;

}

private void Form4\_Rashod\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "bDDataSet.Rashod". При необходимости она может быть перемещена или удалена.

this.rashodTableAdapter.Fill(this.bDDataSet.Rashod);

textBox1.Text = dv.Rows[focus].Cells[0].Value.ToString();

textBox2.Text = dv.Rows[focus].Cells[4].Value.ToString();

textBox3.Text = dv.Rows[focus].Cells[2].Value.ToString();

textBox4.Text = dv.Rows[focus].Cells[3].Value.ToString();

textBox5.Text = dv.Rows[focus].Cells[5].Value.ToString();

textBox6.Text = dv.Rows[focus].Cells[6].Value.ToString();

textBox9.Text = dv.Rows[focus].Cells[7].Value.ToString();

textBox10.Text = dv.Rows[focus].Cells[8].Value.ToString();

textBox11.Text = dv.Rows[focus].Cells[1].Value.ToString();

textBox12.Text = dv.Rows[focus].Cells[10].Value.ToString();

}

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

private void button3\_Click(object sender, EventArgs e)

{

f5.f44 = true;

f5.f21 = false;

f5.ShowDialog();

}

private void textBox8\_TextChanged(object sender, EventArgs e)

{

try

{

if (textBox8.Text != "")

{ int i = int.Parse(textBox8.Text); }

}

catch (Exception)

{ MessageBox.Show("Можно вводить только символы!"); textBox8.Text = ""; }

en();

}

private void button1\_Click(object sender, EventArgs e)

{

if (int.Parse(textBox8.Text) > 0 && int.Parse(textBox9.Text) - int.Parse(textBox8.Text) >= 0)

{

string connStringAcc = string.Format("Provider=Microsoft.Jet.OLEDB.4.0;Data Source={0}", Application.StartupPath + "\\BD.mdb");

OleDbConnection connection = new OleDbConnection(connStringAcc);

connection.Open();

OleDbCommand command = new OleDbCommand();

command.Connection = connection;

int cost = int.Parse(textBox10.Text) + 0;

int count = int.Parse(textBox9.Text) + 0;

command.CommandText = "insert into [Rashod] ([idbase], [cost], [name], [izgotovitel], [postavka], [prihodndokum], [seria], [edinicizmerenia], [otdelenie], [countizrashod], [datepoluch], [srokgodnosti], [datarashod]) values (" + int.Parse(dv.Rows[focus].Cells[11].Value.ToString()) + "," + cost + ",'" + textBox1.Text + "','" + textBox2.Text + "','" + textBox3.Text + "','" + textBox4.Text + "','" + textBox5.Text + "','" + textBox6.Text + "','" + textBox7.Text + "'," + int.Parse(textBox8.Text) + ",'" + textBox11.Text + "','" + textBox12.Text + "','" + DateTime.Now.ToShortDateString() + "')";

command.ExecuteNonQuery();

connection.Close();

frm1.newcount = int.Parse(textBox9.Text) - int.Parse(textBox8.Text);

frm1.lastfocus = int.Parse(dv.Rows[focus].Cells[11].Value.ToString());

frm1.newsum = int.Parse(dv.Rows[focus].Cells[9].Value.ToString()) - (int.Parse(textBox8.Text) \* int.Parse(textBox10.Text));

Close();

}

else

{

MessageBox.Show("Введенное значение превышает количество, либо оно равно нулю!");

}

}

private void textBox7\_TextChanged(object sender, EventArgs e)

{

en();

}

}

}

**Листинг 5 – Модуль класса Form5\_VibDvij.cs**

{======================================================================}

*{ класс Form5\_VibDvij.cs }*

*{ гр. П-10 };*

*{ Разработчик: Васюхневич П.В. }*

*{ Модифицирован 8 мая 2018г }*

**{----------------------------------------------------------------------}**

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ULS

{

public partial class Form5\_VibDvij : Form

{

Form1 frm1;

Form4\_Rashod frm4;

public bool f44;

public bool f21;

Form21OtPoOdnomuOtdeleniu ff21;

public Form5\_VibDvij(Form1 f, Form4\_Rashod f4, Form21OtPoOdnomuOtdeleniu f2111)

{

frm1 = f;

InitializeComponent();

frm4 = f4;

ff21 = f2111;

}

private void button2\_Click(object sender, EventArgs e)

{

frm4.en();

Close();

}

private void Form5\_VibOPrepar\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "bDDataSet.Otdelenie". При необходимости она может быть перемещена или удалена.

this.otdelenieTableAdapter.Fill(this.bDDataSet.Otdelenie);

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

try

{

otdelenieBindingSource.Position = otdelenieBindingSource.Find("Name", textBox1.Text);

}

catch (Exception)

{ }

}

private void button1\_Click(object sender, EventArgs e)

{

if (f44 == true)

{

frm4.func(dataGridView1.CurrentRow.Cells[0].Value.ToString());

frm4.en();

}

if (f21==true)

{

ff21.func(dataGridView1.CurrentRow.Cells[0].Value.ToString());

ff21.en();

}

Close();

}

}

}

**Листинг 6 – Модуль класса Form6\_Edit.cs**

{======================================================================}

*{ класс Form6\_Edit.cs }*

*{ гр. П-10 };*

*{ Разработчик: Васюхневич П.В. }*

*{ Модифицирован 8 мая 2018г }*

**{----------------------------------------------------------------------}**

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ULS

{

public partial class Form6\_Edit : Form

{

Form1 frm1;

Form3\_VibPreparat f3;

DataGridView dv;

Form24\_VibIzgotovit f24;

Form25\_VibFirmPost f25;

int focus;

public Form6\_Edit(Form1 f, DataGridView s, int index)

{

InitializeComponent();

frm1 = f;

f3 = new Form3\_VibPreparat(f, f.f2, this);

dv = s;

focus = index;

f24 = new Form24\_VibIzgotovit(f.f2, this);

f25 = new Form25\_VibFirmPost(f.f2,this);

}

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

void en()

{

if (textBox1.Text != "" && textBox2.Text != "" && textBox3.Text != "" && textBox4.Text != "" && textBox5.Text != "" && textBox6.Text != "" && textBox7.Text != "" && textBox8.Text != "")

button1.Enabled = true;

else

button1.Enabled = false;

}

private void button3\_Click(object sender, EventArgs e)

{

f3.fr2 = false;

f3.fr6 = true;

f3.ShowDialog();

}

public void func(string s, string s2)

{

textBox1.Text = s;

textBox4.Text = s2;

}

public void func2(string s)

{

textBox7.Text = s;

}

public void func3(string s)

{

textBox8.Text = s;

}

public void upd()

{

string connStringAcc = string.Format("Provider=Microsoft.Jet.OLEDB.4.0;Data Source={0}", Application.StartupPath + "\\BD.mdb");

OleDbConnection connection = new OleDbConnection(connStringAcc);

connection.Open();

OleDbCommand command = new OleDbCommand();

command.Connection = connection;

int count = (int.Parse(textBox5.Text));

int cost = (int.Parse(textBox6.Text));

string name = (textBox1.Text);

string izgotovitel = (textBox7.Text);

string postav = (textBox8.Text);

string prihodn = (textBox2.Text);

string seria = (textBox3.Text);

string edizm = (textBox4.Text);

int summ = (int.Parse(textBox5.Text) \* int.Parse(textBox6.Text));

string d1 = (dateTimePicker1.Value.ToShortDateString());

string d2 = (dateTimePicker2.Value.ToShortDateString());

int lastfocus = int.Parse(dv.Rows[focus].Cells[11].Value.ToString());

command.CommandText = string.Format("UPDATE [Base] SET [firstcount]=" + count + ", [count]=" + count + ", [cost]=" + cost + ", [name]='" + name + "', [izgotovitel]='" + izgotovitel + "', [postavschik]='" + postav + "', [prihodndokum]='" + prihodn + "', [seria]='" + seria + "', [edenicizmerenia]='" + edizm + "', [datepoluch]='" + d1 + "', [srokgodnosti]='" + d2 + "', [summa]=" + summ + " WHERE (([id]=" + lastfocus + "))");

command.ExecuteNonQuery();

connection.Close();

}

private void Form6\_Edit\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "bDDataSet.FirmaPostav". При необходимости она может быть перемещена или удалена.

this.firmaPostavTableAdapter.Fill(this.bDDataSet.FirmaPostav);

// TODO: данная строка кода позволяет загрузить данные в таблицу "bDDataSet.Izgotovitel". При необходимости она может быть перемещена или удалена.

this.izgotovitelTableAdapter.Fill(this.bDDataSet.Izgotovitel);

// TODO: данная строка кода позволяет загрузить данные в таблицу "bDDataSet.FirmaPostav". При необходимости она может быть перемещена или удалена.

this.firmaPostavTableAdapter.Fill(this.bDDataSet.FirmaPostav);

textBox1.Text = dv.Rows[focus].Cells[0].Value.ToString();

textBox7.Text = dv.Rows[focus].Cells[4].Value.ToString();

textBox8.Text = dv.Rows[focus].Cells[2].Value.ToString();

textBox2.Text = dv.Rows[focus].Cells[3].Value.ToString();

textBox3.Text = dv.Rows[focus].Cells[5].Value.ToString();

dateTimePicker1.Value= DateTime.Parse(dv.Rows[focus].Cells[1].Value.ToString());

textBox4.Text = dv.Rows[focus].Cells[6].Value.ToString();

textBox5.Text = dv.Rows[focus].Cells[7].Value.ToString();

textBox6.Text = dv.Rows[focus].Cells[8].Value.ToString();

dateTimePicker2.Value = DateTime.Parse(dv.Rows[focus].Cells[10].Value.ToString());

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

en();

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

en();

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

en();

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

en();

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

en();

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

en();

}

private void textBox5\_TextChanged(object sender, EventArgs e)

{

try

{

if (textBox5.Text != "")

{ int i = int.Parse(textBox5.Text); }

}

catch (Exception)

{ MessageBox.Show("Можно вводить только символы!"); textBox5.Text = ""; }

en();

}

private void textBox6\_TextChanged(object sender, EventArgs e)

{

try

{

if (textBox6.Text != "")

{ int i = int.Parse(textBox6.Text); }

}

catch (Exception)

{ MessageBox.Show("Можно вводить только символы!"); textBox5.Text = ""; }

en();

}

private void button1\_Click(object sender, EventArgs e)

{

upd();

Close();

}

private void Form6\_Edit\_Activated(object sender, EventArgs e)

{

this.izgotovitelTableAdapter.Fill(this.bDDataSet.Izgotovitel);

this.firmaPostavTableAdapter.Fill(this.bDDataSet.FirmaPostav);

}

private void button4\_Click(object sender, EventArgs e)

{

f24.fr2 = false;

f24.fr6 = true;

f24.ShowDialog();

}

private void button5\_Click(object sender, EventArgs e)

{

f25.fr2 = false;

f25.fr6 = true;

f25.ShowDialog();

}

}

}

**Листинг 7 – Модуль класса Form7\_DeleteZapis.cs**

{======================================================================}

*{ класс Form7\_DeleteZapis.cs }*

*{ гр. П-10 };*

*{ Разработчик: Васюхневич П.В. }*

*{ Модифицирован 8 мая 2018г }*

**{----------------------------------------------------------------------}**

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ULS

{

public partial class Form7\_DeleteZapis : Form

{

Form1 frm1;

DataGridView dv;

int focus;

public Form7\_DeleteZapis(Form1 f, DataGridView s, int index)

{

InitializeComponent();

frm1 = f;

dv = s;

focus = index;

}

private void Form7\_DeleteZapis\_Load(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

private void button1\_Click(object sender, EventArgs e)

