Одеський національний політехнічний університет

Кафедра «Комп'ютеризовані системи управління»

**КУРСОВА РОБОТА**

з дисципліни «Програмування та теорія алгоритмів»

на тему: «Створення інформаційної бази»

Варіант 3

Студента 1 курсу, групи АТ-182

спеціальності «Автоматизація та комп'ютерно-інтегровані технології»

Бутескула Андрія Юрійовича

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 – 2019 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 3

Создать базу данных, содержащую результаты экзаменационной сессии студентов курса. Каждая запись хранит:

* фамилию, инициалы студента;
* *номер зачётки;*
* наименование факультета;
* курс;
* результаты сессии для каждого студента; представляет массив структур из 5 элементов; структура включает:

- название предмета;

- фамилию преподавателя;

- оценку;

*Предусмотреть:*

а) выдачу справки о студентах - отличниках;

б) выдачу справки о студентах, имеющих двойки;

в) выдачу справки о названии дисциплины, по которой наибольшее число студентов получили двойки;

г) удаление всех записей, у которых есть двойка хотя бы по одному предмету.

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 | 28.04.2019 |
|  | Теоретическая подготовка | Изучение литературы по дисциплине | 28.04.2019 |
|  | Проект | Создание репазитория и проекта в программе Visual Studio | 29.04.2019 |
|  | Дизайн | Создание дизайна для приложения | 29.04.2019 |
|  | Классы | Написания новых классов с полями поставленными в задании, а также свойств для работы с ними | 29.04.2019 |
|  | Связка | Настройка взаимосвязи класса с графическими инструментами | 29.04.2019 |
|  | DataGridView | Добавление инструмента DataGridView для вывода информации и дальнейшей работы | 30.04.2019 |
|  | Выдача отчётов | Работа над индивидуальным задание | 3.05.2019 |
|  | Експорт и импорт данных | Написание кода для сохранения и загрузки данных во внешний файл/из файла | 9.05.2019 |
|  | Меню | Проектирование и доработка меню для пользователя | 12.05.2019 |
|  | Корректность ввода | Написания функции контроля корректности всех вводимых данных | 13.05.2019 |
|  | Пароль | Добавление ввода пароля при выполнении операций изменения данных | 13.05.2019 |
|  | Справка | Создание справочной системы, для пользователей | 19.05.2019 |
|  | Оптимизация | Исправление ошибок и очищение не нужного кода | 21.05.2019 |
|  | Документация | Подготовка пояснительной записки о программе | 24.05.2019 |

## Описание проектирование программы

Основа программы состоит из двух классов: класс (Student) который состоит из четырех полей, а также свойств и конструкторов.

Пример кода данного класа:

class Student

{

private string name;

private string record\_number;

private string name\_faculty;

private string course;

}

public Student()

{

}

public Student(string name, string record\_number, string name\_faculty, string course)

{

Name = name;

Record\_number = record\_number;

Name\_faculty = name\_faculty;

Course = course;

}

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

Пример кода:

class Study

{

private string subject;

private string name\_teacher;

private double mark;

}

public Study(string subject, string name\_teacher, double mark)

{

Subject = subject;

Name\_teacher = name\_teacher;

Mark = mark;

}

public Study()

{

}

## 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:

void Input\_student()

Функция разблокировки кнопок:

void Button\_true()

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:

а) выдачу справки о студентах - отличниках;

б) выдачу справки о студентах, имеющих двойки;

в) выдачу справки о названии дисциплины, по которой наибольшее число студентов получили двойки;

г) удаление всех записей, у которых есть двойка хотя бы по одному предмету.

