Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра Програмної інженерії

КУРСОВА РОБОТА

ПОЯСНЮВАЛЬНА ЗАПИСКА

з дисципліни “ Об’єктно - орієнтоване програмування”

Довідник абітурієнта

Керівник, ст. викл. Черепанова О. Ю.

Студент гр. ПЗПІ-20-4 Васильєв Д. О.

Комісія:

Ст. викл. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Черепанова Ю.Ю.

Доц. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Побіженко І. О.

Доц. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кравець Н. С.

Харків 2019

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ**

Кафедра: ***Програмної інженерії***

Дисципліна: ***Об’єктно-орієнтоване програмування***

Спеціальність: ***121***  ***Інженерія програмного забезпечення***

Освітня програма: ***Програмна інженерія***

Курс 1 .Група *ПЗПІ-20 -4* . Семестр *2* .

***ЗАВДАННЯ***

***на курсовий проект студента***

*Васильєва Дмитра Олександровича*

1 Тема проекту: *Зарахування абітурієнтів*

2 Термін здачі студентом закінченого проекту: ***“19” - червня - 2021 р***.

3 Вихідні дані до проекту:

*Основні функції програми для додавання, змінення, видалення та пошуку даних. Графічний інтерфейс, запис та зчитування файлу.*

4 Зміст розрахунково-пояснювальної записки:

*Вступ,* *специфікація програми, проектна специфікація, інструкція користувача,  висновки\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5 Перелік графічного матеріалу:

*Схема об’єктної моделі, алгоритми, приклади екранних форм* \_\_\_\_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |
| --- | --- | --- |
| *№* | *Назва етапу* | *Термін виконання* |
| 1 | Видача теми, узгодження і затвердження теми. | 24-02-2019 р. |
| 2 | Формулювання вимог до програми. | 29-03-2019 – 30-03-2019 р. |
| 3 | Розробка об’єктної моделі. | 01-04-2019 – 05-04-2019 р |
| 4 | Розробка підсистеми зберігання та пошуку спортсменів за критеріями. | 02-05-2019 – 08-05-2019 р. |
| 5 | Розробка функцій додавання та видалення із списку. | 13-05-2019 – 20-05-2019 р. |
| 6 | Розробка функцій зберігання та завантаження даних. | 21-05-2019 – 23-05-2019 р. |
| 7 | Тестування і доопрацювання розробленої програмної системи. | 24-05-2019 – 26-05-2019 р. |
| 8 | Оформлення пояснювальної записки, додатків, графічного матеріалу. | 27-05-2019 – 01-06-2019 р. |
| 9 | Захист | 05-06-2019 – 15-06-2019 р. |

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ студент Васильєв Д. О.

Керівник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ст. викл Черепанова О. Ю.

« 19 » лютого 2021 р.

**РЕФЕРАТ**

Пояснювальна записка до курсової роботи: 39 с., 26 рис., 1 додаток, 5 джерел.

КУРСОВИЙ ПРОЕКТ, МОВА ПРОГРАМУВАННЯ C #, ОБ'ЕКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ, ПЛАТФОРМА .NET, ЗАРАХУВАННЯ АБІТУРІЄНТІВ, НАВЧАННЯ, ВИЩА ОСВІТА, ВНЗ, АВТОРИЗАЦІЯ, ПОШУК.

Метою роботи є розробка програми “Зарахування абітурієнтів” на засадах об'єктно-орієнтованого програмування.

Методи розробки базуються на використанні середовища розробки JetBrains Rider та Microsoft Visual Studio 2019, Windows Forms, платформи .NET Framework 5.0, мови програмування C#.

В результаті була отримана програма під назвою “Зарахування абітурієнтів”, яка дозволяє додавати дані абітурієнтів до бази даних, такі як: ім’я, фамілія, по батькові, номер телефону, можливість вчитися на контрактній формі навчання, бали за 3 екзамени, бал атестату. Є можливість дивитися, видаляти та редагувати дані про спеціальності. Також, є функція пошуку спеціальності, вишу за будь-якою з характеристик. Є можливість запису даних у файл формату JSON.