{

string connStringAcc = string.Format("Provider=Microsoft.Jet.OLEDB.4.0;Data Source={0}", Application.StartupPath + "\\BD.mdb");

OleDbConnection connection = new OleDbConnection(connStringAcc);

connection.Open();

OleDbCommand command = new OleDbCommand();

command.Connection = connection;

int lastfocus = int.Parse(dv.Rows[focus].Cells[11].Value.ToString());

command.CommandText = string.Format("DELETE FROM [Base] WHERE (([id]=" + lastfocus + "))");

command.ExecuteNonQuery();

connection.Close();

Close();

}

}

}

**Листинг 8 – Модуль класса Form8\_AddPreparat.cs**

{======================================================================}

*{ класс Form8\_AddPreparat.cs }*

*{ гр. П-10 };*

*{ Разработчик: Васюхневич П.В. }*

*{ Модифицирован 8 мая 2018г }*

**{----------------------------------------------------------------------}**

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ULS

{

public partial class Form8\_AddPreparat : Form

{

Form1 frm1;

public Form8\_AddPreparat(Form1 f)

{

InitializeComponent();

frm1 = f;

}

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

private void Form8\_AddPreparat\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

string connStringAcc = string.Format("Provider=Microsoft.Jet.OLEDB.4.0;Data Source={0}", Application.StartupPath + "\\BD.mdb");

OleDbConnection connection = new OleDbConnection(connStringAcc);

connection.Open();

OleDbCommand command = new OleDbCommand();

command.Connection = connection;

command.CommandText = "insert into [NamePreparat] (Name, Type) values ('" + textBox1.Text + "','" + comboBox1.Text + "')";

command.ExecuteNonQuery();

connection.Close();

Close();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

if (textBox1.Text != "")

{

button1.Enabled = true;

}

else

button1.Enabled = false;

}

}

}

{======================================================================}

**Листинг 9 – Модуль класса Form9\_DeletePrepapat.cs**

*{ класс Form9\_DeletePrepapat.cs }*

*{ гр. П-10 };*

*{ Разработчик: Васюхневич П.В. }*

*{ Модифицирован 8 мая 2018г }*

**{----------------------------------------------------------------------}**

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ULS

{

public partial class Form9\_DeletePrepapat : Form

{

Form1 frm1;

public Form9\_DeletePrepapat(Form1 f)

{

InitializeComponent();

frm1 = f;

}

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

private void Form9\_DeletePrepapat\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "bDDataSet.NamePreparat". При необходимости она может быть перемещена или удалена.

this.namePreparatTableAdapter.Fill(this.bDDataSet.NamePreparat);

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

dataGridView1.Rows.RemoveAt(dataGridView1.SelectedCells[0].RowIndex);

namePreparatTableAdapter.Update(bDDataSet);

}

catch(Exception)

{ }

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

try

{

namePreparatBindingSource.Position = namePreparatBindingSource.Find("Name", textBox1.Text);

}

catch (Exception)

{ }

}

}

}

**Листинг 10 – Модуль класса Form10\_AddPostavschik.cs**

{======================================================================}

*{ класса Form10\_AddPostavschik.cs }*

*{ гр. П-10 };*

*{ Разработчик: Васюхневич П.В. }*

*{ Модифицирован 8 мая 2018г }*

**{----------------------------------------------------------------------}**

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ULS

{

public partial class Form10\_AddPostavschik : Form

{

Form1 frm1;

public Form10\_AddPostavschik(Form1 F)

{

InitializeComponent();

frm1 = F;

}

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

private void Form10\_AddPostavschik\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

string connStringAcc = string.Format("Provider=Microsoft.Jet.OLEDB.4.0;Data Source={0}", Application.StartupPath + "\\BD.mdb");

OleDbConnection connection = new OleDbConnection(connStringAcc);

connection.Open();

OleDbCommand command = new OleDbCommand();

command.Connection = connection;

command.CommandText = "insert into [FirmaPostav] (Name) values ('" + textBox1.Text + "')";

command.ExecuteNonQuery();

connection.Close();

Close();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

if (textBox1.Text != "")

{

button1.Enabled = true;

}

else

button1.Enabled = false;

}

}

}

**Листинг 11 – Модуль класса Form11\_DeletePostavschik.cs**

{======================================================================}

*{ класса Form11\_DeletePostavschik.cs }*

*{ гр. П-10 };*

*{ Разработчик: Васюхневич П.В. }*

*{ Модифицирован 8 мая 2018г }*

**{----------------------------------------------------------------------}**

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ULS

{

public partial class Form11\_DeletePostavschik : Form

{

Form1 frm1;

public Form11\_DeletePostavschik(Form1 f)

{

InitializeComponent();

frm1 = f;

}

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

private void Form11\_DeletePostavschik\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "bDDataSet.FirmaPostav". При необходимости она может быть перемещена или удалена.

this.firmaPostavTableAdapter.Fill(this.bDDataSet.FirmaPostav);

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

dataGridView1.Rows.RemoveAt(dataGridView1.SelectedCells[0].RowIndex);

firmaPostavTableAdapter.Update(bDDataSet);

}

catch (Exception)

{ }

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

try

{

firmaPostavBindingSource.Position = firmaPostavBindingSource.Find("Name", textBox1.Text);

}

catch (Exception)

{ }

}

}

}

**Листинг 12 – Модуль класса Form12\_AddIzgotovitel.cs**

{======================================================================}

*{ класс Form12\_AddIzgotovitel.cs }*

*{ гр. П-10 };*

*{ Разработчик: Васюхневич П.В. }*

*{ Модифицирован 8 мая 2018г }*

**{----------------------------------------------------------------------}**

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ULS

{

public partial class Form12\_AddIzgotovitel : Form

{

Form1 frm1;

public Form12\_AddIzgotovitel(Form1 f)

{

InitializeComponent();

frm1 = f;

}

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

private void Form12\_AddIzgotovitel\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

string connStringAcc = string.Format("Provider=Microsoft.Jet.OLEDB.4.0;Data Source={0}", Application.StartupPath + "\\BD.mdb");

OleDbConnection connection = new OleDbConnection(connStringAcc);

connection.Open();

OleDbCommand command = new OleDbCommand();

command.Connection = connection;

command.CommandText = "insert into [Izgotovitel] (Name) values ('" + textBox1.Text + "')";

command.ExecuteNonQuery();

connection.Close();

Close();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

if (textBox1.Text != "")

{

button1.Enabled = true;

}

else

button1.Enabled = false;

}

}

}

**Листинг 13 – Модуль класса Form13\_DeleteIzgotovitel.cs**

{======================================================================}

*{ класс Form13\_DeleteIzgotovitel.cs }*

*{ гр. П-10 };*

*{ Разработчик: Васюхневич П.В. }*

*{ Модифицирован 8 мая 2018г }*

**{----------------------------------------------------------------------}**

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ULS

{

public partial class Form13\_DeleteIzgotovitel : Form

{

Form1 frm1;

public Form13\_DeleteIzgotovitel(Form1 f)

{

InitializeComponent();

frm1 = f;

}

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

private void Form13\_DeleteIzgotovitel\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "bDDataSet.Izgotovitel". При необходимости она может быть перемещена или удалена.

this.izgotovitelTableAdapter.Fill(this.bDDataSet.Izgotovitel);

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

dataGridView1.Rows.RemoveAt(dataGridView1.SelectedCells[0].RowIndex);

izgotovitelTableAdapter.Update(bDDataSet);

}

catch (Exception)

{ }

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

try

{

izgotovitelBindingSource.Position = izgotovitelBindingSource.Find("Name", textBox1.Text);

}

catch (Exception)

{ }

}

}

}

**Листинг 14 – Модуль класса Form14\_AddPeremeschenie.cs**

{======================================================================}

*{ класс Form14\_AddPeremeschenie.cs }*

*{ гр. П-10 };*

*{ Разработчик: Васюхневич П.В. }*

*{ Модифицирован 8 мая 2018г }*

**{----------------------------------------------------------------------}**

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ULS

{

public partial class Form14\_AddPeremeschenie : Form

{

Form1 frm1;

public Form14\_AddPeremeschenie(Form1 f)

{

InitializeComponent();

frm1 = f;

}

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

private void Form14\_AddPeremeschenie\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

string connStringAcc = string.Format("Provider=Microsoft.Jet.OLEDB.4.0;Data Source={0}", Application.StartupPath + "\\BD.mdb");

OleDbConnection connection = new OleDbConnection(connStringAcc);

connection.Open();

OleDbCommand command = new OleDbCommand();

command.Connection = connection;

command.CommandText = "insert into [Otdelenie] (Name) values ('" + textBox1.Text + "')";

command.ExecuteNonQuery();

connection.Close();

Close();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

if (textBox1.Text != "")

{

button1.Enabled = true;

}

else

button1.Enabled = false;

}

}

}

**Листинг 15 – Модуль класса Form15\_DeletePeremeschenie.cs**

{======================================================================}

*{ класс Form15\_DeletePeremeschenie.cs}*

*{ гр. П-10 };*

*{ Разработчик: Васюхневич П.В. }*

*{ Модифицирован 8 мая 2018г }*

**{----------------------------------------------------------------------}**

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ULS

{

public partial class Form15\_DeletePeremeschenie : Form

{

Form1 frm1;

public Form15\_DeletePeremeschenie(Form1 f)

{

InitializeComponent();

frm1 = f;

}

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

private void Form15\_DeletePeremeschenie\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "bDDataSet.Otdelenie". При необходимости она может быть перемещена или удалена.

this.otdelenieTableAdapter.Fill(this.bDDataSet.Otdelenie);

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

dataGridView1.Rows.RemoveAt(dataGridView1.SelectedCells[0].RowIndex);

otdelenieTableAdapter.Update(bDDataSet);

}

catch (Exception)

{ }

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

try

{

otdelenieBindingSource.Position = otdelenieBindingSource.Find("Name", textBox1.Text);

}

catch (Exception)

{ }

}

}

}

**Листинг 16 – Модуль класса Form16\_OtOSrokeGodnosti.cs**

{======================================================================}

*{ класс Form16\_OtOSrokeGodnosti.cs }*

*{ гр. П-10 };*

*{ Разработчик: Васюхневич П.В. }*

*{ Модифицирован 8 мая 2018г }*

**{----------------------------------------------------------------------}**

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using Microsoft.Office.Interop.Excel;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Reflection;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Excel = Microsoft.Office.Interop.Excel;

namespace ULS

{

public partial class Form16\_OtOSrokeGodnosti : Form

{

Object Range;

Excel.\_Worksheet workSheet;

Form1 frm1;

public Form16\_OtOSrokeGodnosti(Form1 f)

{

InitializeComponent();

frm1 = f;

}

private void button2\_Click(object sender, EventArgs e)

{

Hide();

}

private void Form16\_OtOSrokeGodnosti\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

ExportToExcel();

Close();

}

private void ExportToExcel()

{

int count = 0;

for (int i = 0; i < frm1.dataGridView1.Rows.Count-1; i++)

{

if (DateTime.Parse(frm1.dataGridView1.Rows[i].Cells[10].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker1.Value.ToShortDateString())) <= 0)

{ count++; }

}

if (count != 0)

{

Excel.Application exApp = new Excel.Application();

exApp.Visible = true;

exApp.Workbooks.Add();

workSheet = (Worksheet)exApp.ActiveSheet;

SetMerge("A1:H1", 3);

