

**ДЕРЖАВНА СЛУЖБА УКРАЇНИ З НАДЗВИЧАЙНИХ СИТУАЦІЙ
ЛЬВІВСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ БЕЗПЕКИ
ЖИТТЄДІЯЛЬНОСТІ**

Факультет цивільного захисту

Кафедра інформаційних технологій та систем електронних комунікацій

ЗАТВЕРДЖУЮ

Начальник кафедри інформаційних
технологій та систем електронних
комунікацій,
кандидат технічних наук, доцент
підполковник служби цивільного
захисту

_____ Олександр ПРИДАТКО
« ____ » _____ 20 ____ року

**МЕТОДИЧНА РОЗРОБКА
ДО ПРАКТИЧНИХ ЗАНЯТЬ З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Навчальна дисципліна: Об'єктно-орієнтоване програмування

Рівень вищої освіти, курс перший(бакалаврський), 2 курс

Спеціальність (спеціалізація) : 122 Комп'ютерні науки

Освітня програма: Комп'ютерні науки

Форма здобуття освіти денна

Розробник:

Старший викладач кафедри
інформаційних технологій та
систем електронних комунікацій
доктор філософії

_____ Юлія НАЗАР

Методичну розробку розглянуто та затверджено на засіданні кафедри
інформаційних технологій та систем електронних комунікацій,
протокол від « ____ » _____ 20 ____ № ____

Практичне заняття № 10

Тема: Використання Stream API для обробки колекцій в Java

Мета: Ознайомитися з основними методами Stream API, навчитися застосовувати filter(), map(), sorted(), collect() та інші операції для обробки даних.

Час: 2 академічні години

Хід виконання роботи

1. Дано клас Student із полями name та grade.

```
class Student {  
    private String name;  
    private int grade;  
  
    public Student(String name, int grade) {  
        this.name = name;  
        this.grade = grade;  
    }  
  
    public String getName() { return name; }  
    public int getGrade() { return grade; }  
  
    @Override  
    public String toString() {  
        return name + " - " + grade;  
    }  
}
```

2. В main-методі створено список студентів:

```
List<Student> students = Arrays.asList(  
    new Student("Анна", 85),  
    new Student("Олексій", 75),  
    new Student("Марія", 90),  
    new Student("Іван", 60),  
    new Student("Олег", 95),  
    new Student("Тетяна", 88)  
);
```

3. Із використанням колекцій та лямбда виразів зробити:
 - Відфільтрувати студентів із середньою оцінкою > 80:

```
List<Student> filteredStudents = new ArrayList<>();  
for (Student student : students) {  
    if (student.getGrade() > 80) {  
        filteredStudents.add(student);  
    }  
}
```

- Відсортувати студентів за оцінкою в порядку спадання:

```
filteredStudents.sort((s1, s2) -> Integer.compare(s2.getGrade(),  
s1.getGrade()));  
  
System.out.println("Студенти з оцінкою > 80 (відсортовано): " +  
filteredStudents);
```

- Отримати список лише імен студентів:

```
List<String> studentNames = new ArrayList<>();  
for (Student student : filteredStudents) {  
    studentNames.add(student.getName());  
}  
  
System.out.println("Імена студентів з оцінкою > 80: " + studentNames);
```

- Визначити середню оцінку отриманих студентів:

```
int sum = 0;  
for (Student student : students) {  
    sum += student.getGrade();  
}  
double average = (double) sum / students.size();  
System.out.println("Середня оцінка всіх студентів: " + average);
```

4. Зробити всі вище зазначені дії із використанням Stream API

```
List<String> filteredAndSortedNames = students.stream()  
    .filter(student -> student.getGrade() > 80)  
    .sorted(Comparator.comparingInt(Student::getGrade).reversed())  
    .map(Student::getName)  
    .collect(Collectors.toList());  
  
System.out.println("Імена студентів з оцінкою > 80 (відсортовано): " +  
filteredAndSortedNames);
```

```
double average = students.stream()  
    .mapToInt(Student::getGrade)  
    .average()  
    .orElse(0);  
  
System.out.println("Середня оцінка всіх студентів: " + average);
```

ІНДИВІДУАЛЬНІ ЗАВДАННЯ

В даній практичній роботі студентам пропонується виконати 2 завдання відповідно до варіанту ДВОМА СПОСОБАМИ (1 – через використання лямбда виразів та звичайних колекцій; 2 – через StreamAPI). В програмі передбачено 10 варіантів. Якщо ваш варіант перевищує 10, значить робите варіант №варіанту-10. (для 12 варіанту – 2, 17 – 7...).

Варіант 1.

Завдання 1. Аналіз фінансових транзакцій. Є список об'єктів

Transaction(id, type, amount). Завдання:

- Відфільтрувати транзакції типу "DEPOSIT"
- Відсортувати за сумою у порядку спадання
- Обчислити загальну суму всіх транзакцій
- Отримати список ID транзакцій

Завдання 2. Обробка списку користувачів. Є список *User(name, age, email)*.

Завдання:

- Відфільтрувати користувачів старше 18 років
- Отримати список email-адрес
- Визначити користувача з найменшим віком
- Підрахувати кількість користувачів

Варіант 2.

