Одеський національний політехнічний університет

Кафедра «Комп'ютеризовані системи управління»

**КУРСОВА РОБОТА**

з дисципліни «Програмування та теорія алгоритмів»

на тему: «Створення інформаційної бази»

Варіант 5

Студента 1 курсу, групи АТ-192

спеціальності «Автоматизація та комп'ютерно-інтегровані технології»

**Грибков М.**

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 5

Сформировать базу данных, в которую заносят результаты соревнований зимней Олимпиады. Каждая запись содержит следующие сведения:

* *регистрационный номер спортсмена;*
* ФИО спортсмена;
* название страны (Украина, США и т.д.);
* вид соревнований (бобслей, коньки и др.);
* занятое место.

*Предусмотреть*:

а) выдачу списка призёров по каждому виду спорта;

б) выдачу справок о количестве золотых, серебряных и бронзовых медалей, завоёванных каждой страной - участницей Олимпиады;

в) выдачу сведений о всех призёрах страны с заданным названием;

г) удаление из базы записей о спортсменах с указанной фамилией (по требованию допинг-контроля);

1. Спроектировать структуру информационной базы для своего варианта задания (Приложение А).
2. Создать основную таблицу информационной базы в виде динамического массива структур (использование списка из элементов типа структура оценивается выше).
3. Предусмотреть функции контроля корректности всех вводимых данных (например, неотрицательный возраст, дата продажи не ранее, чем 100 лет назад и т.п.).
4. Заполнить основную таблицу не менее чем 20 записями.
5. Обеспечить сохранение и загрузку данных во внешний файл/из файла.
6. Реализовать меню пользователя, соответствующие функции которого обеспечивают:
7. создание информационной базы той предметной области, которая определена постановкой задачи;
8. добавление новых записей в базу данных (с контролем корректности вводимых значений);
9. поиск записей по *ключевому* полю;
10. редактирование записей базы данных;
11. удаление ненужных записей;
12. выдачу отчётов по запросу.
13. Выполнить оригинальную обработку, указанную в постановке конкретной задачи для вашего варианта (Приложение А).
14. Организовать ввод пароля при выполнении операций изменения данных.
15. 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 (Sporstman) который состоит из 7 свойств, а также 3 переопределенных методов конструктора по умолчанию.

Пример кода данного класа:

public sealed class Sporstman

{

public int Registration\_ID { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public string Patronymic { get; set; }

public string Country { get; set; }

public string TypeOfCompetition{ get; set; }

public int Place { get; set; }

public override string ToString()

{

return $"{FirstName} {LastName} from {Country} in {TypeOfCompetition} took {Place} place";

}

public override bool Equals(object obj)

{

if (((Sporstman)obj).FirstName == this.FirstName && ((Sporstman)obj).LastName == this.LastName)

return true;

if (obj.GetType() == this.GetType())

return true;

return false;

}

public override int GetHashCode()

{

return this.Registration\_ID + 1000;

}

}Course = course;

}

Второй класс (WinterOlimpic), который также включает в себе 3 поля, свойства к каждому полю и конструкторы.

Пример кода:

public class WinterOlimpic

{

private List<Sporstman> sporstmens;

public WinterOlimpic()

{

sporstmens = new List<Sporstman>();

}

public List<Sporstman> Sporstmens { get => sporstmens; }

public void GetInformationAboutPlacesEveryCountry(string countryName, out int numberOFGold, out int numberOfSilver,

out int numberOfBronze,out List<Sporstman> countryWinners)

{

numberOFGold = 0;

numberOfSilver = 0;

numberOfBronze = 0;

countryWinners = new List<Sporstman>();

foreach (var item in Sporstmens)

{

if (item.Country == countryName)

{

if (item.Place == 1)

{

numberOFGold += 1;

countryWinners.Add(item);

}

if (item.Place == 2)

{

numberOfSilver += 1;

countryWinners.Add(item);

}

if (item.Place == 3)

{

numberOfBronze += 1;

countryWinners.Add(item);