SetMerge("A2:H2", 3);

workSheet.Cells[1, 1].EntireRow.Font.Bold = true;

workSheet.Cells[2, 1].EntireRow.Font.Bold = true;

workSheet.Cells[4, 1].EntireRow.Font.Bold = true;

workSheet.Cells[4, 2].EntireRow.Font.Bold = true;

workSheet.Cells[4, 3].EntireRow.Font.Bold = true;

workSheet.Cells[4, 4].EntireRow.Font.Bold = true;

workSheet.Cells[4, 5].EntireRow.Font.Bold = true;

workSheet.Cells[4, 6].EntireRow.Font.Bold = true;

workSheet.Cells[4, 7].EntireRow.Font.Bold = true;

workSheet.Cells[4, 8].EntireRow.Font.Bold = true;

SetBorderStyle("A4:H4", 1);

workSheet.Cells[1, 1] = "Отчет";

workSheet.Cells[2, 1] = "по лекарственным средствам у которых истёк срок годности на " + dateTimePicker1.Value.ToShortDateString() + " года.";

workSheet.Cells[4, 1] = "№";

workSheet.Cells[4, 2] = "Наименование";

workSheet.Cells[4, 3] = "Дата получ.";

workSheet.Cells[4, 4] = "Срок годн.";

workSheet.Cells[4, 5] = "Ед. измерения";

workSheet.Cells[4, 6] = "Колич.";

workSheet.Cells[4, 7] = "Цена за ед.";

workSheet.Cells[4, 8] = "Общая сумма";

workSheet.Cells[count+5, 5] = "ИТОГО:";

workSheet.Cells[count + 5, 5].EntireRow.Font.Bold = true;

SetBorderStyle("E" + (count + 5).ToString() + ":H" + (count + 5).ToString(), 1);

SetColumnWidth("A1", 3);

SetColumnWidth("B1", 32);

SetColumnWidth("C1", 11);

SetColumnWidth("D1", 11);

SetColumnWidth("E1", 14);

SetColumnWidth("F1", 7);

SetColumnWidth("G1", 12);

SetColumnWidth("H1", 13);

int rowExcel = 5;

int col = 0;

int sum = 0;

for (int i = 0; i < frm1.dataGridView1.Rows.Count-1; i++)

{

if (DateTime.Parse(frm1.dataGridView1.Rows[i].Cells[10].Value.ToString()).CompareTo(dateTimePicker1.Value) < 0 || DateTime.Parse(frm1.dataGridView1.Rows[i].Cells[10].Value.ToString()).CompareTo(dateTimePicker1.Value) == 0)

{

SetBorderStyle("A" + (i + 5).ToString() + ":H" + (i + 5).ToString(), 1);

workSheet.Cells[rowExcel, "A"] = frm1.dataGridView1.Rows[i].Cells[11].Value;

workSheet.Cells[rowExcel, "B"] = frm1.dataGridView1.Rows[i].Cells[0].Value;

workSheet.Cells[rowExcel, "C"] = frm1.dataGridView1.Rows[i].Cells[1].Value;

workSheet.Cells[rowExcel, "D"] = frm1.dataGridView1.Rows[i].Cells[10].Value;

workSheet.Cells[rowExcel, "E"] = frm1.dataGridView1.Rows[i].Cells[6].Value;

workSheet.Cells[rowExcel, "F"] = frm1.dataGridView1.Rows[i].Cells[7].Value;

workSheet.Cells[rowExcel, "G"] = frm1.dataGridView1.Rows[i].Cells[8].Value;

workSheet.Cells[rowExcel, "H"] = frm1.dataGridView1.Rows[i].Cells[9].Value;

++rowExcel;

col = col + int.Parse(frm1.dataGridView1.Rows[i].Cells[7].Value.ToString());

sum = sum + int.Parse(frm1.dataGridView1.Rows[i].Cells[9].Value.ToString());

}

}

workSheet.Cells[count + 5, 6] = col.ToString();

workSheet.Cells[count + 5, 8] = sum.ToString();

workSheet.Cells[count + 5, 6].EntireRow.Font.Bold = true;

workSheet.Cells[count + 5, 8].EntireRow.Font.Bold = true;

}

else

{ MessageBox.Show("На данную дату нет просроченных медикаментов!"); }

}

private void SetColumnWidth(string range, double Width)//TODO: Ширина столбца

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Width };

Range.GetType().InvokeMember("ColumnWidth", BindingFlags.SetProperty, null, Range, args);

}

private void SetBorderStyle(string range, int Style)//TODO: Стиль границ ячейки

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Style };

object Borders = Range.GetType().InvokeMember("Borders", BindingFlags.GetProperty, null, Range, null);

Borders = Range.GetType().InvokeMember("LineStyle", BindingFlags.SetProperty, null, Borders, args);

}

private void SetWrapText(string range, bool Value)//TODO: Перенос по словам

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Value };

Range.GetType().InvokeMember("WrapText", BindingFlags.SetProperty, null, Range, args);

}

private void SetRowHeight(string range, double Height)//TODO: Высота строк

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty,

null, workSheet, new object[] { range });

object[] args = new object[] { Height };

Range.GetType().InvokeMember("RowHeight", BindingFlags.SetProperty, null, Range, args);

}

private void SetMerge(string range, int Alignment)//TODO: Выравнивание текста в ячейке

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Alignment };

Range.GetType().InvokeMember("MergeCells", BindingFlags.SetProperty, null, Range, new object[] { true });

Range.GetType().InvokeMember("HorizontalAlignment", BindingFlags.SetProperty, null, Range, args);

}

private void SetFont(string range, Microsoft.Office.Interop.Excel.Font font)

{

//Range.Font.Name

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty,

null, workSheet, new object[] { range });

object Font = Range.GetType().InvokeMember("Font", BindingFlags.GetProperty,

null, Range, null);

Range.GetType().InvokeMember("Name", BindingFlags.SetProperty, null,

Font, new object[] { font.Name });

Range.GetType().InvokeMember("Size", BindingFlags.SetProperty, null,

Font, new object[] { font.Size });

}

}

}

**Листинг 17 – Модуль класса Form17\_OtORashodMed.cs**

{======================================================================}

*{ класс Form17\_OtORashodMed.cs }*

*{ гр. П-10 };*

*{ Разработчик: Васюхневич П.В. }*

*{ Модифицирован 8 мая 2018г }*

**{----------------------------------------------------------------------}**

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using Microsoft.Office.Interop.Excel;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Reflection;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Excel = Microsoft.Office.Interop.Excel;

namespace ULS

{

public partial class Form17\_OtORashodMed : Form

{

Object Range;

Excel.\_Worksheet workSheet;

Form1 frm1;

public Form17\_OtORashodMed(Form1 f)

{

InitializeComponent();

frm1 = f;

}

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

private void Form17\_OtORashodMed\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "bDDataSet.Otdelenie". При необходимости она может быть перемещена или удалена.

this.otdelenieTableAdapter.Fill(this.bDDataSet.Otdelenie);

// TODO: данная строка кода позволяет загрузить данные в таблицу "bDDataSet.Rashod". При необходимости она может быть перемещена или удалена.

this.rashodTableAdapter.Fill(this.bDDataSet.Rashod);

}

private void button1\_Click(object sender, EventArgs e)

{

ExportToExcel();

Close();

}

private void ExportToExcel()

{

int count = 0;

for (int j = 0; j < frm1.dataGridView1.Rows.Count - 1; j++)

for (int i = 0; i < dataGridView1.Rows.Count - 1; i++)

if (int.Parse(frm1.dataGridView1.Rows[j].Cells[11].Value.ToString()) == int.Parse(dataGridView1.Rows[i].Cells[1].Value.ToString()))

if (DateTime.Parse(dataGridView1.Rows[i].Cells[12].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker1.Value.ToShortDateString()))>=0 && DateTime.Parse(dataGridView1.Rows[i].Cells[12].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker2.Value.ToShortDateString()))<=0)

count++;

if (count != 0)

{

Excel.Application exApp = new Excel.Application();

exApp.Visible = true;

exApp.Workbooks.Add();

workSheet = (Worksheet)exApp.ActiveSheet;

SetMerge("A1:K1", 3);

SetMerge("A2:K2", 3);

workSheet.Cells[1, 1].EntireRow.Font.Bold = true;

workSheet.Cells[2, 1].EntireRow.Font.Bold = true;

workSheet.Cells[4, 1].EntireRow.Font.Bold = true;

workSheet.Cells[4, 2].EntireRow.Font.Bold = true;

workSheet.Cells[4, 3].EntireRow.Font.Bold = true;

workSheet.Cells[4, 4].EntireRow.Font.Bold = true;

workSheet.Cells[4, 5].EntireRow.Font.Bold = true;

workSheet.Cells[4, 6].EntireRow.Font.Bold = true;

workSheet.Cells[4, 7].EntireRow.Font.Bold = true;

workSheet.Cells[4, 8].EntireRow.Font.Bold = true;

workSheet.Cells[4, 9].EntireRow.Font.Bold = true;

workSheet.Cells[4, 10].EntireRow.Font.Bold = true;

workSheet.Cells[4, 11].EntireRow.Font.Bold = true;

SetBorderStyle("A4:K4", 1);

workSheet.Cells[1, 1] = "Общий отчет";

workSheet.Cells[2, 1] = "израсходованных лекарственных средств с " + dateTimePicker1.Value.ToShortDateString() + " по " + dateTimePicker2.Value.ToShortDateString() + " года.";

workSheet.Cells[4, 1] = "№";

workSheet.Cells[4, 2] = "Наименование";

workSheet.Cells[4, 3] = "Фирма-поставщик";

workSheet.Cells[4, 4] = "Назв. документа";

workSheet.Cells[4, 5] = "Фирма-изготовитель";

workSheet.Cells[4, 6] = "Серия препарата";

workSheet.Cells[4, 7] = "Дата получ.";

workSheet.Cells[4, 8] = "Ед. измерения";

workSheet.Cells[4, 9] = "Колич.";

workSheet.Cells[4, 10] = "Цена за ед.";

workSheet.Cells[4, 11] = "Общая сумма";

SetColumnWidth("A1", 3);

SetColumnWidth("B1", 32);

SetColumnWidth("C1", 19);

SetColumnWidth("D1", 19);

SetColumnWidth("E1", 22);

SetColumnWidth("F1", 15);

SetColumnWidth("G1", 12);

SetColumnWidth("H1", 16);

SetColumnWidth("I1", 9);

SetColumnWidth("J1", 11);

SetColumnWidth("K1", 13);

int rowExcel = 5;

int obcol = 0;

int obsum = 0;

int col;

int ocount = 0;

for (int j = 0; j < frm1.dataGridView1.Rows.Count - 1; j++)

{

col = 0;

for (int i = 0; i < dataGridView1.Rows.Count - 1; i++)

{

if (int.Parse(frm1.dataGridView1.Rows[j].Cells[11].Value.ToString()) == int.Parse(dataGridView1.Rows[i].Cells[1].Value.ToString()))

{

if (DateTime.Parse(dataGridView1.Rows[i].Cells[12].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker1.Value.ToShortDateString())) >= 0 && DateTime.Parse(dataGridView1.Rows[i].Cells[12].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker2.Value.ToShortDateString())) <= 0)

{

col = col + int.Parse(dataGridView1.Rows[i].Cells[9].Value.ToString());

}

}

}

if (col > 0)

