

## МОДУЛЬ 3

### Лабораторна робота 3.3

#### ДОСЛІДЖЕННЯ МЕХАНІЗМУ СЕРІАЛІЗАЦІЇ ТА ТИПОВОЇ ТРИРІВНЕВОЇ АРХІТЕКТУРИ

**Мета роботи** – дослідити механізм серіалізації у мові програмування C# 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 реалізуються 3-ма різними проектами: 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.

2. 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.

### Загальна структура рішення

Клас **Entity** (відповідного типу, вказаному у варіанті), об'єкт (-и) якого містить інформацію про визначену сутність, наприклад, студента.

Клас **EntityContext**, що містить узагальнений набір методів для створення/відкриття файлу бази даних (БД), оновлення, закриття та видалення файлу.

При реалізації завдання на оцінку «відмінно» в рішенні буде клас(-и) **DataProvider**, який відповідає за конкретний спосіб серіалізації. Наприклад, **JSONProvider** матиме методи серіалізації та десеріалізації об'єктів в json файли.

Клас **EntityService**, який містить методи додавання, видалення та пошуку інформації про визначену сутність. Для виконання необхідних операцій дані повинні отримуватися або після взаємодії з класом **EntityContext** (наприклад, прочитати всю інформацію з файлу), або після їх введення користувачем у класі **Menu**. Клас **EntityService** не повинен безпосередньо працювати з файлами та взаємодіяти з кінцевим користувачем через консоль. Використання класу *Console* та *Stream* з похідними від них в цьому класі **забороняється**.

Клас **Menu**, в якому містяться методи для роботи з консоллю (введення/ виведення даних користувачем через інтерфейс).

### Архітектура рішення

Повинна бути реалізована 3-рівнева архітектура застосування.

Клас **Menu** – найвищий рівень (рівень представлення), який призначений для взаємодії з користувачем, та відповідає за виклик відповідних методів класу **EntityService** і передачу туди введених користувачем даних.

Клас **EntityService** – проміжний рівень (рівень бізнес логіки). У випадку, якщо, наприклад, передбачається тільки запис у файл введеної користувачем інформації без жодної обробки, то відповідний метод класу **EntityService** буде викликати необхідний метод класу **EntityContext**.

Клас **EntityContext** – найнижчий рівень (рівень доступу до даних), який відповідає за фізичну роботу з даними. В цьому класі відбувається безпосередній запис у файл та зчитування з нього.

Клас **DataProvider** знаходиться також на рівні DAL. При цьому може бути реалізований в окремому проекті.

Дані, з якими відбувається маніпуляція отримуються через об'єкт класу **Entity**. Об'єкти класу **Entity** допускається використовувати на кожному рівні (крім «відмінно»).

Таблиця 1

### Варіанти

Номер варіанта	Клас, що серіалізується		
	дані	методи та властивості	кількість об'єктів
1	Студент: Номер квитка, ПІБ, рік народження, Номер групи	Переведення на наступний курс; обчислення віку; виведення інформації про студента	3
2	Книга: Серійний номер, назва, рік видання, вартість примірника, кількість примірників	Збільшення вартості на певний процент; загальна вартість тиражу; виведення інформації про книгу	4
3	Товар: Код, назва, виробник, вартість, кількість у партії	Загальна вартість партії; збільшення вартості на певний процент; виведення інформації про товар	5
4	Коло: Колір заповнення, колір контуру, радіус	Обчислення площі; обчислення окружності; виведення інформації про фігуру	6
5	Прямокутник: Колір заповнення, колір контуру, довжини сторін	Обчислення площі; обчислення периметра; виведення інформації про фігуру	5

Номер варіанта	Клас, що серіалізується		
	дані	методи та властивості	кількість об'єктів
6	Трапеція: Колір заповнення, колір контуру, довжини основ, висота	Обчислення площі; обчислення периметра; виведення інформації про фігуру	4
7	Вектор: Колір лінії, координати кінця вектора	Обчислення довжини вектора; збільшення вектора на певний відрізок; виведення інформації про вектор	3
8	Рядок: Значення, довжина	Пошук заданого символу; зміна порядку наступності на протилежний; додавання нового рядка до існуючого; виведення рядка	4
9	Вектор: Колір лінії, полярні координати (довжина і кут)	Зменшення вектора в певну кількість разів; збільшення вектора в певну кількість разів; виведення інформації про вектор	5
10	Рядок: Значення, довжина, закритий ключ шифрування (різниця кодів та напрямки)	Шифрування рядка(заміна значущих символів на символи з кодом, більшим або меншим на певне значення); дешифрування рядка; виведення рядка	6
11	Рядок: Значення, довжина	Пошук підрядка; вставка підрядка; заміна підрядка; виведення рядка	5
12	Рахунок: Код власника, сума	Поповнення рахунку; зняття готівки; переведення на іншого власника; перерахування за певним курсом; виведення інформації про рахунок	4
13	Товар: Назва, код, дата виготовлення, термін придатності	Визначення придатності до споживання; визначення кінцевого строку реалізації; виведення інформації про товар	3
14	Рядок: Значення, довжина	Отримання підрядка з цифрами; отримання підрядка з буквами; перегрупування символів: спочатку букви, потім цифри; виведення рядка	4
15	Рейс: №, дата, пункт вибуття, пункт прибуття, час виїзду, час прибуття	Тривалість рейсу; час, що минув після закінчення рейсу; виведення інформації по рейс	5