}

}

else

{

throw new ArgumentException("This Country not exist(or not registered on the Olimpic list)");

}

}

}

}

## Функции программы

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

Запись всех данных в таблицу:

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

Функция изменения данных в БД:

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

Удаление из базы записей о спортсменах с указанной фамилией (по требованию допинг-контроля):

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

Выдача списка призёров по каждому виду спорта:

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

Выдача сведений о всех призёрах с заданным названием страны:

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

Запись информации в файл осуществлялось с помощью классов TextWriter и класса TextReader:

void saveToFiletxtToolStripMenuItem\_Click(object sender, EventArgs e)

{

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

{

try

{

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

{

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

{

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

{

if (j == 7)

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

else

writer.Write(RegistrationDataView.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();

}

}

}

void readFromFileToolStripMenuItem\_Click(object sender, EventArgs e)

{

using (TextReader text = new StreamReader(File.OpenRead($@"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'))

{

InfoFromFileRichTextBoxTaabIndex.Text += item.ToString();

}

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

}

else

{

MessageBox.Show("Empty file");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

text.Close();

}

}

}

## Библиотеки

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

using Kursach5Variant\_Olimpyc.Utility;

using Kursach5Variant\_Olympyc.Model.Model;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Windows.Forms;

# РАЗДЕЛ 2 СПРАВОЧНАЯ СИСТЕМА

## Краткое описание продукта

Приложение представляет собой ПО, в котором разработана база данных «С». Функциональность программы включает:

а) выдачу списка призёров по каждому виду спорта;

б) выдачу справок о количестве золотых, серебряных и бронзовых ме далей, завоёванных каждой страной - участницей Олимпиады;

в) выдачу сведений о всех призёрах страны с заданным названием;

г) удаление из базы записей о спортсменах с указанной фамилией (по требованию допинг-контроля);

## Как добавить Спортсмена

Для того, чтобы добавить *Спортсмена* и всю информацию относящейся к нему:

1. Вам нужно перейти в раздел "Registration"