ЗМІСТ

[Вступ 6](#_Toc532289188)

[1 Специфікація програми 7](#_Toc532289189)

[1.1 Інтерфейс користувача 7](#_Toc532289190)

[1.2 Окремі функції користувача 8](#_Toc532289191)

[2 Проектна специфікація 13](#_Toc532289192)

[2.1 Об’єктна модель програми 13](#_Toc532289193)

[2.2 Детальна інформація про класи  15](#_Toc532289194)

[2.3 Формат даних 22](#_Toc532289194)

[3 Інструкція користувача 24](#_Toc532289195)

[3.1 Запуск програми  24](#_Toc532289194)

[3.2 Робота програми  24](#_Toc532289194)

[Висновки 28](#_Toc532289196)

[Перелік джерел посилання 29](#_Toc532289196)

[Додаток А Код програми 30](#_Toc532289197)

ВСТУП

Метою даної курсової роботи є вивчення основних принципів об’єктно-орієнтованого програмування, та оволодіння навичками розробки об’єктних програм. Ця робота зосереджує студента на найбільш важливих, абстрактних рисах програмування, які не залежать від таких обставин, як тип процесора або операційна система. Об’єктно-орієнтоване програмування дуже слушно надає таку можливість, бо воно не залежить навіть від мови програмування, адже основні його принципи вдало застосовуються, переносяться та масштабуються на різних мовах.

Ця курсова робота дає студенту отримати широкий спектр знань в області об’єктно-орієнтованого програмування, а також навичок, необхідних для застосування цих вмінь на практиці. Як-от у створенні програмних додатків за допомогою засобів платформи .NET, Windows Forms, мови програмування C#, способу зберігання даних у пам’яті комп’ютера у форматі JSON.

Інструментарій Windows Forms містить в собі набір бібліотек, класів та методів, необхідних для розробки графічних програмних додатків для десктопних платформ Microsoft Windows, що мають на меті полегшення життя користувача у вирішенні буденних задач, питань бізнесу та освіти. Візуальний інтерфейс робить взаємодію людина-комп’ютер максимально-простою для сприйняття та розуміння.

Навички здобуті під час розробки даного курсового проекту допоможуть мені в майбутньому в опануванні подальшого інструментарію платформи .NET та створенні простих, зрозумілих, захищених від втручання програмних засобів у WEB, Desktop тощо.

**1 СПЕЦИФІКАЦІЯ ПРОГРАМИ**

* 1. Інтерфейс користувача

При запуску програми ­­­- головна форма, на якій знаходиться основний інтерфейс користувача(див.рис.1.1). За замовчуванням введено деякі значнення.

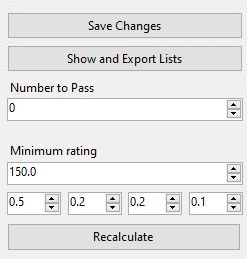
****

Рисунок 1.1 – Головна форма

До інтерфейсу користувача також входить форма з таблицею, що відображує записи в базі даних(див.рис.1.2):

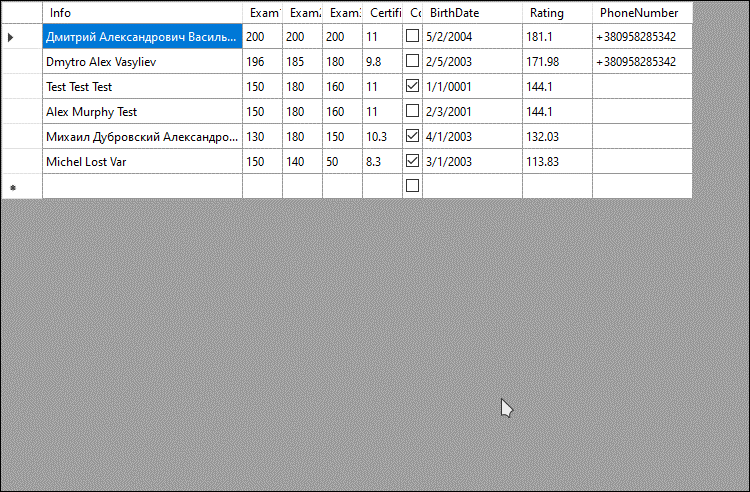


Рисунок 1.2 – Форма відображення

* 1. Окремі функції користувача

При заповнені усіх полів, користувачу пропонуються наступні можливості:

1. Зберігти списки зарахованих абітурієнтів, абітурієнтів до співбесіди та конфліктів(якщо є декілька людей з однаковим рейтинговим балом, не всі з яких пройшли конкурс), натиснувши кнопку «Show And Export Lists».
2. Натиснувши «Recalculate» користувач зможе перерахувати конкурсні бали абітурієнтів, змінивши бали за екзамени чи бал атестату. Зміни будуть відображені в полі «Rating».
3. Користувач має змогу побачити присутні, змінити та додати дані про спеціальності у таблиці, яка представлена за допомогою DataGridView в нижній частині форми (див.рис.1.5)

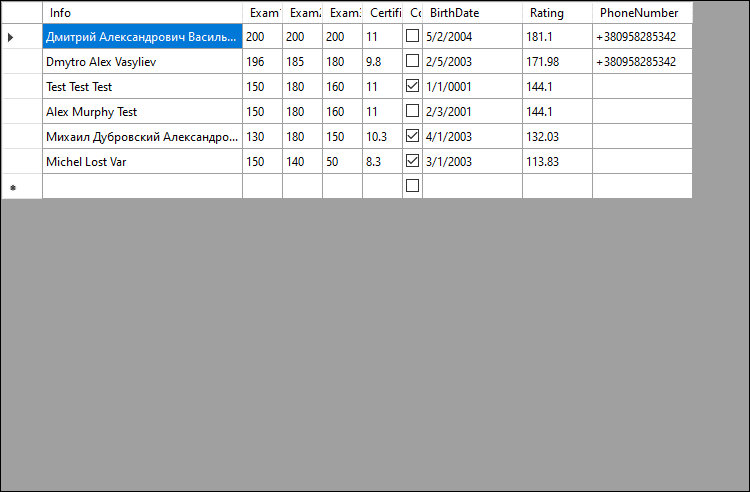


Рисунок 1.5 - Форма відображення даних

При використанні дочірньої форми, користувач може сортувати дані по відповідним критеріям, які занесені до таблиці(див.рис.1.6)

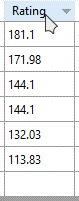


Рисунок 1.6 – Відсортована форма згідно рейтинговим балом

Користувач може додати нові записи до бази даних, якщо просто введе дані в останню лінію в формі. Вона помічена зірочкою.(див.рис.1.7)

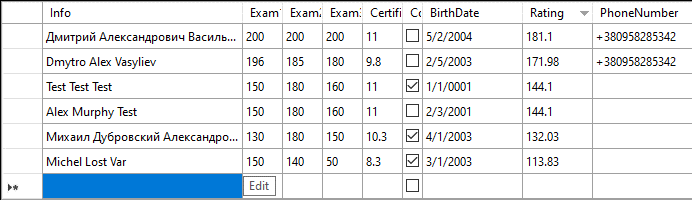


Рисунок 1.7 – Функція додавання

Нова лінія одразу ж з’явиться в списку та користувач може натиснути «Save changes» в правій частині екрану, щоб зберегти зміни до файлу(див.рис.1.1).

**2 ПРОЕКТНА СПЕЦИФІКАЦІЯ**

* 1. Об’єктна модель програми

Програма складається з 3-х проектів, кожен з которих репрезентує окремі частини додатку (див.рис.2.1).

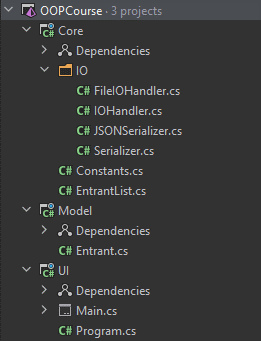


Рисунок 2.1 – Склад програми

* 1. Проект «Model»

У даному проекті знаходяться класи, які репрезентують модель інформації, що зберігається в файлах.