Запись информации в файл осуществлялось с помощью инструмента SaveFileDialog и класса BinaryWriter:

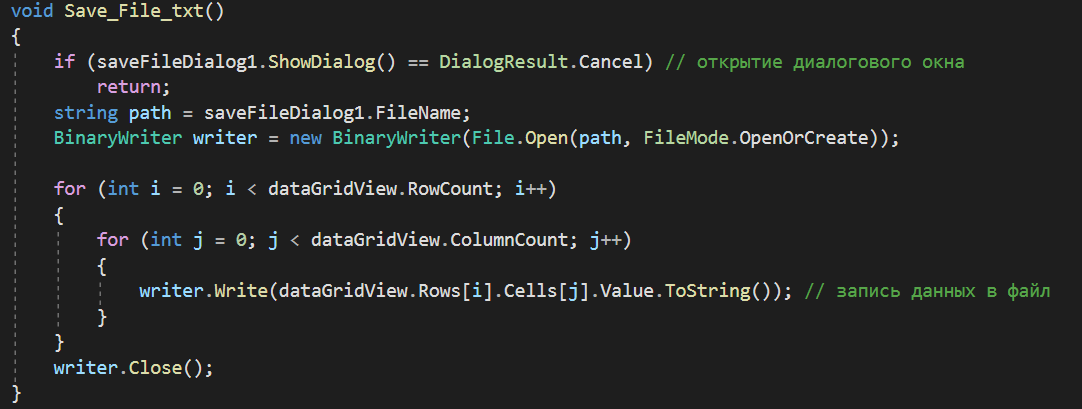


Рис. 1

## Библиотеки

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

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Windows.Forms;

using System.IO;

using Excel = Microsoft.Office.Interop.Excel;

using System.Diagnostics;

# РАЗДЕЛ 2 СПРАВОЧНАЯ СИСТЕМА

## Краткое описание продукта

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

* управление данными студента: добавление, удаление и редактирование;
* добавление новой записи после записи с указанной фамилией.
* удаление записей о студенте если есть хотя бы одна двойка;
* выдачу сведений о студентах отличниках и также двоечниках
* выдача справки о названии дисциплины с наибольшим числом оценки 2
* сохранение данных и открытие в файл/из файла

## Как добавить студента

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

1. Вам нужно перейти в раздел "Create"

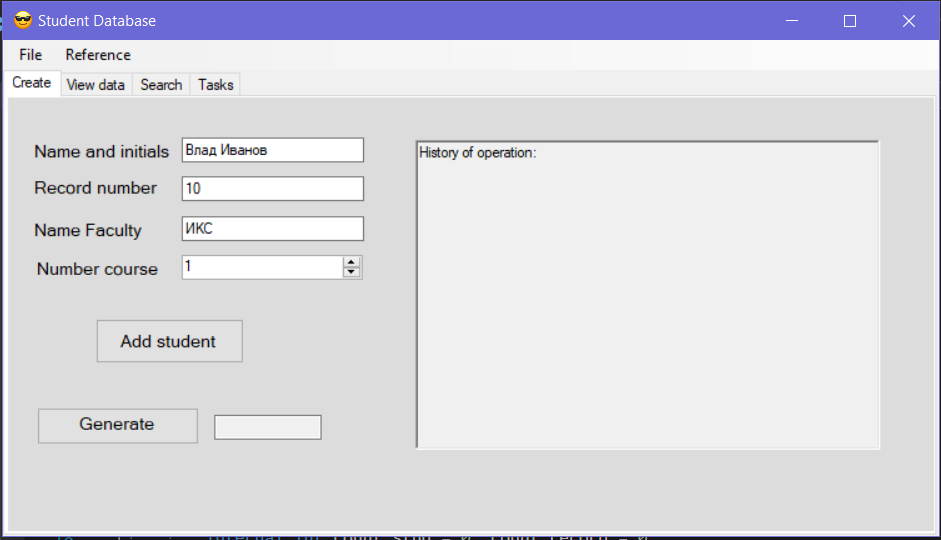
****

Рис. 2

1. Заполнить соответствующие поля нужной вам информацией (Пример можете посмотреть предварительно наведя на поле ввода)
2. Нажать на кнопку "Add student"
3. В новом появившеймся окне заполнить поля и нажать "Add subject"

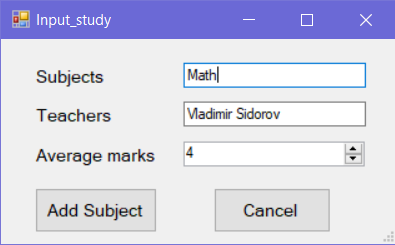
****

Рис. 3

1. Проделать шаг 4, для новых открывшихся окон
2. После этого, на экране появится окно, которое оповистит вас о успешном добавлении студента.