{

SetBorderStyle("A" + (rowExcel).ToString() + ":K" + (rowExcel).ToString(), 1);

workSheet.Cells[rowExcel, "A"] = frm1.dataGridView1.Rows[j].Cells[11].Value;

workSheet.Cells[rowExcel, "B"] = frm1.dataGridView1.Rows[j].Cells[0].Value;

workSheet.Cells[rowExcel, "C"] = frm1.dataGridView1.Rows[j].Cells[2].Value;

workSheet.Cells[rowExcel, "D"] = frm1.dataGridView1.Rows[j].Cells[3].Value;

workSheet.Cells[rowExcel, "E"] = frm1.dataGridView1.Rows[j].Cells[4].Value;

workSheet.Cells[rowExcel, "F"] = frm1.dataGridView1.Rows[j].Cells[5].Value;

workSheet.Cells[rowExcel, "G"] = frm1.dataGridView1.Rows[j].Cells[1].Value;

workSheet.Cells[rowExcel, "H"] = frm1.dataGridView1.Rows[j].Cells[6].Value;

workSheet.Cells[rowExcel, "I"] = col;

workSheet.Cells[rowExcel, "J"] = frm1.dataGridView1.Rows[j].Cells[8].Value;

workSheet.Cells[rowExcel, "K"] = col \* (int.Parse(frm1.dataGridView1.Rows[j].Cells[8].Value.ToString()));

++rowExcel;

obcol = obcol + col;

obsum = obsum + (col \* (int.Parse(frm1.dataGridView1.Rows[j].Cells[8].Value.ToString())));

ocount++;

}

}

SetBorderStyle("H" + (ocount + 5).ToString() + ":K" + (ocount + 5).ToString(), 1);

workSheet.Cells[ocount + 5, 8] = "ИТОГО:";

workSheet.Cells[ocount + 5, 8].EntireRow.Font.Bold = true;

workSheet.Cells[ocount + 5, 9] = obcol.ToString();

workSheet.Cells[ocount + 5, 11] = obsum.ToString();

workSheet.Cells[ocount + 5, 9].EntireRow.Font.Bold = true;

workSheet.Cells[ocount + 5, 11].EntireRow.Font.Bold = true;

}

else

{ MessageBox.Show("За этот период нет израсходованных медикаментов!"); }

}

private void SetColumnWidth(string range, double Width)//TODO: Ширина столбца

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Width };

Range.GetType().InvokeMember("ColumnWidth", BindingFlags.SetProperty, null, Range, args);

}

private void SetBorderStyle(string range, int Style)//TODO: Стиль границ ячейки

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Style };

object Borders = Range.GetType().InvokeMember("Borders", BindingFlags.GetProperty, null, Range, null);

Borders = Range.GetType().InvokeMember("LineStyle", BindingFlags.SetProperty, null, Borders, args);

}

private void SetWrapText(string range, bool Value)//TODO: Перенос по словам

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Value };

Range.GetType().InvokeMember("WrapText", BindingFlags.SetProperty, null, Range, args);

}

private void SetRowHeight(string range, double Height)//TODO: Высота строк

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty,

null, workSheet, new object[] { range });

object[] args = new object[] { Height };

Range.GetType().InvokeMember("RowHeight", BindingFlags.SetProperty, null, Range, args);

}

private void SetMerge(string range, int Alignment)//TODO: Выравнивание текста в ячейке

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Alignment };

Range.GetType().InvokeMember("MergeCells", BindingFlags.SetProperty, null, Range, new object[] { true });

Range.GetType().InvokeMember("HorizontalAlignment", BindingFlags.SetProperty, null, Range, args);

}

private void SetFont(string range, Microsoft.Office.Interop.Excel.Font font)

{

//Range.Font.Name

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty,

null, workSheet, new object[] { range });

object Font = Range.GetType().InvokeMember("Font", BindingFlags.GetProperty,

null, Range, null);

Range.GetType().InvokeMember("Name", BindingFlags.SetProperty, null,

Font, new object[] { font.Name });

Range.GetType().InvokeMember("Size", BindingFlags.SetProperty, null,

Font, new object[] { font.Size });

}

}

}

**Листинг 18 – Модуль класса Form18\_OtoRashodeMedSDate.cs**

{======================================================================}

*{ класс Form18\_OtoRashodeMedSDate.cs }*

*{ гр. П-10 };*

*{ Разработчик: Васюхневич П.В. }*

*{ Модифицирован 8 мая 2018г }*

**{----------------------------------------------------------------------}**

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using Microsoft.Office.Interop.Excel;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Reflection;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Excel = Microsoft.Office.Interop.Excel;

namespace ULS

{

public partial class Form18\_OtoRashodeMedSDate : Form

{

Object Range;

Excel.\_Worksheet workSheet;

Form1 frm1;

public Form18\_OtoRashodeMedSDate(Form1 f)

{

InitializeComponent();

frm1 = f;

}

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

private void Form18\_OtoRashodeMedSDate\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "bDDataSet.Rashod". При необходимости она может быть перемещена или удалена.

this.rashodTableAdapter.Fill(this.bDDataSet.Rashod);

}

private void ExportToExcel()

{

int count = 0;

for (int j = 0; j < frm1.dataGridView1.Rows.Count - 1; j++)

for (int i = 0; i < dataGridView1.Rows.Count - 1; i++)

if (int.Parse(frm1.dataGridView1.Rows[j].Cells[11].Value.ToString()) == int.Parse(dataGridView1.Rows[i].Cells[1].Value.ToString()))

if (DateTime.Parse(dataGridView1.Rows[i].Cells[12].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker1.Value.ToShortDateString())) >= 0 && DateTime.Parse(dataGridView1.Rows[i].Cells[12].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker2.Value.ToShortDateString())) <= 0)

count++;

if (count != 0)

{

Excel.Application exApp = new Excel.Application();

exApp.Visible = true;

exApp.Workbooks.Add();

workSheet = (Worksheet)exApp.ActiveSheet;

SetMerge("A1:L1", 3);

SetMerge("A2:L2", 3);

workSheet.Cells[1, 1].EntireRow.Font.Bold = true;

workSheet.Cells[2, 1].EntireRow.Font.Bold = true;

workSheet.Cells[4, 1].EntireRow.Font.Bold = true;

workSheet.Cells[4, 2].EntireRow.Font.Bold = true;

workSheet.Cells[4, 3].EntireRow.Font.Bold = true;

workSheet.Cells[4, 4].EntireRow.Font.Bold = true;

workSheet.Cells[4, 5].EntireRow.Font.Bold = true;

workSheet.Cells[4, 6].EntireRow.Font.Bold = true;

workSheet.Cells[4, 7].EntireRow.Font.Bold = true;

workSheet.Cells[4, 8].EntireRow.Font.Bold = true;

workSheet.Cells[4, 9].EntireRow.Font.Bold = true;

workSheet.Cells[4, 10].EntireRow.Font.Bold = true;

workSheet.Cells[4, 11].EntireRow.Font.Bold = true;

workSheet.Cells[4, 12].EntireRow.Font.Bold = true;

SetBorderStyle("A4:L4", 1);

workSheet.Cells[1, 1] = "Общий отчет";

workSheet.Cells[2, 1] = "израсходованных лекарственных средств с " + dateTimePicker1.Value.ToShortDateString() + " по " + dateTimePicker2.Value.ToShortDateString() + " года.";

workSheet.Cells[4, 1] = "№";

workSheet.Cells[4, 2] = "Наименование";

workSheet.Cells[4, 3] = "Фирма-поставщик";

workSheet.Cells[4, 4] = "Назв. документа";

workSheet.Cells[4, 5] = "Фирма-изготовитель";

workSheet.Cells[4, 6] = "Серия препарата";

workSheet.Cells[4, 7] = "Дата получ.";

workSheet.Cells[4, 8] = "Дата расх.";//nen

workSheet.Cells[4, 9] = "Ед. измерения";

workSheet.Cells[4, 10] = "Колич.";

workSheet.Cells[4, 11] = "Цена за ед.";

workSheet.Cells[4, 12] = "Общая сумма";

SetColumnWidth("A1", 3);

SetColumnWidth("B1", 32);

SetColumnWidth("C1", 19);

SetColumnWidth("D1", 19);

SetColumnWidth("E1", 22);

SetColumnWidth("F1", 15);

SetColumnWidth("G1", 12);

SetColumnWidth("H1", 12);

SetColumnWidth("I1", 16);

SetColumnWidth("J1", 9);

SetColumnWidth("K1", 11);

SetColumnWidth("L1", 13);

int rowExcel = 5;

int obcol = 0;

int obsum = 0;

int col;

int ocount = 0;

for (int j = 0; j < frm1.dataGridView1.Rows.Count - 1; j++)

{

col = 0;

for (int i = 0; i < dataGridView1.Rows.Count - 1; i++)

{

if (int.Parse(frm1.dataGridView1.Rows[j].Cells[11].Value.ToString()) == int.Parse(dataGridView1.Rows[i].Cells[1].Value.ToString()))

{

if (DateTime.Parse(dataGridView1.Rows[i].Cells[12].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker1.Value.ToShortDateString())) >= 0 && DateTime.Parse(dataGridView1.Rows[i].Cells[12].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker2.Value.ToShortDateString())) <= 0)

{

col = col + int.Parse(dataGridView1.Rows[i].Cells[9].Value.ToString());

SetBorderStyle("A" + (rowExcel).ToString() + ":L" + (rowExcel).ToString(), 1);

workSheet.Cells[rowExcel, "A"] = frm1.dataGridView1.Rows[j].Cells[11].Value;

workSheet.Cells[rowExcel, "B"] = frm1.dataGridView1.Rows[j].Cells[0].Value;

workSheet.Cells[rowExcel, "C"] = frm1.dataGridView1.Rows[j].Cells[2].Value;

workSheet.Cells[rowExcel, "D"] = frm1.dataGridView1.Rows[j].Cells[3].Value;

workSheet.Cells[rowExcel, "E"] = frm1.dataGridView1.Rows[j].Cells[4].Value;

workSheet.Cells[rowExcel, "F"] = frm1.dataGridView1.Rows[j].Cells[5].Value;

workSheet.Cells[rowExcel, "G"] = frm1.dataGridView1.Rows[j].Cells[1].Value;

workSheet.Cells[rowExcel, "H"] = dataGridView1.Rows[i].Cells[12].Value;

workSheet.Cells[rowExcel, "I"] = frm1.dataGridView1.Rows[j].Cells[6].Value;

workSheet.Cells[rowExcel, "J"] = col;

workSheet.Cells[rowExcel, "K"] = frm1.dataGridView1.Rows[j].Cells[8].Value;

workSheet.Cells[rowExcel, "L"] = col \* (int.Parse(frm1.dataGridView1.Rows[j].Cells[8].Value.ToString()));

++rowExcel;

obcol = obcol + col;

obsum = obsum + (col \* (int.Parse(frm1.dataGridView1.Rows[j].Cells[8].Value.ToString())));

ocount++;

}

}

}

}

SetBorderStyle("H" + (ocount + 5).ToString() + ":L" + (ocount + 5).ToString(), 1);

workSheet.Cells[ocount + 5, 9] = "ИТОГО:";

workSheet.Cells[ocount + 5, 9].EntireRow.Font.Bold = true;

workSheet.Cells[ocount + 5, 10] = obcol.ToString();

workSheet.Cells[ocount + 5, 12] = obsum.ToString();

workSheet.Cells[ocount + 5, 10].EntireRow.Font.Bold = true;

workSheet.Cells[ocount + 5, 12].EntireRow.Font.Bold = true;

}

else

{ MessageBox.Show("За этот период нет израсходованных медикаментов!"); }

}

private void SetColumnWidth(string range, double Width)//TODO: Ширина столбца

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Width };

Range.GetType().InvokeMember("ColumnWidth", BindingFlags.SetProperty, null, Range, args);

}

private void SetBorderStyle(string range, int Style)//TODO: Стиль границ ячейки

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Style };

object Borders = Range.GetType().InvokeMember("Borders", BindingFlags.GetProperty, null, Range, null);