1. «Entrant.cs» - клас, що відповідає за зберігання інформації про абітурієнта(див.рис.2.4). Також у файлі містяться допоміжні структури, а саме:
   1. «Grades» - структура, що зберігає інформацію про оцінки абітурієнта(див.рис.2.2). Також має метод «Apply», що розраховує рейтинговий бал за переданих коефіцієнтів екзамену та атестату.

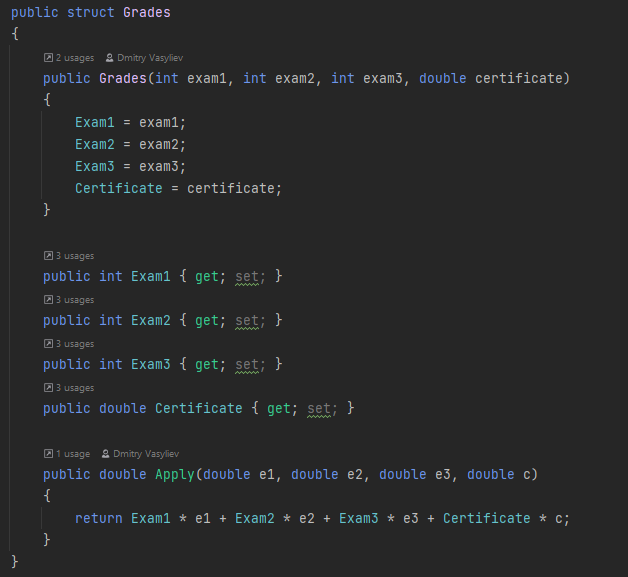


Рисунок 2.2 – структура «Grades»

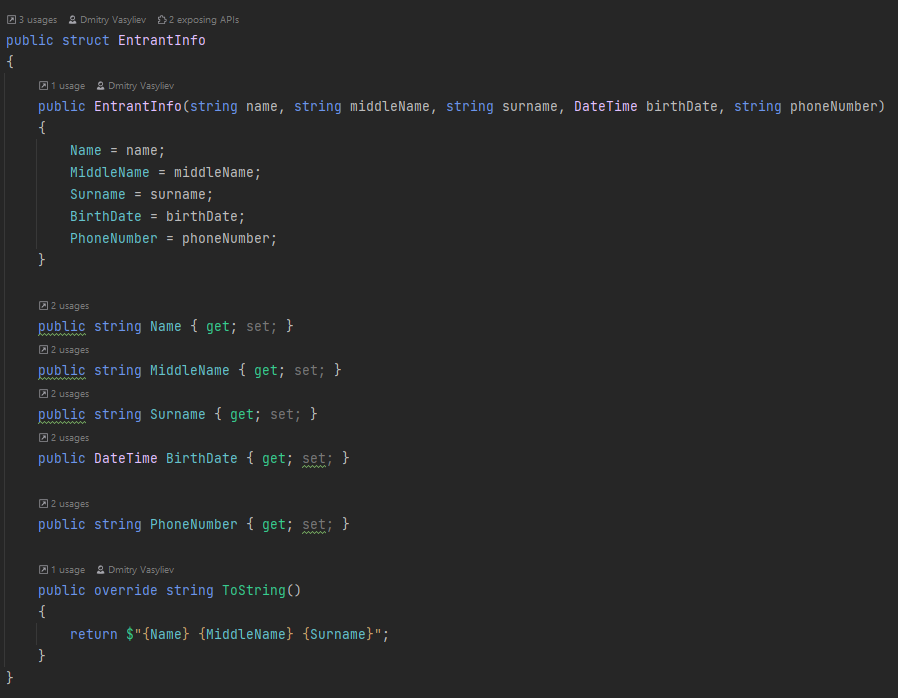
* 1. «EntrantInfo» - структура, що зберігає анектні дані абітурієнта(див.рис.2.3).

