**Практичне завдання**

**Stream API**

***Рівень 1: Базовий***

1. Завдання: Є список об'єктів класу `Person`, що містить поля `name`, `age`, та `city`. Використовуючи Streams API, створіть мапу, де ключем є місто, а значенням – список імен людей, які живуть в цьому місті.

2. Завдання: Є список об'єктів класу `Transaction`, що містить поля `id`, `amount`, `type` (наприклад, "credit" або "debit"). Знайдіть суму всіх "credit" транзакцій.

3. Завдання: Є список рядків, що представляють речення. Використовуючи Streams API, створіть мапу, де ключем є слово, а значенням – кількість його використання в усіх реченнях.

***Рівень 2: Середній***

1. Завдання: Є список об'єктів класу `Employee`, що містить поля `name`, `salary`, `department`. Використовуючи Streams API, знайдіть трьох працівників з найвищою зарплатою в кожному відділі та поверніть їх у вигляді мапи, де ключем є відділ, а значенням – список працівників.

2. Завдання: Є складна структура даних, що містить список об'єктів `Order`, кожен з яких має список `Item`. Використовуючи Streams API, знайдіть загальну кількість різних продуктів, які були замовлені.

3. Завдання: Є список об'єктів класу `Student`, що містить поля `name`, `grades` (список оцінок). Використовуючи Streams API, знайдіть середню оцінку кожного студента та поверніть мапу, де ключем є ім'я студента, а значенням – його середня оцінка. Відсортуйте мапу за середніми оцінками в порядку спадання.

***Рівень 3: Високий***

1. Завдання: Є класи `Book`, `Author`, та `Publisher`, де:

- `Book` має поля `title`, `price`, та посилання на `Author` і `Publisher`.

- `Author` має поля `name` та `age`.

- `Publisher` має поля `name` та `country`.

Використовуючи Streams API, створіть мапу, де ключем є країна видавця, а значенням – список авторів з цієї країни, які мають книги, вартість яких перевищує $50.

- Підказка: Використовуйте методи `filter`, `collect`, `groupingBy`, `Collectors.mapping`.

2. Завдання: Є класи `Course`, `Student`, та `Enrollment`, де:

- `Course` має поля `title`, `credits`.

- `Student` має поля `name`, `year`.

- `Enrollment` містить посилання на `Course` та `Student`, а також поле `grade`.

Використовуючи Streams API, знайдіть середню оцінку для кожного курсу, якщо курс відвідували студенти старших курсів (припустимо, що старші курси — це ті, у яких `year > 2`).

- Підказка: Використовуйте методи `filter`, `collect`, `Collectors.groupingBy`, `Collectors.averagingDouble`.

3. Завдання: Є класи `Team`, `Player`, та `Match`, де:

- `Team` має поля `name`, `city`.

- `Player` має поля `name`, `position`, та посилання на `Team`.

- `Match` має поля `date`, `team1`, `team2`, та список `Player` (що грали в матчі).

Використовуючи Streams API, знайдіть всіх гравців, які зіграли більше ніж 3 матчі за одну команду, та створіть мапу, де ключем є назва команди, а значенням – список таких гравців.

- Підказка: Використовуйте методи `filter`, `flatMap`, `groupingBy`, `Collectors.counting`.