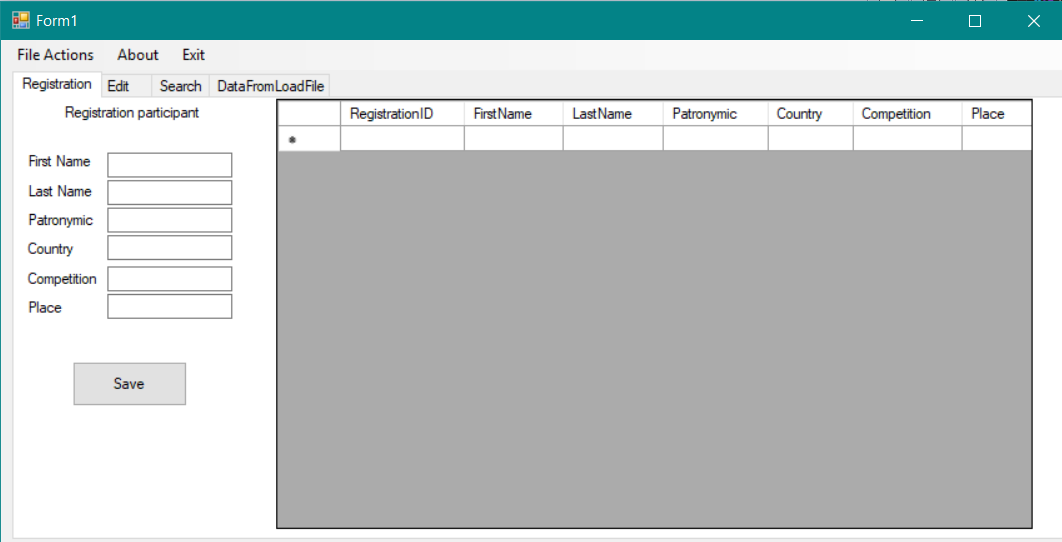


Рис. 1

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

## 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 Actions" и выбирите кнопку "Reas from file".

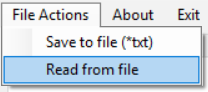


Рис. 2

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

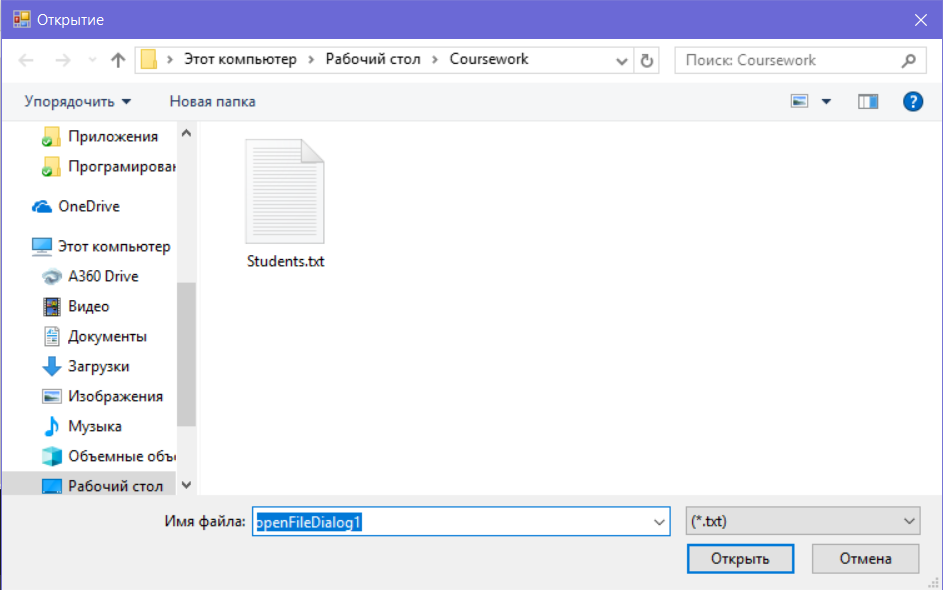


Рис. 5

## 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 Actions" и выберите кнопку "Save to file (\*txt)".

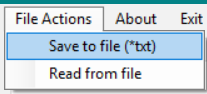


Рис. 3

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

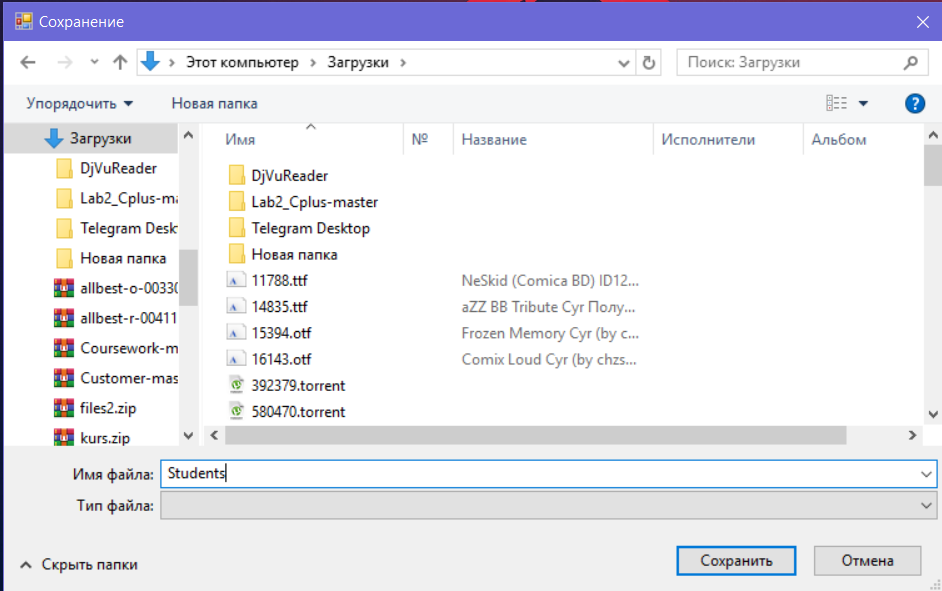


Рис. 7

# 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/>