**Міністерство освіти і науки України**

**Національний авіаційний університет**

**Навчально-науковий інститут комп’ютерних**

**інформаційних технологій**



Лабораторна робота №3.3

з дисципліни «Об’єктивно-орієнтоване програмування»

«Дослідження механізму серіалізації та типової трирівневої архітектури»

Виконав студент:

групи ПІ-224Б

Ляшенко Б.М.

Перевірив викладач:

Домків Т.С

Київ 2022

Історія контролю

|  |  |  |
| --- | --- | --- |
| **Дата** | **Опис** | **Автор** |
| 16.10.2022 | Дослідження механізму серіалізації та типової трирівневої архітектури | Ляшенко Богдан Михайлович |

**Мета**: дослідити механізм серіалізації у мові програмування С# та навчитися розділяти функціонал застосування між рівнями архітектури.

**Завдання**:

**Частина 1**. Дослідження механізму серіалізації. Усі завдання частини 1 допускається реалізовувати в одному проекті одного рішення

1. Описати клас, заданий варіантом (табл 1), та передбачити для нього можливість серіалізації.
2. Створити масив об’єктів класу, вказаного в п.1, та серіалізувати їх у файл.
3. Створити новий масив та відновити в ньому значення серіалізованих об’єктів.
4. Виконати серіалізацію та десеріалізацію об‘єктів будь-якої колекції. Порівняти з масивом.
5. Продемонструвати використання бінарної серіалізації, XML-серіалізації, JSON та користувацької серіалізації:

* для отримання оцінки «задовільно» реалізувати будь-яку серіалізацію
* для отримання оцінки «добре» реалізувати бінарну, користувацьку, XML та JSON серіалізацію.

**Частина 2**. Дослідження використання серіалізації в типовій трирівневій архітектурі

застосування

1. Спроектувати застосування із розділенням рішення (solution) на 3 рівні, які реалізовуються 3- ма різними проекти: рівень доступу до даних (DAL), рівень бізнес-логіки (BLL), рівень представлення (PL).

В рішенні (solution) лабораторної роботи повинні бути:

* Проект з класом Program, в якому є метод Main(), за допомогою якого запускається застосування. В методі Main() повинен бути тільки виклик методу MainMenu() з класу Menu().
* Проект з класом EntityService. На цей проект повинен посилатися проект з класом Menu.
* Проект з класом EntityContext. На цей проект повинен посилатися проект з класом EntityService.
* В кожному проекті можуть бути також допоміжні класи та інтерфейси. Так само, як і в кожному класі можуть бути також інші члени, окрім тих, що вказані в завданні. Також можна додавати в рішення інші проекти, які будуть сприяти ізоляції рівнів.

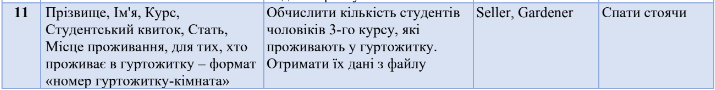
1. Реалізувати завдання за варіантом з табл. 2. Передбачити взаємодію з кожною сутністю. А також збереження та читання з файлів за допомогою механізму серіалізації:

* для отримання оцінки «задовільно» дозволяється реалізувати тільки клас (-и) EntityContext на рівні DAL, EntityService на рівні BLL та Menu на рівні BLL (за необхідності – статичний). Клас Entity може використовуватися будь-де. Може бути реалізований тільки один спосіб серіалізації (як в ч. 1);
* для отримання оцінки «добре» необхідно реалізувати на рівні BLL власний клас (-и) Exception (відповідно предметній області). Обробка виняткових ситуацій повинна бути не в момент їх виникнення/ генерації. Серіалізація повинна бути XML або JSON;
* для отримання оцінки «відмінно» клас(-и) Entity повинен бути тільки на рівні DAL. Для зменшення зв’язків між рівнями, в BLL та PL повинні бути реалізовані власні класи сутностей. Клас Menu повинен бути в окремому проекті. Крім імплементації бінарної, XML, JSON та користувацької серіалізації, користувач повинен мати можливість вибору типу та імені файлу, в який потрібно серіалізувати дані. Для цього назва файлу та розташування повинні передаватися з рівня представлення (PL) на нижчі рівні. А також - створити абстракцію DataProvider, яка відповідає за роботу з файлами відповідного типу (для кожного типу серіалізації – свій провайдер). Уникати повторення коду шляхом використання абстракцій та узагальнень.

**Індивідуальне Завдання**

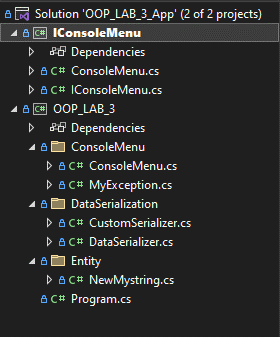
**Варіант № 11**



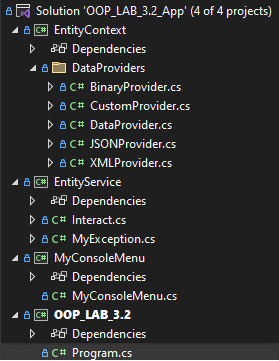


**Хід роботи**

1. **Частина 1**



1. **Частина 2**



**Висновок**: під час виконання індивідуального завдання лабораторної роботи, я дослідив механізм серіалізації у мові програмування С# та навчився розділяти функціонал застосування між рівнями архітектури.