## 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 "Open txt".

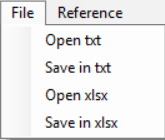


Рис. 4

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

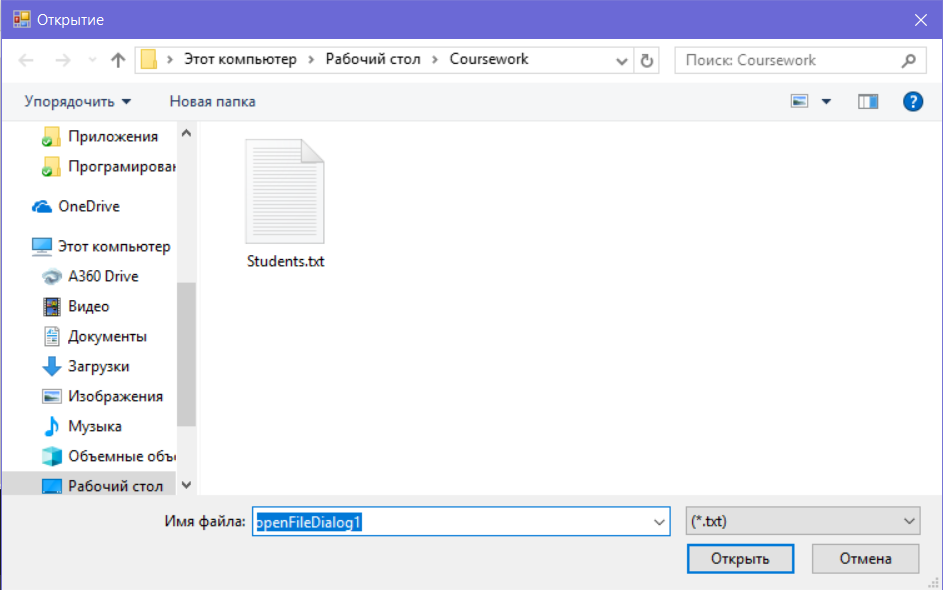


Рис. 5

2. Экспорт из файла с расшерением "xls"

Для этого надо предварительно нажать на кнопку "Open xls", появится тоже окно что и в первом случае и вы должны будете выбрать файл интересующий вас.