Borders = Range.GetType().InvokeMember("LineStyle", BindingFlags.SetProperty, null, Borders, args);

}

private void SetWrapText(string range, bool Value)//TODO: Перенос по словам

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Value };

Range.GetType().InvokeMember("WrapText", BindingFlags.SetProperty, null, Range, args);

}

private void SetRowHeight(string range, double Height)//TODO: Высота строк

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty,

null, workSheet, new object[] { range });

object[] args = new object[] { Height };

Range.GetType().InvokeMember("RowHeight", BindingFlags.SetProperty, null, Range, args);

}

private void SetMerge(string range, int Alignment)//TODO: Выравнивание текста в ячейке

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Alignment };

Range.GetType().InvokeMember("MergeCells", BindingFlags.SetProperty, null, Range, new object[] { true });

Range.GetType().InvokeMember("HorizontalAlignment", BindingFlags.SetProperty, null, Range, args);

}

private void SetFont(string range, Microsoft.Office.Interop.Excel.Font font)

{

//Range.Font.Name

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty,

null, workSheet, new object[] { range });

object Font = Range.GetType().InvokeMember("Font", BindingFlags.GetProperty,

null, Range, null);

Range.GetType().InvokeMember("Name", BindingFlags.SetProperty, null,

Font, new object[] { font.Name });

Range.GetType().InvokeMember("Size", BindingFlags.SetProperty, null,

Font, new object[] { font.Size });

}

private void button1\_Click(object sender, EventArgs e)

{

ExportToExcel();

Close();

}

}

}

**Листинг 19 – Модуль класса Form19\_OtOPriobretenii.cs**

{======================================================================}

*{класс Form19\_OtOPriobretenii.cs}*

*{ гр. П-10 };*

*{ Разработчик: Васюхневич П.В.}*

*{ Модифицирован 8 мая 2018г }*

**{----------------------------------------------------------------------}**

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using Microsoft.Office.Interop.Excel;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Reflection;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Excel = Microsoft.Office.Interop.Excel;

namespace ULS

{

public partial class Form19\_OtOPriobretenii : Form

{

Object Range;

Excel.\_Worksheet workSheet;

Form1 frm1;

public Form19\_OtOPriobretenii(Form1 f)

{

InitializeComponent();

frm1 = f;

}

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

private void Form19\_OtOPriobretenii\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

ExportToExcel();

Close();

}

private void ExportToExcel()

{

int count = 0;

for (int j = 0; j < frm1.dataGridView1.Rows.Count - 1; j++)

if (DateTime.Parse(frm1.dataGridView1.Rows[j].Cells[1].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker1.Value.ToShortDateString())) >= 0 && DateTime.Parse(frm1.dataGridView1.Rows[j].Cells[1].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker2.Value.ToShortDateString())) <= 0)

count++;

if (count != 0)

{

Excel.Application exApp = new Excel.Application();

exApp.Visible = true;

exApp.Workbooks.Add();

workSheet = (Worksheet)exApp.ActiveSheet;

SetMerge("A1:L1", 3);

SetMerge("A2:L2", 3);

workSheet.Cells[1, 1].EntireRow.Font.Bold = true;

workSheet.Cells[2, 1].EntireRow.Font.Bold = true;

workSheet.Cells[4, 1].EntireRow.Font.Bold = true;

workSheet.Cells[4, 2].EntireRow.Font.Bold = true;

workSheet.Cells[4, 3].EntireRow.Font.Bold = true;

workSheet.Cells[4, 4].EntireRow.Font.Bold = true;

workSheet.Cells[4, 5].EntireRow.Font.Bold = true;

workSheet.Cells[4, 6].EntireRow.Font.Bold = true;

workSheet.Cells[4, 7].EntireRow.Font.Bold = true;

workSheet.Cells[4, 8].EntireRow.Font.Bold = true;

workSheet.Cells[4, 9].EntireRow.Font.Bold = true;

workSheet.Cells[4, 10].EntireRow.Font.Bold = true;

workSheet.Cells[4, 11].EntireRow.Font.Bold = true;

workSheet.Cells[4, 12].EntireRow.Font.Bold = true;

SetBorderStyle("A4:L4", 1);

workSheet.Cells[1, 1] = "Отчет";

workSheet.Cells[2, 1] = "приобретённых лекарственных средств с " + dateTimePicker1.Value.ToShortDateString() + " по " + dateTimePicker2.Value.ToShortDateString() + " года.";

workSheet.Cells[4, 1] = "№";

workSheet.Cells[4, 2] = "Наименование";

workSheet.Cells[4, 3] = "Фирма-поставщик";

workSheet.Cells[4, 4] = "Назв. документа";

workSheet.Cells[4, 5] = "Фирма-изготовитель";

workSheet.Cells[4, 6] = "Серия препарата";

workSheet.Cells[4, 7] = "Дата получ.";

workSheet.Cells[4, 8] = "Срок годн.";//nen

workSheet.Cells[4, 9] = "Ед. измерения";

workSheet.Cells[4, 10] = "Колич.";

workSheet.Cells[4, 11] = "Цена за ед.";

workSheet.Cells[4, 12] = "Общая сумма";

SetColumnWidth("A1", 3);

SetColumnWidth("B1", 32);

SetColumnWidth("C1", 19);

SetColumnWidth("D1", 19);

SetColumnWidth("E1", 22);

SetColumnWidth("F1", 15);

SetColumnWidth("G1", 12);

SetColumnWidth("H1", 12);

SetColumnWidth("I1", 16);

SetColumnWidth("J1", 9);

SetColumnWidth("K1", 11);

SetColumnWidth("L1", 13);

int rowExcel = 5;

int obsum = 0;

int col = 0;

for (int j = 0; j < frm1.dataGridView1.Rows.Count - 1; j++)

if (DateTime.Parse(frm1.dataGridView1.Rows[j].Cells[1].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker1.Value.ToShortDateString())) >= 0 && DateTime.Parse(frm1.dataGridView1.Rows[j].Cells[1].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker2.Value.ToShortDateString())) <= 0)

{

SetBorderStyle("A" + (rowExcel).ToString() + ":L" + (rowExcel).ToString(), 1);

workSheet.Cells[rowExcel, "A"] = frm1.dataGridView1.Rows[j].Cells[11].Value;

workSheet.Cells[rowExcel, "B"] = frm1.dataGridView1.Rows[j].Cells[0].Value;

workSheet.Cells[rowExcel, "C"] = frm1.dataGridView1.Rows[j].Cells[2].Value;

workSheet.Cells[rowExcel, "D"] = frm1.dataGridView1.Rows[j].Cells[3].Value;

workSheet.Cells[rowExcel, "E"] = frm1.dataGridView1.Rows[j].Cells[4].Value;

workSheet.Cells[rowExcel, "F"] = frm1.dataGridView1.Rows[j].Cells[5].Value;

workSheet.Cells[rowExcel, "G"] = frm1.dataGridView1.Rows[j].Cells[1].Value;

workSheet.Cells[rowExcel, "H"] = frm1.dataGridView1.Rows[j].Cells[10].Value;

workSheet.Cells[rowExcel, "I"] = frm1.dataGridView1.Rows[j].Cells[6].Value;

workSheet.Cells[rowExcel, "J"] = frm1.dataGridView1.Rows[j].Cells[12].Value;

workSheet.Cells[rowExcel, "K"] = frm1.dataGridView1.Rows[j].Cells[8].Value;

workSheet.Cells[rowExcel, "L"] = int.Parse(frm1.dataGridView1.Rows[j].Cells[12].Value.ToString()) \* int.Parse(frm1.dataGridView1.Rows[j].Cells[8].Value.ToString());

++rowExcel;

obsum = obsum + int.Parse(frm1.dataGridView1.Rows[j].Cells[12].Value.ToString()) \* int.Parse(frm1.dataGridView1.Rows[j].Cells[8].Value.ToString());

col = col + int.Parse(frm1.dataGridView1.Rows[j].Cells[12].Value.ToString());

}

SetBorderStyle("H" + (count + 5).ToString() + ":L" + (count + 5).ToString(), 1);

workSheet.Cells[count + 5, 9] = "ИТОГО:";

workSheet.Cells[count + 5, 9].EntireRow.Font.Bold = true;

workSheet.Cells[count + 5, 10] = col.ToString();

workSheet.Cells[count + 5, 12] = obsum.ToString();

workSheet.Cells[count + 5, 10].EntireRow.Font.Bold = true;

workSheet.Cells[count + 5, 12].EntireRow.Font.Bold = true;

}

else

{ MessageBox.Show("За этот период нет приобретенных медикаментов!"); }

}

private void SetColumnWidth(string range, double Width)//TODO: Ширина столбца

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Width };

Range.GetType().InvokeMember("ColumnWidth", BindingFlags.SetProperty, null, Range, args);

}

private void SetBorderStyle(string range, int Style)//TODO: Стиль границ ячейки

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Style };

object Borders = Range.GetType().InvokeMember("Borders", BindingFlags.GetProperty, null, Range, null);

Borders = Range.GetType().InvokeMember("LineStyle", BindingFlags.SetProperty, null, Borders, args);

}

private void SetWrapText(string range, bool Value)//TODO: Перенос по словам

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Value };

Range.GetType().InvokeMember("WrapText", BindingFlags.SetProperty, null, Range, args);

}

private void SetRowHeight(string range, double Height)//TODO: Высота строк

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty,

null, workSheet, new object[] { range });

object[] args = new object[] { Height };

Range.GetType().InvokeMember("RowHeight", BindingFlags.SetProperty, null, Range, args);

}

private void SetMerge(string range, int Alignment)//TODO: Выравнивание текста в ячейке

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Alignment };

Range.GetType().InvokeMember("MergeCells", BindingFlags.SetProperty, null, Range, new object[] { true });

Range.GetType().InvokeMember("HorizontalAlignment", BindingFlags.SetProperty, null, Range, args);

}

private void SetFont(string range, Microsoft.Office.Interop.Excel.Font font)

{

//Range.Font.Name

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty,

null, workSheet, new object[] { range });

object Font = Range.GetType().InvokeMember("Font", BindingFlags.GetProperty,

null, Range, null);

Range.GetType().InvokeMember("Name", BindingFlags.SetProperty, null,

Font, new object[] { font.Name });

Range.GetType().InvokeMember("Size", BindingFlags.SetProperty, null,

Font, new object[] { font.Size });

}

}

}

**Листинг 20 – Модуль класса Form20OtPoVsemOtdeleniam.cs**

{======================================================================}

*{класс Form20OtPoVsemOtdeleniam.cs}*

*{ гр. П-10 };*

*{ Разработчик: Васюхневич П.В.}*

*{ Модифицирован 8 мая 2018г }*

**{----------------------------------------------------------------------}**

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using Microsoft.Office.Interop.Excel;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Reflection;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Excel = Microsoft.Office.Interop.Excel;

namespace ULS

{

public partial class Form20OtPoVsemOtdeleniam : Form

{

Object Range;

Excel.\_Worksheet workSheet;

Form1 frm1;

public Form20OtPoVsemOtdeleniam(Form1 f)

{

InitializeComponent();

frm1 = f;

}

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

private void Form20OtPoVsemOtdeleniam\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "bDDataSet.Otdelenie". При необходимости она может быть перемещена или удалена.

this.otdelenieTableAdapter.Fill(this.bDDataSet.Otdelenie);

// TODO: данная строка кода позволяет загрузить данные в таблицу "bDDataSet.Rashod". При необходимости она может быть перемещена или удалена.

this.rashodTableAdapter.Fill(this.bDDataSet.Rashod);

}

private void ExportToExcel()

{

int count = 0;

for (int u = 0; u < dataGridView2.Rows.Count - 1; u++)

for (int j = 0; j < frm1.dataGridView1.Rows.Count - 1; j++)

for (int i = 0; i < dataGridView1.Rows.Count - 1; i++)

if (dataGridView2.Rows[u].Cells[1].Value.ToString() == dataGridView1.Rows[i].Cells[13].Value.ToString())

if (int.Parse(frm1.dataGridView1.Rows[j].Cells[11].Value.ToString()) == int.Parse(dataGridView1.Rows[i].Cells[1].Value.ToString()))

if (DateTime.Parse(dataGridView1.Rows[i].Cells[12].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker1.Value.ToShortDateString())) >= 0 && DateTime.Parse(dataGridView1.Rows[i].Cells[12].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker2.Value.ToShortDateString())) <= 0)

count++;

if (count != 0)

{

Excel.Application exApp = new Excel.Application();

exApp.Visible = true;

exApp.Workbooks.Add();

workSheet = (Worksheet)exApp.ActiveSheet;

SetMerge("A1:H1", 3);

SetMerge("A2:H2", 3);

workSheet.Cells[1, 1].EntireRow.Font.Bold = true;

workSheet.Cells[2, 1].EntireRow.Font.Bold = true;

workSheet.Cells[1, 1] = "Отчёт израсходованных лекарственных средств";

workSheet.Cells[2, 1] = "с " + dateTimePicker1.Value.ToShortDateString() + " по " + dateTimePicker2.Value.ToShortDateString() + " года для всех отделений.";

SetColumnWidth("A1", 3);

SetColumnWidth("B1", 32);

SetColumnWidth("C1", 19);

SetColumnWidth("D1", 19);

SetColumnWidth("E1", 22);

SetColumnWidth("F1", 15);

SetColumnWidth("G1", 12);

SetColumnWidth("H1", 16);

int rowExcel = 6;

int obcol = 0;

int obsum = 0;

int col;

for (int u = 0; u < dataGridView2.Rows.Count - 1; u++)

{

SetMerge("A" + (rowExcel - 2).ToString() + ":H" + (rowExcel - 2).ToString(), 3);

workSheet.Cells[(rowExcel - 2), 1].EntireRow.Font.Bold = true;

workSheet.Cells[(rowExcel - 2), 1] = dataGridView2.Rows[u].Cells[1].Value.ToString();

int xcv = rowExcel - 1;

workSheet.Cells[xcv, 1] = "№";

workSheet.Cells[xcv, 2] = "Наименование";

workSheet.Cells[xcv, 3] = "Фирма-поставщик";

workSheet.Cells[xcv, 4] = "Фирма-изготовитель";

workSheet.Cells[xcv, 5] = "Ед. измерения";

workSheet.Cells[xcv, 6] = "Колич.";

workSheet.Cells[xcv, 7] = "Цена за ед.";

workSheet.Cells[xcv, 8] = "Общая сумма";

workSheet.Cells[xcv, 1].EntireRow.Font.Bold = true;

workSheet.Cells[xcv, 2].EntireRow.Font.Bold = true;

workSheet.Cells[xcv, 3].EntireRow.Font.Bold = true;

workSheet.Cells[xcv, 4].EntireRow.Font.Bold = true;

workSheet.Cells[xcv, 5].EntireRow.Font.Bold = true;

workSheet.Cells[xcv, 6].EntireRow.Font.Bold = true;

workSheet.Cells[xcv, 7].EntireRow.Font.Bold = true;

workSheet.Cells[xcv, 8].EntireRow.Font.Bold = true;

SetColumnWidth("A" + (xcv).ToString(), 3);

SetColumnWidth("A" + (xcv).ToString(), 32);

SetColumnWidth("A" + (xcv).ToString(), 19);

SetColumnWidth("A" + (xcv).ToString(), 19);

SetColumnWidth("A" + (xcv).ToString(), 22);

SetColumnWidth("A" + (xcv).ToString(), 15);

SetColumnWidth("A" + (xcv).ToString(), 12);

SetColumnWidth("A" + (xcv).ToString(), 16);

SetBorderStyle("A" + (xcv).ToString() + ":H" + (xcv).ToString(), 1);

for (int j = 0; j < frm1.dataGridView1.Rows.Count - 1; j++)

{

col = 0;

for (int i = 0; i < dataGridView1.Rows.Count - 1; i++)

{

if (dataGridView2.Rows[u].Cells[1].Value.ToString() == dataGridView1.Rows[i].Cells[13].Value.ToString())

{

if (int.Parse(frm1.dataGridView1.Rows[j].Cells[11].Value.ToString()) == int.Parse(dataGridView1.Rows[i].Cells[1].Value.ToString()))

{

if (DateTime.Parse(dataGridView1.Rows[i].Cells[12].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker1.Value.ToShortDateString())) >= 0 && DateTime.Parse(dataGridView1.Rows[i].Cells[12].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker2.Value.ToShortDateString())) <= 0)

{

col = col + int.Parse(dataGridView1.Rows[i].Cells[9].Value.ToString());

}

}

}

}

if (col > 0)

{

SetBorderStyle("A" + (rowExcel).ToString() + ":H" + (rowExcel).ToString(), 1);

workSheet.Cells[rowExcel, "A"] = frm1.dataGridView1.Rows[j].Cells[11].Value;

workSheet.Cells[rowExcel, "B"] = frm1.dataGridView1.Rows[j].Cells[0].Value;

workSheet.Cells[rowExcel, "C"] = frm1.dataGridView1.Rows[j].Cells[2].Value;

workSheet.Cells[rowExcel, "D"] = frm1.dataGridView1.Rows[j].Cells[4].Value;

workSheet.Cells[rowExcel, "E"] = frm1.dataGridView1.Rows[j].Cells[6].Value;

workSheet.Cells[rowExcel, "F"] = col;

workSheet.Cells[rowExcel, "G"] = frm1.dataGridView1.Rows[j].Cells[8].Value;

workSheet.Cells[rowExcel, "H"] = (col \* int.Parse(frm1.dataGridView1.Rows[j].Cells[8].Value.ToString()));

++rowExcel;

obcol = obcol + col;

obsum = obsum + (col \* (int.Parse(frm1.dataGridView1.Rows[j].Cells[8].Value.ToString())));

}

}

SetBorderStyle("E" + (rowExcel).ToString() + ":H" + (rowExcel).ToString(), 1);

workSheet.Cells[rowExcel, 5] = "ИТОГО:";

workSheet.Cells[rowExcel, 6] = obcol.ToString();

workSheet.Cells[rowExcel, 8] = obsum.ToString();

workSheet.Cells[rowExcel, 6].EntireRow.Font.Bold = true;

workSheet.Cells[rowExcel, 8].EntireRow.Font.Bold = true;

rowExcel += 5;

obcol = 0;

obsum = 0;

}

}

else

{ MessageBox.Show("За этот период нет расходов по отделениям!"); }

}

private void SetColumnWidth(string range, double Width)//TODO: Ширина столбца

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Width };

Range.GetType().InvokeMember("ColumnWidth", BindingFlags.SetProperty, null, Range, args);

}

private void SetBorderStyle(string range, int Style)//TODO: Стиль границ ячейки

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Style };

object Borders = Range.GetType().InvokeMember("Borders", BindingFlags.GetProperty, null, Range, null);

Borders = Range.GetType().InvokeMember("LineStyle", BindingFlags.SetProperty, null, Borders, args);

}

private void SetWrapText(string range, bool Value)//TODO: Перенос по словам

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Value };

Range.GetType().InvokeMember("WrapText", BindingFlags.SetProperty, null, Range, args);

}

private void SetRowHeight(string range, double Height)//TODO: Высота строк

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty,

null, workSheet, new object[] { range });

object[] args = new object[] { Height };

Range.GetType().InvokeMember("RowHeight", BindingFlags.SetProperty, null, Range, args);

}

private void SetMerge(string range, int Alignment)//TODO: Выравнивание текста в ячейке

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Alignment };

Range.GetType().InvokeMember("MergeCells", BindingFlags.SetProperty, null, Range, new object[] { true });

Range.GetType().InvokeMember("HorizontalAlignment", BindingFlags.SetProperty, null, Range, args);

}

private void SetFont(string range, Microsoft.Office.Interop.Excel.Font font)

{

//Range.Font.Name

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty,

null, workSheet, new object[] { range });

object Font = Range.GetType().InvokeMember("Font", BindingFlags.GetProperty,

null, Range, null);

Range.GetType().InvokeMember("Name", BindingFlags.SetProperty, null,

Font, new object[] { font.Name });

Range.GetType().InvokeMember("Size", BindingFlags.SetProperty, null,

Font, new object[] { font.Size });

}

private void button1\_Click(object sender, EventArgs e)

{

ExportToExcel();

Close();

}

}

}

**Листинг 21 – Модуль класса Form21OtPoOdnomuOtdeleniu.cs**

{======================================================================}

*{класс Form21OtPoOdnomuOtdeleniu.cs}*

*{ гр. П-10 };*

*{ Разработчик: Васюхневич П.В.}*

*{ Модифицирован 8 мая 2018г }*

**{----------------------------------------------------------------------}**

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using Microsoft.Office.Interop.Excel;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Reflection;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Excel = Microsoft.Office.Interop.Excel;

namespace ULS

{

public partial class Form21OtPoOdnomuOtdeleniu : Form

{

Form5\_VibDvij f5;

Object Range;

Excel.\_Worksheet workSheet;

Form1 frm1;

public Form21OtPoOdnomuOtdeleniu(Form1 f, Form4\_Rashod f4)

{

f5 = new Form5\_VibDvij(f,f4, this);

InitializeComponent();

frm1 = f;

}

private void button2\_Click(object sender, EventArgs e)

{

Hide();

}

private void Form21OtPoOdnomuOtdeleniu\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "bDDataSet.Otdelenie". При необходимости она может быть перемещена или удалена.

this.otdelenieTableAdapter.Fill(this.bDDataSet.Otdelenie);

// TODO: данная строка кода позволяет загрузить данные в таблицу "bDDataSet.Rashod". При необходимости она может быть перемещена или удалена.

this.rashodTableAdapter.Fill(this.bDDataSet.Rashod);

}

private void button1\_Click(object sender, EventArgs e)

{

ExportToExcel();

Close();

}

private void ExportToExcel()

{

int count = 0;

for (int u = 0; u < dataGridView2.Rows.Count - 1; u++)

for (int j = 0; j < frm1.dataGridView1.Rows.Count - 1; j++)

for (int i = 0; i < dataGridView1.Rows.Count - 1; i++)

if (dataGridView2.Rows[u].Cells[1].Value.ToString() == textBox1.Text.ToString())

if (dataGridView2.Rows[u].Cells[1].Value.ToString() == dataGridView1.Rows[i].Cells[13].Value.ToString())

if (int.Parse(frm1.dataGridView1.Rows[j].Cells[11].Value.ToString()) == int.Parse(dataGridView1.Rows[i].Cells[1].Value.ToString()))

if (DateTime.Parse(dataGridView1.Rows[i].Cells[12].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker1.Value.ToShortDateString())) >= 0 && DateTime.Parse(dataGridView1.Rows[i].Cells[12].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker2.Value.ToShortDateString())) <= 0)

count++;

if (count != 0)

{

Excel.Application exApp = new Excel.Application();

exApp.Visible = true;

exApp.Workbooks.Add();

workSheet = (Worksheet)exApp.ActiveSheet;

SetMerge("A1:H1", 3);

SetMerge("A2:H2", 3);

workSheet.Cells[1, 1].EntireRow.Font.Bold = true;

workSheet.Cells[2, 1].EntireRow.Font.Bold = true;

workSheet.Cells[1, 1] = "Отчёт израсходованных лекарственных средств";

workSheet.Cells[2, 1] = "с " + dateTimePicker1.Value.ToShortDateString() + " по " + dateTimePicker2.Value.ToShortDateString() + " года для всех отделений.";

SetColumnWidth("A1", 3);

SetColumnWidth("B1", 32);

SetColumnWidth("C1", 19);

SetColumnWidth("D1", 19);

SetColumnWidth("E1", 22);

SetColumnWidth("F1", 15);

SetColumnWidth("G1", 12);

SetColumnWidth("H1", 16);

int rowExcel = 6;

int obcol = 0;

int obsum = 0;

int col;

for (int u = 0; u < dataGridView2.Rows.Count - 1; u++)

{

if (dataGridView2.Rows[u].Cells[1].Value.ToString() == textBox1.Text)

{

SetMerge("A" + (rowExcel - 2).ToString() + ":H" + (rowExcel - 2).ToString(), 3);

workSheet.Cells[(rowExcel - 2), 1].EntireRow.Font.Bold = true;

workSheet.Cells[(rowExcel - 2), 1] = dataGridView2.Rows[u].Cells[1].Value.ToString();

int xcv = rowExcel - 1;

workSheet.Cells[xcv, 1] = "№";

workSheet.Cells[xcv, 2] = "Наименование";

workSheet.Cells[xcv, 3] = "Фирма-поставщик";

workSheet.Cells[xcv, 4] = "Фирма-изготовитель";

workSheet.Cells[xcv, 5] = "Ед. измерения";

workSheet.Cells[xcv, 6] = "Колич.";

workSheet.Cells[xcv, 7] = "Цена за ед.";

workSheet.Cells[xcv, 8] = "Общая сумма";

workSheet.Cells[xcv, 1].EntireRow.Font.Bold = true;

workSheet.Cells[xcv, 2].EntireRow.Font.Bold = true;

workSheet.Cells[xcv, 3].EntireRow.Font.Bold = true;

workSheet.Cells[xcv, 4].EntireRow.Font.Bold = true;

workSheet.Cells[xcv, 5].EntireRow.Font.Bold = true;

workSheet.Cells[xcv, 6].EntireRow.Font.Bold = true;

workSheet.Cells[xcv, 7].EntireRow.Font.Bold = true;

workSheet.Cells[xcv, 8].EntireRow.Font.Bold = true;

SetColumnWidth("A" + (xcv).ToString(), 3);

SetColumnWidth("A" + (xcv).ToString(), 32);

SetColumnWidth("A" + (xcv).ToString(), 19);

SetColumnWidth("A" + (xcv).ToString(), 19);

SetColumnWidth("A" + (xcv).ToString(), 22);

SetColumnWidth("A" + (xcv).ToString(), 15);

SetColumnWidth("A" + (xcv).ToString(), 12);

SetColumnWidth("A" + (xcv).ToString(), 16);

SetBorderStyle("A" + (xcv).ToString() + ":H" + (xcv).ToString(), 1);

for (int j = 0; j < frm1.dataGridView1.Rows.Count - 1; j++)

{

col = 0;

for (int i = 0; i < dataGridView1.Rows.Count - 1; i++)

{

if (dataGridView2.Rows[u].Cells[1].Value.ToString() == dataGridView1.Rows[i].Cells[13].Value.ToString())

{

if (int.Parse(frm1.dataGridView1.Rows[j].Cells[11].Value.ToString()) == int.Parse(dataGridView1.Rows[i].Cells[1].Value.ToString()))

{

if (DateTime.Parse(dataGridView1.Rows[i].Cells[12].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker1.Value.ToShortDateString())) >= 0 && DateTime.Parse(dataGridView1.Rows[i].Cells[12].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker2.Value.ToShortDateString())) <= 0)

{

col = col + int.Parse(dataGridView1.Rows[i].Cells[9].Value.ToString());

}

}

}

}

if (col > 0)

{

SetBorderStyle("A" + (rowExcel).ToString() + ":H" + (rowExcel).ToString(), 1);

workSheet.Cells[rowExcel, "A"] = frm1.dataGridView1.Rows[j].Cells[11].Value;

workSheet.Cells[rowExcel, "B"] = frm1.dataGridView1.Rows[j].Cells[0].Value;

workSheet.Cells[rowExcel, "C"] = frm1.dataGridView1.Rows[j].Cells[2].Value;

workSheet.Cells[rowExcel, "D"] = frm1.dataGridView1.Rows[j].Cells[4].Value;

workSheet.Cells[rowExcel, "E"] = frm1.dataGridView1.Rows[j].Cells[6].Value;

workSheet.Cells[rowExcel, "F"] = col;

workSheet.Cells[rowExcel, "G"] = frm1.dataGridView1.Rows[j].Cells[8].Value;

workSheet.Cells[rowExcel, "H"] = (col \* int.Parse(frm1.dataGridView1.Rows[j].Cells[8].Value.ToString()));

++rowExcel;

obcol = obcol + col;

obsum = obsum + (col \* (int.Parse(frm1.dataGridView1.Rows[j].Cells[8].Value.ToString())));

}

}

SetBorderStyle("E" + (rowExcel).ToString() + ":H" + (rowExcel).ToString(), 1);

workSheet.Cells[rowExcel, 5] = "ИТОГО:";

workSheet.Cells[rowExcel, 6] = obcol.ToString();

workSheet.Cells[rowExcel, 8] = obsum.ToString();

workSheet.Cells[rowExcel, 6].EntireRow.Font.Bold = true;

workSheet.Cells[rowExcel, 8].EntireRow.Font.Bold = true;

rowExcel += 5;

obcol = 0;

obsum = 0;

}

}

}

else

{ MessageBox.Show("За этот период нет расходов по отделениям!"); }

}

private void SetColumnWidth(string range, double Width)//TODO: Ширина столбца

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Width };

Range.GetType().InvokeMember("ColumnWidth", BindingFlags.SetProperty, null, Range, args);

}

private void SetBorderStyle(string range, int Style)//TODO: Стиль границ ячейки

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Style };

object Borders = Range.GetType().InvokeMember("Borders", BindingFlags.GetProperty, null, Range, null);

Borders = Range.GetType().InvokeMember("LineStyle", BindingFlags.SetProperty, null, Borders, args);

}

private void SetWrapText(string range, bool Value)//TODO: Перенос по словам

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Value };

Range.GetType().InvokeMember("WrapText", BindingFlags.SetProperty, null, Range, args);

}

private void SetRowHeight(string range, double Height)//TODO: Высота строк

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty,

null, workSheet, new object[] { range });

object[] args = new object[] { Height };

Range.GetType().InvokeMember("RowHeight", BindingFlags.SetProperty, null, Range, args);

}

private void SetMerge(string range, int Alignment)//TODO: Выравнивание текста в ячейке

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Alignment };

Range.GetType().InvokeMember("MergeCells", BindingFlags.SetProperty, null, Range, new object[] { true });

Range.GetType().InvokeMember("HorizontalAlignment", BindingFlags.SetProperty, null, Range, args);

}

private void SetFont(string range, Microsoft.Office.Interop.Excel.Font font)

{

//Range.Font.Name

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty,

null, workSheet, new object[] { range });

object Font = Range.GetType().InvokeMember("Font", BindingFlags.GetProperty,

null, Range, null);

Range.GetType().InvokeMember("Name", BindingFlags.SetProperty, null,

Font, new object[] { font.Name });

Range.GetType().InvokeMember("Size", BindingFlags.SetProperty, null,

Font, new object[] { font.Size });

}

private void button3\_Click(object sender, EventArgs e)

{

f5.f21 = true;

f5.f44 = false;

f5.ShowDialog();

}

public void func(string s)

{

textBox1.Text = s;

}

public void en()

{

if (textBox1.Text != "" )

button1.Enabled = true;

else

button1.Enabled = false;

}

}

}

**Листинг 22 – Модуль класса Form22\_OtODvij.cs**

{======================================================================}

*{класс Form22\_OtODvij.cs}*

*{ гр. П-10 };*

*{ Разработчик: Васюхневич П.В.}*

*{ Модифицирован 8 мая 2018г }*

**{----------------------------------------------------------------------}**

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using Microsoft.Office.Interop.Excel;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Reflection;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Excel = Microsoft.Office.Interop.Excel;

namespace ULS

{

public partial class Form22\_OtODvij : Form

{

Object Range;

Excel.\_Worksheet workSheet;

Form1 frm1;

public Form22\_OtODvij(Form1 f)

{

InitializeComponent();

frm1 = f;

}

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

private void Form22\_OtODvij\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "bDDataSet.Rashod". При необходимости она может быть перемещена или удалена.

this.rashodTableAdapter.Fill(this.bDDataSet.Rashod);

}

private void button1\_Click(object sender, EventArgs e)

{

ExportToExcel();

Close();

}

private void ExportToExcel()

{

int count = 0;

for (int j = 0; j < frm1.dataGridView1.Rows.Count - 1; j++)

count++;

if (count != 0)

{

Excel.Application exApp = new Excel.Application();

exApp.Visible = true;

exApp.Workbooks.Add();

workSheet = (Worksheet)exApp.ActiveSheet;

SetMerge("A1:K1", 3);

SetMerge("A2:K2", 3);

SetMerge("A4:A5", 3);

SetMerge("B4:B5", 3);

SetMerge("C4:C5", 3);

SetMerge("D4:E4", 3);

SetMerge("F4:G4", 3);

SetMerge("H4:I4", 3);

SetMerge("J4:K4", 3);

workSheet.Cells[1, 1].EntireRow.Font.Bold = true;

workSheet.Cells[2, 1].EntireRow.Font.Bold = true;

workSheet.Cells[4, 4].EntireRow.Font.Bold = true;

workSheet.Cells[4, 6].EntireRow.Font.Bold = true;

workSheet.Cells[4, 8].EntireRow.Font.Bold = true;

workSheet.Cells[4, 10].EntireRow.Font.Bold = true;

workSheet.Cells[5, 1].EntireRow.Font.Bold = true;

workSheet.Cells[5, 2].EntireRow.Font.Bold = true;

workSheet.Cells[5, 3].EntireRow.Font.Bold = true;

workSheet.Cells[5, 4].EntireRow.Font.Bold = true;

workSheet.Cells[5, 5].EntireRow.Font.Bold = true;

workSheet.Cells[5, 6].EntireRow.Font.Bold = true;

workSheet.Cells[5, 7].EntireRow.Font.Bold = true;

workSheet.Cells[5, 8].EntireRow.Font.Bold = true;

workSheet.Cells[5, 9].EntireRow.Font.Bold = true;

workSheet.Cells[5, 10].EntireRow.Font.Bold = true;

workSheet.Cells[5, 11].EntireRow.Font.Bold = true;

SetBorderStyle("A4:K4", 1);

SetBorderStyle("A5:K5", 1);

workSheet.Cells[1, 1] = "Отчёт о движении медикаментов";

workSheet.Cells[2, 1] = "и средств медицинского назначения с " + dateTimePicker1.Value.ToShortDateString() + " по " + dateTimePicker2.Value.ToShortDateString() + " года.";

workSheet.Cells[4, 1] = "№";

workSheet.Cells[4, 2] = "Наименование";

workSheet.Cells[4, 3] = "Ед. измерения";

workSheet.Cells[4, 4] = "Остаток на начало периода";

workSheet.Cells[4, 6] = "Приход";

workSheet.Cells[4, 8] = "Израсходовано";

workSheet.Cells[4, 10] = "Остаток на конец периода";

workSheet.Cells[5, 4] = "Кол-во";

workSheet.Cells[5, 5] = "Сумма";

workSheet.Cells[5, 6] = "Кол-во";

workSheet.Cells[5, 7] = "Сумма";

workSheet.Cells[5, 8] = "Кол-во";

workSheet.Cells[5, 9] = "Сумма";

workSheet.Cells[5, 10] = "Кол-во";

workSheet.Cells[5, 11] = "Сумма";

SetColumnWidth("A1", 3);

SetColumnWidth("B1", 32);

SetColumnWidth("C1", 19);

SetColumnWidth("D1", 19);

SetColumnWidth("E1", 22);

SetColumnWidth("F1", 15);

SetColumnWidth("G1", 12);

SetColumnWidth("H1", 16);

SetColumnWidth("I1", 9);

SetColumnWidth("J1", 11);

SetColumnWidth("K1", 13);

int rowExcel = 6;

int colstart=0;

int summstart=0;

int colend=0;

int summend=0;

int sumprihod = 0;

int colprihod=0;

int colrashod = 0;

int sumrashod = 0;

for (int j = 0; j < frm1.dataGridView1.Rows.Count - 1; j++)

{

colstart=0;

summstart=0;

colend=0;

summend=0;

sumprihod = 0;

colprihod=0;

colrashod = 0;

sumrashod = 0;

if (DateTime.Parse(frm1.dataGridView1.Rows[j].Cells[1].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker1.Value.ToShortDateString())) == 0)

{

colstart = 0;

summstart = 0;

}

if (DateTime.Parse(frm1.dataGridView1.Rows[j].Cells[1].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker1.Value.ToShortDateString())) > 0)

{

colstart = 0;

summstart = 0;

}

if (DateTime.Parse(frm1.dataGridView1.Rows[j].Cells[1].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker1.Value.ToShortDateString())) < 0)

{

colstart = int.Parse(frm1.dataGridView1.Rows[j].Cells[12].Value.ToString());

summstart = int.Parse(frm1.dataGridView1.Rows[j].Cells[8].Value.ToString()) \* int.Parse(frm1.dataGridView1.Rows[j].Cells[12].Value.ToString());

for (int i = 0; i < dataGridView1.Rows.Count - 1; i++)

{

if (int.Parse(frm1.dataGridView1.Rows[j].Cells[11].Value.ToString()) == int.Parse(dataGridView1.Rows[i].Cells[1].Value.ToString()))

{

if (DateTime.Parse(dataGridView1.Rows[i].Cells[12].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker1.Value.ToShortDateString())) < 0)

{

summstart -= (int.Parse(dataGridView1.Rows[i].Cells[9].Value.ToString()) \* int.Parse(dataGridView1.Rows[i].Cells[10].Value.ToString()));

colstart -= int.Parse(dataGridView1.Rows[i].Cells[9].Value.ToString());

}

}

}

}

if (DateTime.Parse(frm1.dataGridView1.Rows[j].Cells[1].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker1.Value.ToShortDateString())) >= 0 && DateTime.Parse(frm1.dataGridView1.Rows[j].Cells[1].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker2.Value.ToShortDateString())) <= 0)

{

colprihod = int.Parse(frm1.dataGridView1.Rows[j].Cells[12].Value.ToString());

sumprihod = int.Parse(frm1.dataGridView1.Rows[j].Cells[8].Value.ToString()) \* int.Parse(frm1.dataGridView1.Rows[j].Cells[12].Value.ToString());

}

colend = int.Parse(frm1.dataGridView1.Rows[j].Cells[7].Value.ToString());

summend = int.Parse(frm1.dataGridView1.Rows[j].Cells[9].Value.ToString());

if (int.Parse(frm1.dataGridView1.Rows[j].Cells[12].Value.ToString()) > int.Parse(frm1.dataGridView1.Rows[j].Cells[7].Value.ToString()))

{

for (int i = 0; i < dataGridView1.Rows.Count-1; i++)

{

if (int.Parse(frm1.dataGridView1.Rows[j].Cells[11].Value.ToString()) == int.Parse(dataGridView1.Rows[i].Cells[1].Value.ToString()))

{

if (DateTime.Parse(dataGridView1.Rows[i].Cells[12].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker1.Value.ToShortDateString())) >= 0 && DateTime.Parse(dataGridView1.Rows[i].Cells[12].Value.ToString()).CompareTo(DateTime.Parse(dateTimePicker2.Value.ToShortDateString())) <= 0)

{

colrashod += int.Parse(dataGridView1.Rows[i].Cells[9].Value.ToString());

sumrashod += (int.Parse(dataGridView1.Rows[i].Cells[9].Value.ToString()) \* int.Parse(dataGridView1.Rows[i].Cells[10].Value.ToString()));

}

}

}

}

SetBorderStyle("A" + (rowExcel).ToString() + ":K" + (rowExcel).ToString(), 1);

workSheet.Cells[rowExcel, "A"] = frm1.dataGridView1.Rows[j].Cells[11].Value;

workSheet.Cells[rowExcel, "B"] = frm1.dataGridView1.Rows[j].Cells[0].Value;

workSheet.Cells[rowExcel, "C"] = frm1.dataGridView1.Rows[j].Cells[6].Value;

workSheet.Cells[rowExcel, "D"] = colstart;

workSheet.Cells[rowExcel, "E"] = summstart;

workSheet.Cells[rowExcel, "F"] = colprihod;

workSheet.Cells[rowExcel, "G"] = sumprihod;

workSheet.Cells[rowExcel, "H"] = colrashod;

workSheet.Cells[rowExcel, "I"] = sumrashod;

workSheet.Cells[rowExcel, "J"] = colend;

workSheet.Cells[rowExcel, "K"] = summend;

++rowExcel;

}

}

else

{ MessageBox.Show("За этот период нет записей!"); }

}

private void SetColumnWidth(string range, double Width)//TODO: Ширина столбца

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Width };

Range.GetType().InvokeMember("ColumnWidth", BindingFlags.SetProperty, null, Range, args);

}

private void SetBorderStyle(string range, int Style)//TODO: Стиль границ ячейки

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Style };

object Borders = Range.GetType().InvokeMember("Borders", BindingFlags.GetProperty, null, Range, null);

Borders = Range.GetType().InvokeMember("LineStyle", BindingFlags.SetProperty, null, Borders, args);

}

private void SetWrapText(string range, bool Value)//TODO: Перенос по словам

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Value };

Range.GetType().InvokeMember("WrapText", BindingFlags.SetProperty, null, Range, args);

}

private void SetRowHeight(string range, double Height)//TODO: Высота строк

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty,

null, workSheet, new object[] { range });

object[] args = new object[] { Height };

Range.GetType().InvokeMember("RowHeight", BindingFlags.SetProperty, null, Range, args);

}

private void SetMerge(string range, int Alignment)//TODO: Выравнивание текста в ячейке

{

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty, null, workSheet, new object[] { range });

object[] args = new object[] { Alignment };

Range.GetType().InvokeMember("MergeCells", BindingFlags.SetProperty, null, Range, new object[] { true });

Range.GetType().InvokeMember("HorizontalAlignment", BindingFlags.SetProperty, null, Range, args);

}

private void SetFont(string range, Microsoft.Office.Interop.Excel.Font font)

{

//Range.Font.Name

Range = workSheet.GetType().InvokeMember("Range", BindingFlags.GetProperty,

null, workSheet, new object[] { range });

object Font = Range.GetType().InvokeMember("Font", BindingFlags.GetProperty,

null, Range, null);

Range.GetType().InvokeMember("Name", BindingFlags.SetProperty, null,

Font, new object[] { font.Name });

Range.GetType().InvokeMember("Size", BindingFlags.SetProperty, null,

Font, new object[] { font.Size });

}

}

}

**Листинг 23 – Модуль класса Form23\_OProgramme.cs**

{======================================================================}

*{класс Form23\_OProgramme.cs}*

*{ гр. П-10 };*

*{ Разработчик: Васюхневич П.В.}*

*{ Модифицирован 8 мая 2018г }*

**{----------------------------------------------------------------------}**

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ULS

{

public partial class Form23\_OProgramme : Form

{

public Form23\_OProgramme()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

private void Form23\_OProgramme\_Load(object sender, EventArgs e)

{

}

}

}

**Листинг 24 – Модуль класса Form24\_VibIzgotovit.cs**

{======================================================================}

*{класс Form24\_VibIzgotovit.cs}*

*{ гр. П-10 };*

*{ Разработчик: Васюхневич П.В.}*

*{ Модифицирован 8 мая 2018г }*

**{----------------------------------------------------------------------}**

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ULS

{

public partial class Form24\_VibIzgotovit : Form

{

Form2\_Prihod frm2;

Form6\_Edit frm6;

public bool fr2 = false;

public bool fr6 = false;

public Form24\_VibIzgotovit(Form2\_Prihod f2, Form6\_Edit f6)

{

InitializeComponent();

frm2 = f2;

frm6 = f6;

}

private void Form24\_VibIzgotovit\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "bDDataSet.Izgotovitel". При необходимости она может быть перемещена или удалена.

this.izgotovitelTableAdapter.Fill(this.bDDataSet.Izgotovitel);

}

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

private void button1\_Click(object sender, EventArgs e)

{

if (fr2==true)

frm2.func2(dataGridView1.CurrentRow.Cells[0].Value.ToString());

if (fr6==true)

frm6.func2(dataGridView1.CurrentRow.Cells[0].Value.ToString());

Close();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

try

{

izgotovitelBindingSource.Position = izgotovitelBindingSource.Find("Name", textBox1.Text);

}

catch (Exception)

{ }

}

}

}

**Листинг 25 – Модуль класса Form25\_VibFirmPost.cs**

{======================================================================}

*{класс Form25\_VibFirmPost.cs}*

*{ гр. П-10 };*

*{ Разработчик: Васюхневич П.В.}*

*{ Модифицирован 8 мая 2018г }*

**{----------------------------------------------------------------------}**

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ULS

{

public partial class Form25\_VibFirmPost : Form

{

Form2\_Prihod frm2;

Form6\_Edit frm6;

public bool fr2 = false;

public bool fr6 = false;

public Form25\_VibFirmPost(Form2\_Prihod f2, Form6\_Edit f6)

{

InitializeComponent();

frm2 = f2;

frm6 = f6;

}

private void Form25\_VibFirmPost\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "bDDataSet.FirmaPostav". При необходимости она может быть перемещена или удалена.

this.firmaPostavTableAdapter.Fill(this.bDDataSet.FirmaPostav);

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

try

{

firmaPostavBindingSource.Position = firmaPostavBindingSource.Find("Name", textBox1.Text);

}

catch (Exception)

{ }

}

private void button1\_Click(object sender, EventArgs e)

{

if (fr2 == true)

frm2.func3(dataGridView1.CurrentRow.Cells[0].Value.ToString());

if (fr6 == true)

frm6.func3(dataGridView1.CurrentRow.Cells[0].Value.ToString());

Close();

}

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

}

}

# **ПРИЛОЖЕНИЕ Б** СОДЕРЖАНИЕ ЭЛЕКТРОННОГО НОСИТЕЛЯ

На прилагаемом электронном носителе информации расположены следующие директории и файлы:

1. Директория DOC - файл *«Пояснительная записка.doc»* представляет собой текстовый документ спояснительной запиской.
2. Директория «PRG» содержит исходный код проекта Setup.exe и файлы базы данных в директории BD.
3. Директория ***«****ЕХЕ****»*** содержит финальную версию исполняемых файлов проекта *Setup.exe.*