Таблиця 2

№	Елементи класу Student	Операції зі студентами	Додаткові сутності	Додаткові вміння
1	Прізвище, Ім'я, Курс, Студентський квиток, Дата народження у форматі XX-XX-XXXX	Обчислити кількість студентів 3-го курсу, які народилися влітку. Отримати їх дані з файлу	Teacher, Astronaut	Співати
2	Прізвище, Ім'я, Курс, Студентський квиток, Місто прибуття, серія та номер паспорту	Обчислити відсоток студентів 1-го курсу, які приїхали з інших міст. Отримати їх дані з файлу	TaxiDriver, Acrobat	Танцювати
3	Прізвище, Ім'я, Курс, Студентський квиток, Стаття, Місце проживання, для тих, хто проживає в гуртожитку – формат «номер гуртожитку.кімната»	Обчислити кількість студенток 1-го курсу, які проживають у гуртожитку. Отримати їх дані з файлу	Musician, Pilot	Кататися на ковзанах

4	Прізвище, Ім'я, Курс, Студентський квиток, Служба в армії, для тих, хто служив в армії – номер військового квитка	Обчислити кількість студентів 5-го курсу, які служили в армії. Отримати їх дані з файлу	FootballPlayer, Lawyer	Декламувати вірші
5	Прізвище, Курс, Студентський квиток, Середній бал, Країна, номер залікової книжки	Обчислити кількість студентів 3-го курсу, які проживають в Україні. Отримати їх дані з файлу	McdonaldsWorker, Manager	Грати в шахи
6	Прізвище, Ім'я, Курс, Студентський квиток, Хобі - спорт, ідентифікаційний код	Обчислити кількість студентів 2-го курсу, які займаються спортом. Отримати їх дані з файлу	Courier, Fireman	Грати на гітарі
7	Прізвище, Ім'я, Ріст, Вага, Студентський квиток, серія та номер паспорту	Обчислити кількість студентів з ідеальною вагою тіла (Ріст – 110 = Вага) . Отримати їх дані з файлу	Librarian, SoftwareDeveloper	Кататися на велосипеді
8	Прізвище, Ім'я, Курс, Студентський квиток, Дата народження у форматі XX.XX.XXXX	Обчислити кількість студентів 4-го курсу, які народилися навесні. Отримати їх дані з файлу	Baker, Entrepreneur	Стрибати з парашутом
9	Прізвище, Ім'я, Курс, Студентський квиток, Дата народження у форматі XXXX XX XX	Обчислити кількість студентів 2-го курсу, які народилися взимку. Отримати їх дані з файлу	Painter, Farmer	Плавати
10	Прізвище, Ім'я, Курс, Студентський квиток, Стать, Місце проживання, номер залікової книжки	Обчислити кількість студенток 5-го курсу, які постійно проживають у Києві. Отримати їх дані з файлу	Joiner, Photographer	Множити великі числа
11	Прізвище, Ім'я, Курс, Студентський квиток, Стать, Місце проживання, для тих, хто проживає в гуртожитку – формат «номер гуртожитку-кімната»	Обчислити кількість студентів чоловіків 3-го курсу, які проживають у гуртожитку. Отримати їх дані з файлу	Seller, Gardener	Спати стоячи
12	Прізвище, Курс, Студентський квиток, Середній бал, Країна, номер закордонного паспорту	Обчислити кількість іноземних студентів-відмінників 1-го курсу. Отримати їх дані з файлу	Tailor, Singer	Пірнати
13	Прізвище, Ім'я, Курс, Студентський квиток, Стать, Середній бал, ідентифікаційний код	Обчислити кількість студенток-відмінниць 5-го курсу. Отримати їх дані з файлу	Storyteller, Dentist	Готувати
14	Прізвище, Ім'я, Курс, Студентський квиток, Стипендія, серія та номер паспорту	Обчислити кількість студентів 4-го курсу, які не отримують стипендії. Отримати їх дані з файлу	Postman, Mountaineer	Малювати
15	Прізвище, Ім'я, Курс, Студентський квиток, Стать, Середній бал, номер залікової книжки	Обчислити кількість студентів-чоловіків 2-го курсу, які займаються на відмінно. Отримати їх дані з файлу	Doctor, Mechanic	Вигадувати історії

### Контрольні запитання та завдання

1. Розкрийте сутність та призначення механізму серіалізації.
2. Які типи серіалізації існують в C#?
3. Опишіть механізм стандартної серіалізації.
4. Опишіть механізм бінарної серіалізації.
5. Опишіть механізм користувацької серіалізації.
6. Опишіть механізм XML серіалізації.
7. Опишіть механізм JSON серіалізації.
8. В чому полягає концепція багаторівневої архітектури застосування? Для чого вона потрібна?

9. Які існують рівні і як вони зв'язані між собою?
10. Чому необхідно ізолювати рівні чи окремий функціонал?
11. Як розподіляються класи між окремими рівнями та проектами в рішенні?
12. Яким чином допомагає абстракція при проектуванні багаторівневої архітектури?
13. Чим можуть бути корисні узагальнення та колекції при серіалізації та проектуванні багаторівневої архітектури?
14. Чому необхідно, щоб метод виконував тільки одну задачу?