## 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 in txt".

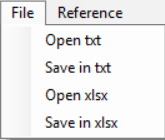


Рис. 6

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

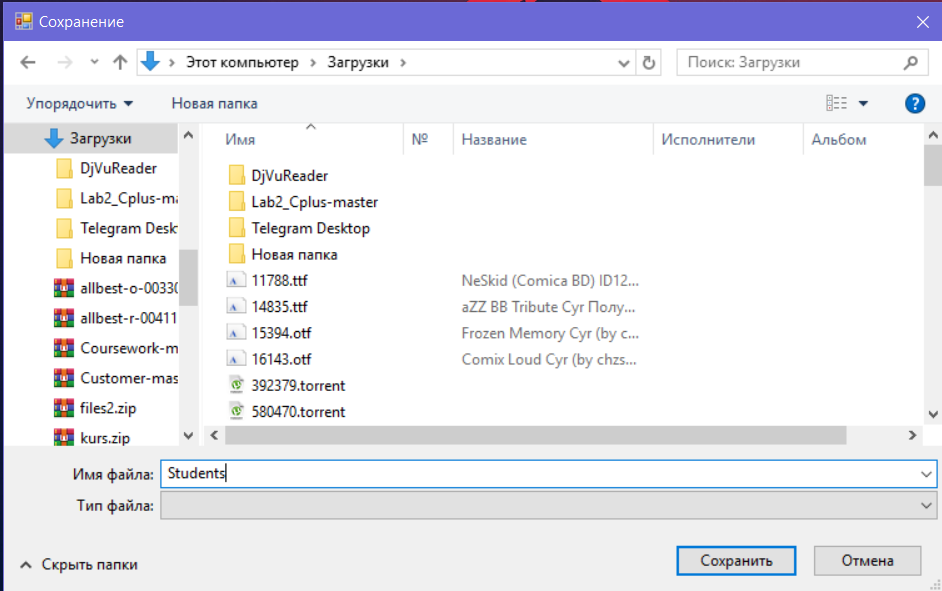


Рис. 7

2. Сохранение с расширением "xlsx"

По тому же принципу, в панели инструметов предварительно нажать на кнопку "Save xlsx", появится тоже окно что и в первом случае и вы должны будете путь и указать имя файла.

## Поиск и редактирование

Для поиска студента нужно:

1. Перейти в раздел "Search"

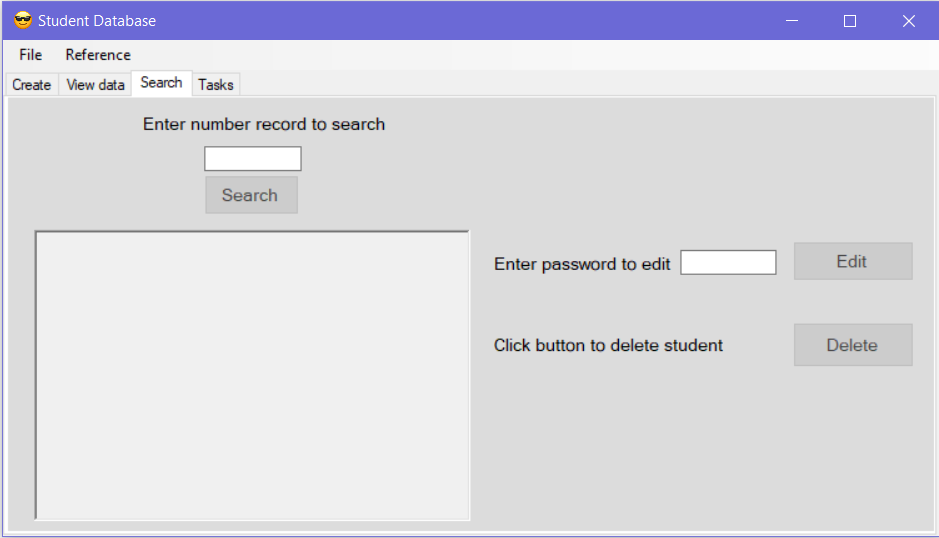


Рис. 8

2. Ввести в соответствующее поле номер зачетки студента которого вы ищите.

3. Нажать кнопку поиск.

Если студент с таким номером зачетки существует в базе, информация о нём выведется на экран, если студента не найдено выведется соответствующее окно, которое оповестит вас, что такого студента нету в базе.

После того как будет найдено студента, у вас появится возможность редактирования либо удаление.

Для редактирования выполните следующий алгоритм:

1. Сгенерируйте пароль в разделе "Create", для прав адмнистратора

2. Введите ваш пароль в соответствуюшее поле в разделе "Search" и нажмите "Edit"

3. В открывшемся окне, можете изменять данные

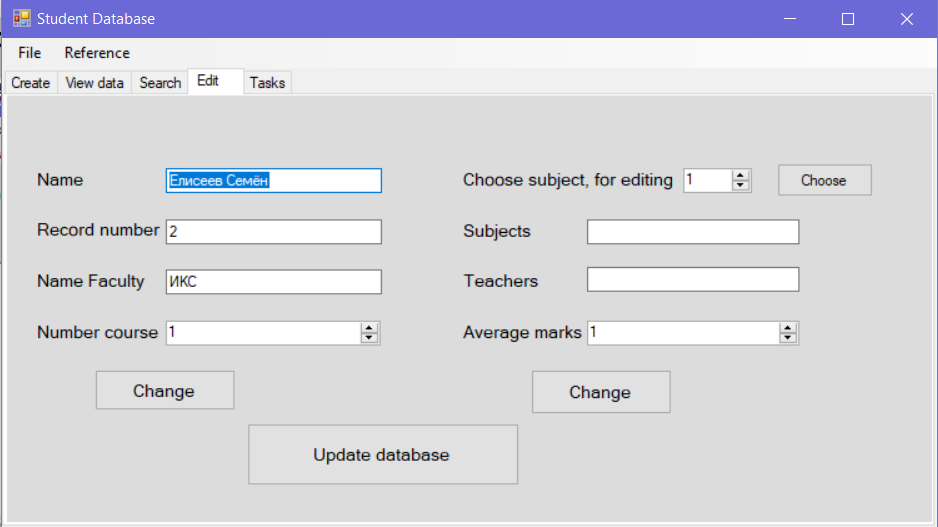


Рис. 9

Чтобы изменять информацию об успеваемости, выберете предмет и нажмите "Choose", после чего редакрируйте нужную вам информацию.

4. Нажмите "Change" для сохранения изминений и для завершения редактирования и просмотра новых отредактированных записей нажать "Update database".

## Выдача отчетов

Для того чтобы посмотреть индивидуальное задание:

1. Перейдите в раздел "Tasks"

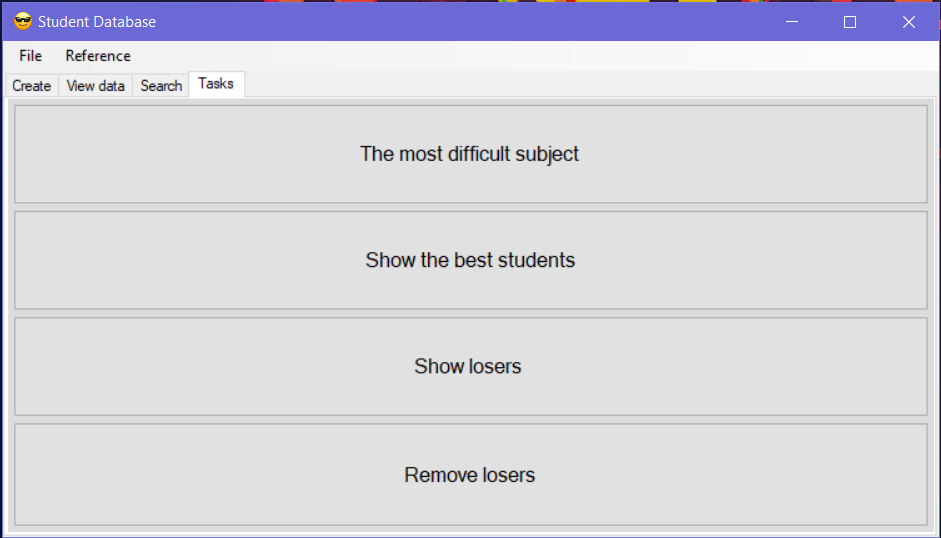
****

Рис. 10

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

В зависимости от выбранной вами кнопкой появится такая информация:

* "The most difficult subject" - Выдача справки о названии дисциплины,

по которой найбольшее число студентов получили двойки

* "Show the best students" - Выдача справки о студентах отличниках
* "Show losers" - Выдача справки о студентах двоечников

"Remove losers" - Удаление студентов что имеют хотя бы одну двойку

# 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/>