Рисунок 2.3 – структура «EntrantInfo»

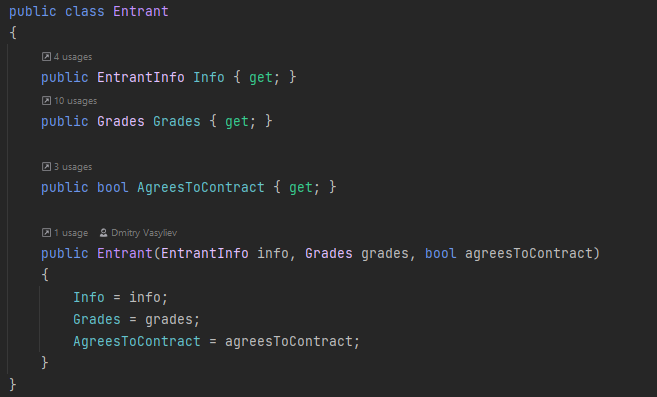


Рисунок 2.4 – структура «Entrant»

* 1. Проект «Core»

У даному проекті знаходяться класи, що відповідають за основну логіку програми.

1. «Constants.cs» - файл с константами, а саме назви файлів для збереження сформованних звітів.

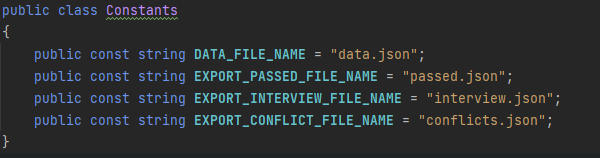


Рисунок 2.5 – клас «Constants»

1. «IO/IOHandler.cs» - інтерфейс, що забезпечує можливість читати та записувати список абітурієнтів з деякого джерела.
2. «IO/FileIOHandler.cs» - клас, що реалізує інтерфейс IOHandler для считування та запису списку абітурієнтів у файл. Перед тим як записати список у файл – клас кодує переданий список абітурієнтів у обраний формат за допомогою IOHandler. Щоб забезпечити можливу підтримку різних форматів, було використано поліморфізм: якщо потрібно буде додати можливість кодування списку до іншого формату – не треба змінювати код IOHandler, потрибно лише создати новий клас, що буде реалізовувати інтерфейс Serializer. Кодування та декодування проходять наступним чином(див.рис.2.6):

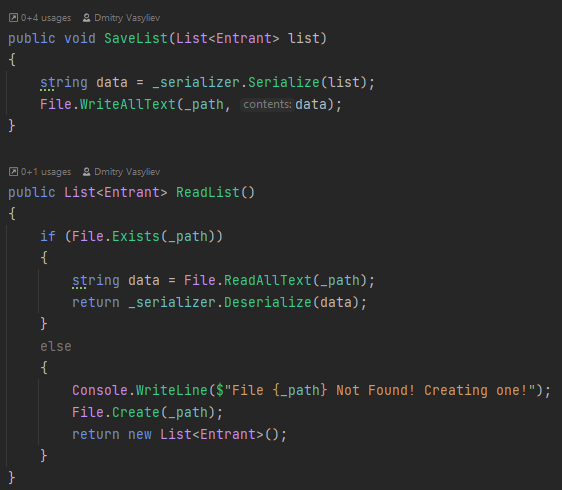


Рисунок 2.6 – фрагмент коду, де відбувається кодування списку у формат JSON та запис до файлу

1. «IO/Serializer.cs» - інтерфейс, що забезпечує можливість кодування та декодування списку абітурієнтів у деякий формат.
2. «IO/JSONSerializer.cs» - клас реалізує інтерфейс Serializer для кодування та декодування списку абітурієнтів у формат JSON, використовуючи бібліотеку System.Text.Json. Кодування(серіалізаціія) відбувається з обробкою можливих виключень наступним чином(див.рис.2.7):

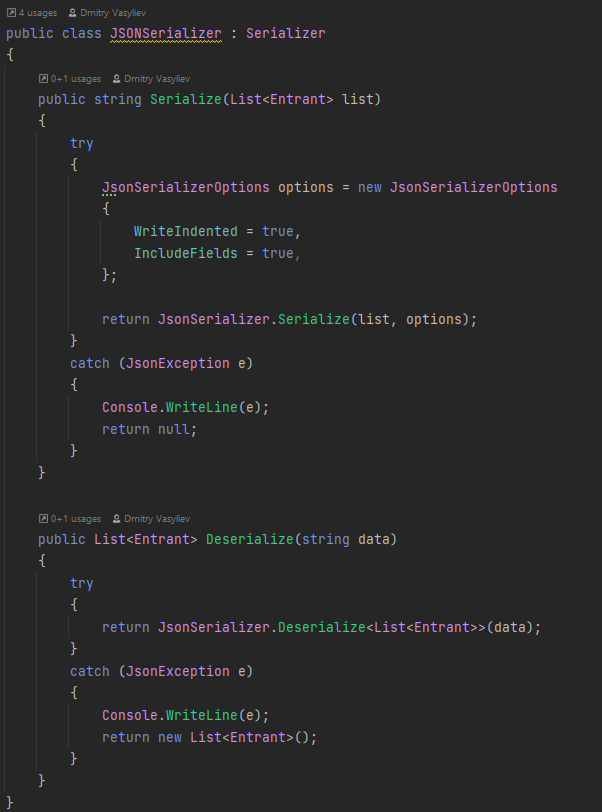


Рисунок 2.7 – фрамент коду, що зображує серіалізацію та десеріалізацію даних у формат JSON

1. «EntrantList.cs» - клас-контейнер, що зберігає список студентів та автоматизує його зчитування та запис у вказане джерело.

Об’єктна модель класів проекту у вигляді діаграми(див.рис.2.6):

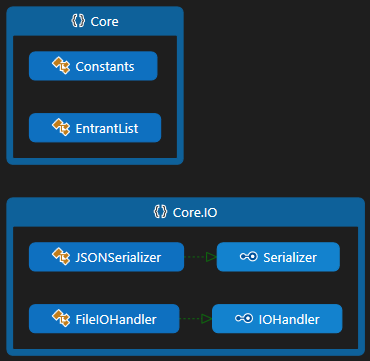


Рисунок 2.8 – об’єктна модель класів проекту «Core»

* 1. Проект «UI»

У цьому проекті зберігаються класи, що відповідають за інтерфейс користувача.

* 1. Формат зберігання даних

Для зберігання списків абітурієнтів та звітів використовується формат JSON.

Для серіалізації й десеріалізації об’єктів була підключена бібліотека System.Text.Json [3].

Фрагмент JSON файлу зображений на рисунку 2.7:

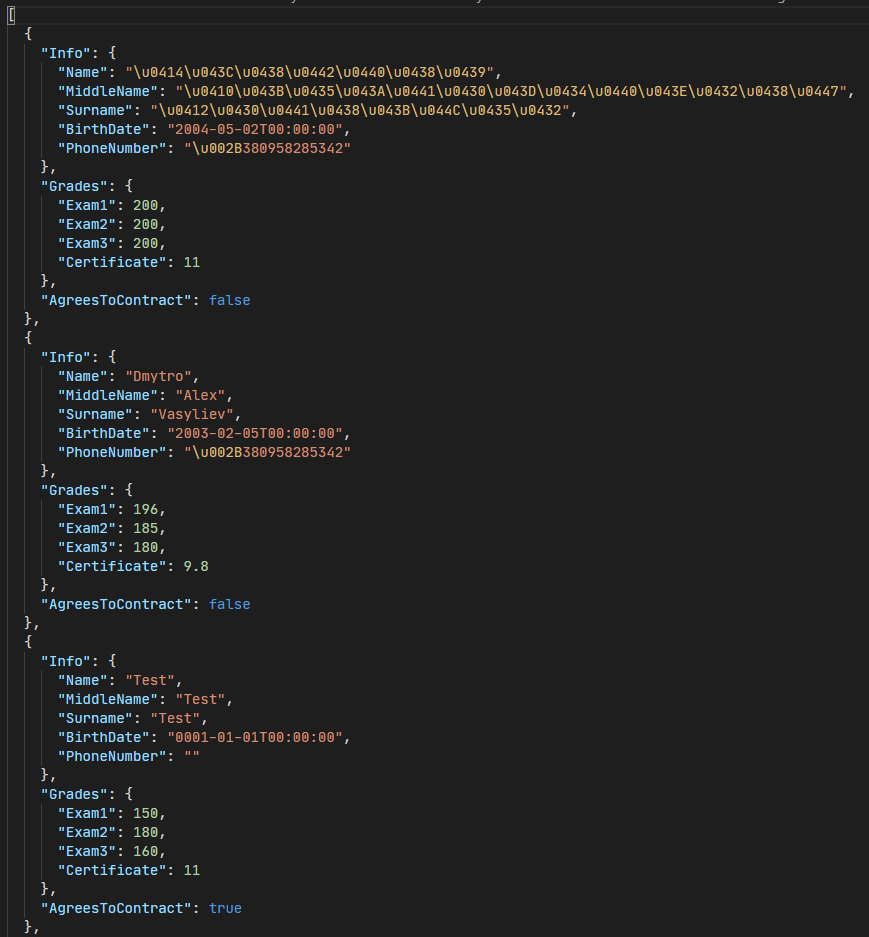


Рисунок 2.9 - Фрагмент з JSON-файлу, створенного додатком

Саме цей формат зберігання даних було обрано, тому що він має такі переваги:

- компактність(порівняно, наприклад, з XML);

- легко читається як людиною, так і комп’ютером;

- наявність в стандартній бібліотеці C# для роботи з JSON.

**3 ІНСТРУКЦІЯ КОРИСТУВАЧА**

3.1 Запуск програми

Для використання даного курсового проекту, перш за все, слід завантажити папку з файлами проекту. Надалі слід виконати наступне:

1. Відкрити папку з проектом, в якій буде файл для завантаження проекту через середовище розробки Visual Studio.
2. У папці з проектом потрібно знайти файл формату «.sln» та відкрити його.
3. Натиснути «CTRL + F5» для того щоб запустити проект.

3.2 Робота з програмою

При запуску проекту відкривається головне вікно в якому знаходяться текстові поля для заповнення даних таблиця та кнопки(див.рис.3.1).

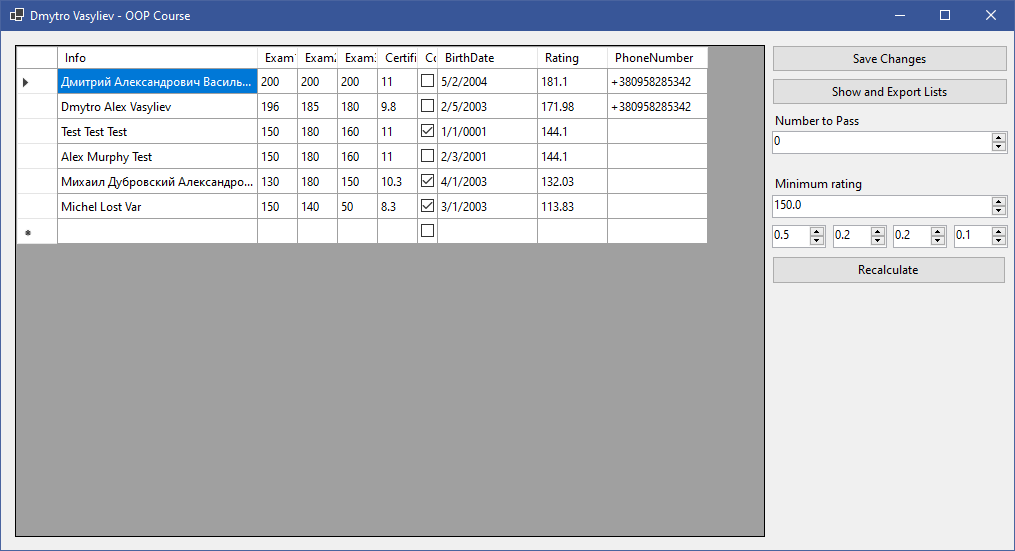


Рисунок 3.1 – Головне вікно

Надалі існує змога додати до таблиці та файл, якщо ввести в останню лінію дані та натиснути кнопку «Save Changes», змінити рейтингові умови та перерахувати рейтинг для всіх абітурієнтів в таблиці за допомогою кнопки «Recalculate», сформувати звіти кнопкою «Show and export lists».

Додаючи абітурієнтів до таблиці необхідно вказати анкетні дані(ім’я, фамілія, по батькові, дата народження, номер телефону), оцінки за 3 екзамени та середній бал атестату, помітити чи згоден абітурієнт вчитися на контрактній основі(див.рис.3.2).

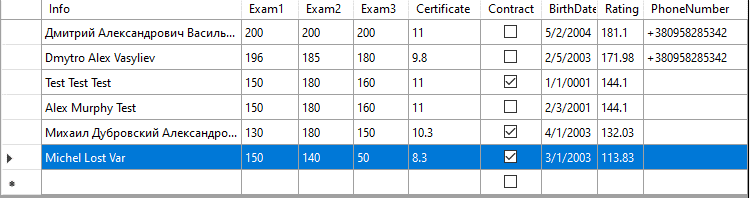


Рисунок 3.2 – приклад хаповнення таблиці

Для задання рейтингових умов слід вказати кількість абітурієнтів до зарахування, мінімальний рейтинговий бал та коефіцієнти для 3-х екзаменів та балу атестату(див рис.3.3).

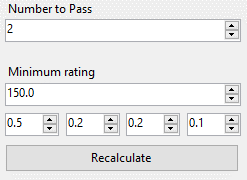


Рисунок 3.3 – приклад заповнення рейтингових умов

**ВИСНОВКИ**

У ході розробки курсового завдання я опанував базовий інструментарій бібліотеки Windows Forms. Я навчився створювати форми, змінювати їх вигляд, розрізняти поняття бекенду та фронтенду розробки додатків у Windows Forms, описувати базову поведінку додавання, видалення, пошуку та зміни даних. Я створив поведінку входу та виходу даних. Це було необхідно для того, щоб кожний користувач мав певний рівень доступу до даних – одні можуть лише переглядати, а інші – ще й змінювати. А система повинна лише мати інтерфейс для швидкого переходу між різними рівнями доступу.

Також у своїй роботі я застосував технологію JSON. Мені це було необхідно для збереження даних у жорсткій пам’яті комп’ютера для можливого доступу до цих файлів і в наступних запусках програми. Формат JSON є достатньо простим та швидким для опанування, а найголовніше – негроміздким для збереження даних. А тому і був обраний мною як пріоритетний посеред інших.

Мова програмування використана мною при розробці додатку – C#. Це одна з офіційних мов платформи .NET, вважається основною мовою компанії Microsoft та підтримується широким колом однодумців та прихильників. Мова представляє собою програмний комплекс для маніпуляції над класами/об’єктам пакету Windows Forms, підтримує широкий спектр бібліотек, необхідних для роботи з об’єктами, дозволяє програмно створювати вирішення багатьох проблем, я от у цій курсовій роботу – розробці довідника абітурієнтів.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Бондарев В.М. Объектно-ориентированное программирование на С#: учебное пособие. Харьков: СМИТ, 2009г. 221 с.
2. Shildt, H. C# 4.0 The Complete Reference. McGraw-Hill Education. New York: NY, 2010. 976с.
3. Documentation .NET. URL: https://www.udemy.com/ (дата звернення: 01.04.2019).
4. Documentation Windows Forms. URL: https://docs.microsoft.com/en-us/dotnet/framework/winforms/ (дата звернення: 01.04.2019).
5. Tutorial JSON URL: https://www.codeguru.com/csharp/csharp/cs\_data/creating-a-json-file-in-c.html (дата звернення: 10.04.2019).

**ДОДАТОК А**

Код програми