Завдання 1. Робота з рейтингами фільмів. Є список *Movie(title, genre, rating)*. Завдання:

- Відфільтрувати фільми з рейтингом > 8.0
- Відсортувати фільми за рейтингом у порядку спадання
- Отримати список лише назв фільмів
- Обчислити середній рейтинг всіх фільмів

Завдання 2. Робота з книгами. Є список *Book(title, author, year)*. Завдання:

- Відфільтрувати книги, видані після 2000 року
- Відсортувати книги за роком видання
- Отримати список лише назв книг
- Визначити книгу з найстарішим роком видання

Варіант 3.

Завдання 1. Обробка товарів в магазині. Є список *Product(name, category, price)*. Завдання:

- Відфільтрувати товари дорожчі за 1000 грн
- Відсортувати товари за ціною у порядку спадання

- Отримати список лише назв товарів
- Обчислити середню ціну всіх товарів

Завдання 2. Аналіз оцінок студентів. Є список *Student(name, group, grade)*.

Завдання:

- Відфільтрувати студентів з оцінкою > 80
- Відсортувати за оцінкою у порядку спадання
- Отримати список лише імен студентів
- Обчислити середню оцінку всіх студентів

Варіант 4.

Завдання 1. Аналіз спортивних результатів. Є список *Athlete(name, sport, score)*. Завдання:

- Відфільтрувати спортсменів з результатом > 90
- Відсортувати за результатом у порядку спадання
- Отримати список лише імен спортсменів
- Визначити спортсмена з найкращим результатом

Завдання 2. Робота з журналом замовлень. Є список *Order(id, customer, totalAmount, status)*. Завдання:

- Відфільтрувати виконані замовлення
- Відсортувати замовлення за сумою у порядку спадання
- Отримати список лише ID замовлень
- Обчислити загальну суму всіх замовлень

Варіант 5.

Завдання 1. Аналіз погодних даних. Є список *WeatherData(date, city, temperature)*. Завдання:

- Відфільтрувати записи, де температура $> 30^{\circ}\text{C}$
- Відсортувати дані за температурою у порядку спадання
- Отримати список лише дат, коли було спекотно
- Обчислити середню температуру всіх записів

Завдання 2. Робота з автомобілями. Є список *Car(brand, model, year, price)*.

Завдання:

- Відфільтрувати автомобілі, випущені після 2015 року
- Відсортувати за ціною у порядку спадання
- Отримати список лише назв моделей

- Обчислити середню ціну всіх автомобілів

Варіант 6.

Завдання 1. Обробка продажів у магазині. Є список *Sale(date, product, quantity, price)*. Завдання:

- Відфільтрувати продажі певного товару
- Відсортувати за кількістю проданих одиниць
- Отримати список дат продажів
- Обчислити загальну виручку від всіх продажів

Завдання 2. Робота з музичним плейлистом. Є список *Song(title, artist, duration)*. Завдання:

- Відфільтрувати пісні, довші за 4 хвилини
- Відсортувати за тривалістю у порядку спадання
- Отримати список лише назв пісень
- Обчислити загальну тривалість всіх пісень

Варіант 7.

Завдання 1. Аналіз даних про місто. Є список *Building(name, height, yearBuilt)*. Завдання:

- Відфільтрувати будівлі, побудовані після 2000 року
- Відсортувати за висотою у порядку спадання
- Отримати список лише назв будівель
- Визначити найвищу будівлю

Завдання 2. Облік домашніх тварин. Є список *Pet(name, type, age, owner)*. Завдання:

- Відфільтрувати всіх собак
- Відсортувати за віком у порядку спадання
- Отримати список лише імен тварин
- Визначити найстарішу тварину

Варіант 8.

Завдання 1. Аналіз переглядів відео. Є список *Video(title, category, views)*.

Завдання:

- Відфільтрувати відео з переглядами $> 100\,000$
- Відсортувати за кількістю переглядів у порядку спадання
- Отримати список лише назв відео
- Визначити відео з найбільшою кількістю переглядів

Завдання 2. Аналіз результатів тестування студентів. Є список *TestResult(studentName, subject, score)*. Завдання:

- Відфільтрувати студентів, які склали тест (> 60 балів)

- Відсортувати за балами у порядку спадання
- Отримати список лише імен студентів
- Обчислити середній бал всіх студентів

Варіант 9.

Завдання 1. Управління проєктами. Є список *Project(name, teamSize, deadline, budget)*. Завдання:

- Відфільтрувати проєкти з бюджетом більше 1 000 000 грн
- Відсортувати за дедлайном у порядку зростання
- Отримати список лише назв проєктів
- Обчислити загальний бюджет всіх проєктів

Завдання 2. Аналіз транспортних засобів. Є список *Transport(type, capacity, speed)*. Завдання:

- Відфільтрувати транспортні засоби з місткістю > 50 осіб
- Відсортувати за швидкістю у порядку спадання
- Отримати список лише назв транспорту
- Обчислити середню швидкість всіх транспортних засобів

Варіант 10.

Завдання 1. Аналіз відгуків про товари. Є список *Review(product, rating, comment)*. Завдання:

- Відфільтрувати відгуки з рейтингом 5
- Відсортувати за довжиною коментаря
- Отримати список лише коментарів
- Обчислити середню оцінку всіх відгуків

Завдання 2. Обробка замовлень їжі. Є список *FoodOrder(customer, dish, price)*. Завдання:

- Відфільтрувати замовлення дорожчі за 300 грн
- Відсортувати за ціною у порядку спадання
- