Одеський національний політехнічний університет

Кафедра «Комп'ютеризовані системи управління»

**КУРСОВА РОБОТА**

з дисципліни «Програмування та теорія алгоритмів»

на тему: «Створення інформаційної бази»

Варіант 15

Студента 1 курсу, групи АТ-192

спеціальності «Автоматизація та комп'ютерно-інтегровані технології»

**Петрова М.В.**

Керівник: доц. Сперанський В.О.

Національна шкала:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оцінка: ECTS\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

м. Одеса – 2020 рік

СОДЕРЖАНИЕ

[ПОСТАНОВКА ЗАДАЧИ 3](#_Toc10276841)

[РАЗДЕЛ 1 ПРОЕКТИРОВАНИЕ И ИХ РЕАЛИЗАЦИЯ 5](#_Toc10276842)

[Календарное планирование проекта 5](#_Toc10276843)

[Описание проектирование программы 6](#_Toc10276844)

[Функции программы 7](#_Toc10276845)

[Библиотеки 8](#_Toc10276846)

[РАЗДЕЛ 2 СПРАВОЧНАЯ СИСТЕМА 9](#_Toc10276847)

[Краткое описание продукта 9](#_Toc10276848)

[Как добавить студента 9](#_Toc10276849)

[Импорт данных 10](#_Toc10276850)

[Експорт данных 11](#_Toc10276851)

[Поиск и редактирование 12](#_Toc10276852)

[Выдача отчетов 13](#_Toc10276853)

[ВЫВОДЫ 14](#_Toc10276854)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ: 15](#_Toc10276855)

# ПОСТАНОВКА ЗАДАЧИ

Вариант 15

Создать базу данных – картотека в библиотеке

Каждая запись содержит следующие сведения:

* название книги,
* автор,
* год издания,
* издательство,
* цена,
* состояние,
* дата выдачи читателю на руки.

Предусмотреть:

1. изменить состояние книг, изданных больше N лет назад;
2. вывести названия всех книг, цена на которые > 100 гривен;
3. вывести названия всех книг данного автора;
4. вывести информацию о книгах, которые находятся на руках дольше 3 месяцев.
5. Спроектировать структуру информационной базы для своего варианта задания (Приложение А).
6. Создать основную таблицу информационной базы в виде динамического массива структур (использование списка из элементов типа структура оценивается выше).
7. Предусмотреть функции контроля корректности всех вводимых данных (например, неотрицательный возраст, дата продажи не ранее, чем 100 лет назад и т.п.).
8. Заполнить основную таблицу не менее чем 20 записями.
9. Обеспечить сохранение и загрузку данных во внешний файл/из файла.
10. Реализовать меню пользователя, соответствующие функции которого обеспечивают:
11. создание информационной базы той предметной области, которая определена постановкой задачи;
12. добавление новых записей в базу данных (с контролем корректности вводимых значений);
13. поиск записей по *ключевому* полю;
14. редактирование записей базы данных;
15. удаление ненужных записей;
16. выдачу отчётов по запросу.
17. Выполнить оригинальную обработку, указанную в постановке конкретной задачи для вашего варианта (Приложение А).
18. Организовать ввод пароля при выполнении операций изменения данных.
19. Добавить справочную систему пользователя по созданному приложению.

# РАЗДЕЛ 1 ПРОЕКТИРОВАНИЕ И ИХ РЕАЛИЗАЦИЯ

## Календарное планирование проекта

|  |  |  |  |
| --- | --- | --- | --- |
| № | Наименование | Содержание | Дата окончания |
|  | Начало | Изучение постановочного задания, предметной области |  |
|  | Теоретическая подготовка | Изучение литературы по дисциплине |  |
|  | Проект | Создание репазитория и проекта в программе Visual Studio |  |
|  | Дизайн | Создание дизайна для приложения |  |
|  | Классы | Написания новых классов с полями поставленными в задании, а также свойств для работы с ними |  |
|  | Связка | Настройка взаимосвязи класса с графическими инструментами |  |
|  | DataGridView | Добавление инструмента DataGridView для вывода информации и дальнейшей работы |  |
|  | Выдача отчётов | Работа над индивидуальным задание |  |
|  | Експорт и импорт данных | Написание кода для сохранения и загрузки данных во внешний файл/из файла |  |
|  | Меню | Проектирование и доработка меню для пользователя |  |
|  | Корректность ввода | Написания функции контроля корректности всех вводимых данных |  |
|  | Пароль | Добавление ввода пароля при выполнении операций изменения данных |  |
|  | Справка | Создание справочной системы, для пользователей |  |
|  | Оптимизация | Исправление ошибок и очищение не нужного кода |  |
|  | Документация | Подготовка пояснительной записки о программе |  |

## Описание проектирование программы

Основа программы состоит из двух классов: класс (Book) который состоит из восьми свойств, а также конструктора по умолчанию.

Пример кода данного класа:

public class Book

{

public int BookID { get; set; } // Порядковый номер книги в Картотекеу(уникальный идентификатор)

public string BookTitle { get; set; } //Название книги(строка)

public string BookAuthor { get; set; } //Автор книги

public DateTime? BookPublicationDate { get; set; } // Дата публикации

public string BookPublishing { get; set; } // Издательство

public decimal BookPrice { get; set; } //цена за ед.

public bool IsExsist { get; set; } // Состояние

public DateTime BookDiliveryDate { get; set; } //дата выдачи читателю на руки.

}

Второй класс (LibraryFileCabinet), который также включает в себе свойство-список книг а также конструктор.

public class LibraryFileCabinet

{

public LibraryFileCabinet()

{

Books = new List<Book>();

}

public List<Book> Books { get; set; }

}

## Функции программы

При написании программы, для правильности и хорошего чтения кода были созданы и задействованы функции-события, их назначение и примеры предоставлены ниже:

Запись всех данных в таблицу:

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

Функция измения состояние книг, изданных больше N лет назад:

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

Вывод данных на экран о студентах в новой форме:

void Fill\_Form\_Show()

Вычисление средней оценки у студентов:

void Average\_mark()

Подгрузка данных в ComboBox для возможности выбора преподавателя:

void Export\_List\_Teachers()

Проверка на корректность ввода текстовой информации:

void Check\_input\_text(object sender, KeyPressEventArgs e)

Проверка на корректность ввода числовой информации:

void Check\_input\_digit(object sender, KeyPressEventArgs e)

Функция генерации случайного пароля:

void Generate\_Password()

Поиск студента в базе по номеру зачетки:

void Search\_Student()

Проверка правильности ввода пароля для допуска к редактированию:

void Check\_password()

Удаление студента из базы:

void Remove\_student()

Функция выбора предмета для редактирования:

void Choose\_subject()

Сохранение изменений основных данных о студенте:

void Edit\_main\_data()

Сохранение изменений успеваемости студента:

void Edit\_progress()

Также в программе были созданы функции, предусмотренные постановкой задачи:

1. изменить состояние книг, изданных больше N лет назад;
2. вывести названия всех книг, цена на которые > 100 гривен;
3. вывести названия всех книг данного автора;
4. вывести информацию о книгах, которые находятся на руках дольше 3 месяцев.

Запись и Чтение информации в/из файл(а) осуществлялось с помощью класов

StreamReader и StreamWriter а также Функций:

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

{

using (TextReader text = new StreamReader(File.OpenRead($@"C:\\DataBaseKyrsach\\info.txt"/\*$@"C:\Users\{SystemInformation.UserName}\Desktop\info.txt"\*/), Encoding.Default))

{

try

{

var str = text.ReadLine();

if (str.Length != 0)

{

foreach (var item in str.Replace('\t', '\n'))

{

cachedRichTextBox.Text += item.ToString();

}

MessageBox.Show("Information has been added");

}

else

{

MessageBox.Show("Empty file");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

text.Close();

}

}

}

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

{

using (TextWriter writer = new StreamWriter(File.Create(/\*$@"C:\Users\{SystemInformation.UserName}\Desktop\*/$@"C:\\DataBaseKyrsach\\info.txt"), Encoding.Default))

{

try

{

if (registrationsBookDataGrid.Rows.Count - 1 != 0)

{

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

{

for (int j = 0; j <gistrationsBookDataGrid.Columns.Count; j++)

{

if (j == 8)

writer.Write(registrationsBookDataGrid.Rows[i].Cells[j].Value.ToString() + '\n');

else

writer.Write(registrationsBookDataGrid.Rows[i].Cells[j].Value.ToString() + '\t');

}

}

MessageBox.Show("Info cashed");

}

else

{

MessageBox.Show("Need more info");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

writer.Close();

}

}

}

Рис. 1

## Библиотеки

Для написания данной программы необходимо подключить некоторые стандартные библиотеки языков С#. Их перечень представлен в коде:

using Kursach\_15Variant.Utility;

using Model.FileCabinet;

using System;

using System.Globalization;

using System.IO;

using System.Linq;

using System.Security.Cryptography.X509Certificates;

using System.Text;

using System.Windows.Forms;

# РАЗДЕЛ 2 СПРАВОЧНАЯ СИСТЕМА

## Краткое описание продукта

Приложение представляет собой ПО, в котором разработана база данных «Картотека в библиотеке».

Функциональность программы включает:

* управление данными картотеки: добавление, удаление и редактирование;
* измениие состояния книг, изданных больше N лет назад;
* вывод названия всех книг, цена на которые > 100 гривен;
* вывод названия всех книг данного автора;
* сохранение данных и открытие в файл/из файла

## Как добавить данные в БД

Для того, чтобы добавить новую запись в картотеку и всю информацию относящююся к книге:

1. Вам нужно перейти в раздел "Book Registration"

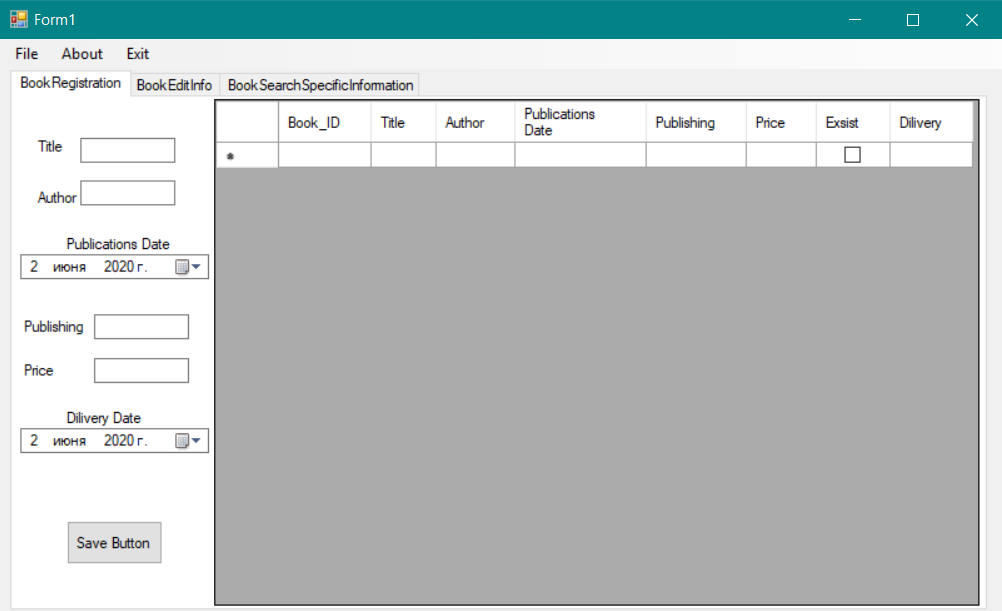


Рис. 1

1. Заполнить соответствующие поля нужной вам информацией (Пример можете посмотреть предварительно наведя на поле ввода)
2. Нажать на кнопку "Save Button"

## Импорт данных

В программе пользеватель имеет возможность импортировать данные двумя способами:

1. Экспорт из файла с расшерением "txt".

1. 1. Для этого в панели инструментов нажмите на "File" и выбирите кнопку "Read from".

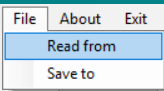


Рис. 2

1. 2. Далее в открывшемся окне, выберете файл и нажмите "Открыть".

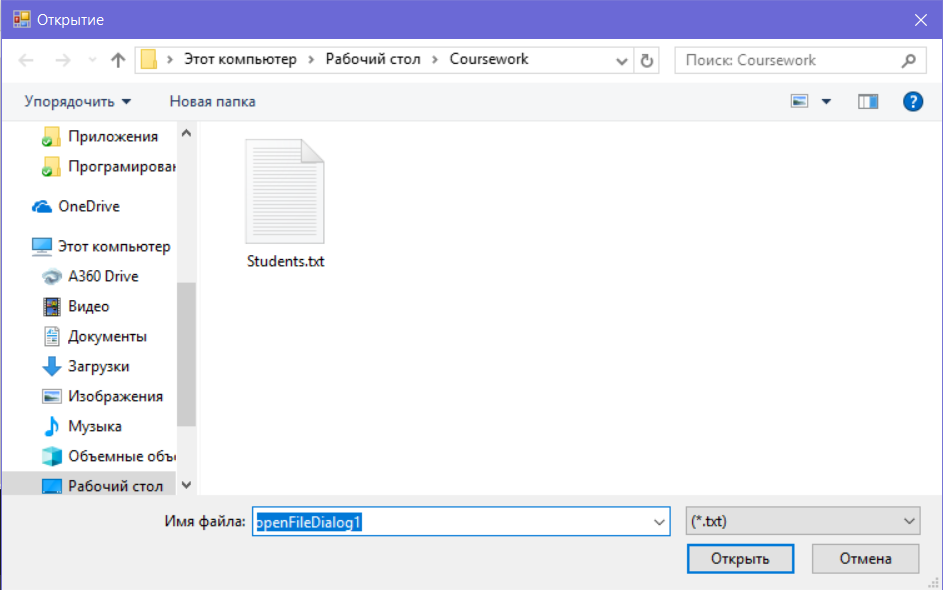


Рис. 3

## 

## Експорт данных

Как и в случае импорта данных, пользователь имеет возможность выбрать способ сохранения:

1. Сохранение с расширением "txt"

1. 1. Для этого в панели инструментов нажмите на "File" и выберите кнопку "Save to".

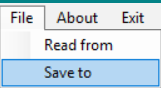


Рис. 4

1. 2. Далее в открывшемся окне, выберете путь для хранения файла и имя , нажмите "Сохранить".

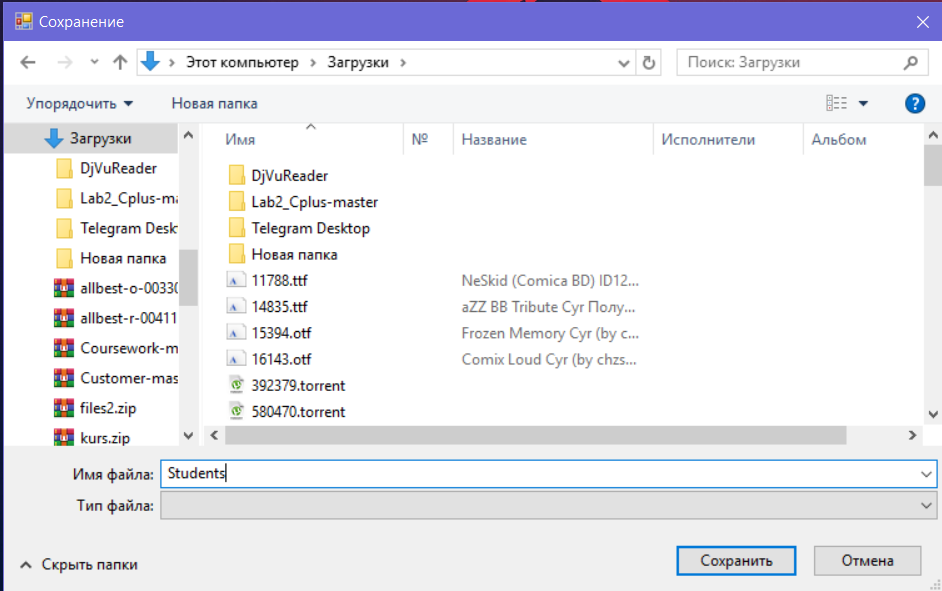


Рис. 5

## 

## Поиск и редактирование

Для измения состояния книг, изданных больше N лет назад нужно:

1. Перейти в раздел "Book Edit Info"

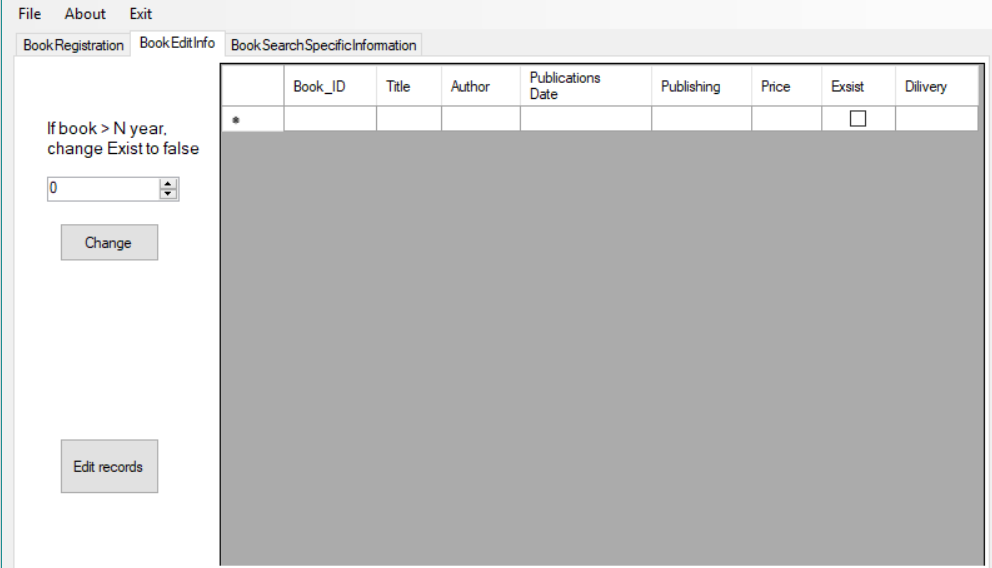


Рис. 6

2. Ввести в соответствующее год.

3. Нажать кнопку Change.

Если книги удовлетворяющие условию существуют в базе, информация о них выведется на экран, если не найдено выведется соответствующее окно, которое оповестит вас, что таких данных нет в базе.

После того как будет найдено книги, у вас появится возможность редактирования.

## 

## 

## Выдача отчетов

Для того чтобы посмотреть индивидуальное задание:

1. Перейдите в раздел "BookSearchSpecificInformation"

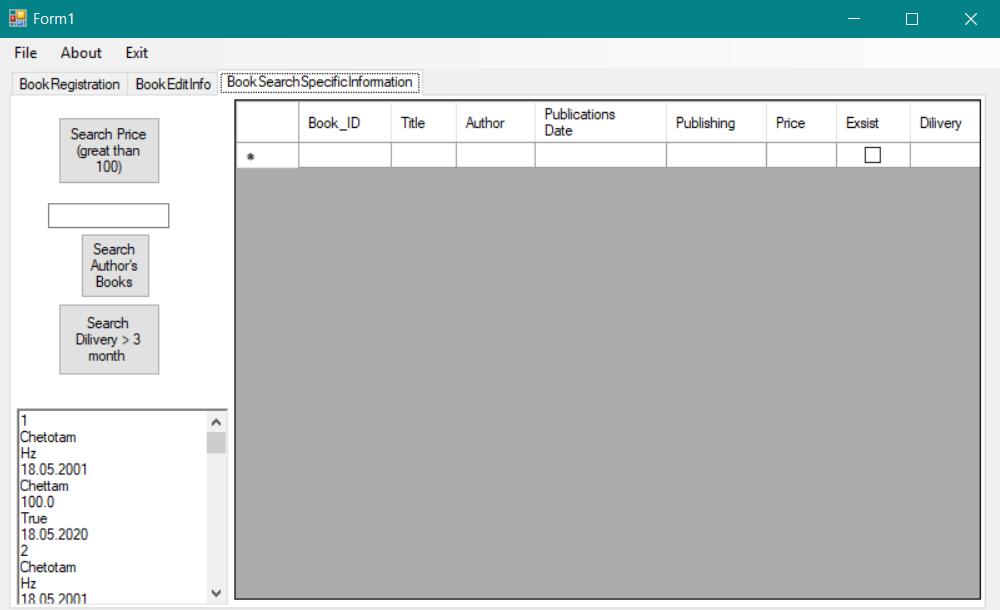


Рис. 7

2. И в этом разделе, выберете интересующий вас вариант нажав на него.

В зависимости от выбранной вами кнопкой появится такая информация:

"Search Price(great than 100)" - вывести названия всех книг, цена на которые > 100 гривен

"Search Author’s Books" - вывести названия всех книг данного автора "Search Delivery > 3 month" - вывести информацию о книгах, которые находятся на руках дольше 3 месяцев.

# ВЫВОДЫ

На основании знаний, приобретенных на курсе «Программирование и теория алгоритмов», я создал приложение которое автоматизирует работу с базой данных. Во время разработки программы я ознакомился с множеством новых инструментов создания приложений и повысил свой уровень профессиональных навыков.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ:

* <https://metanit.com/sharp/general.php>
* <https://ru.stackoverflow.com/>
* <http://www.cyberforum.ru/net-framework/>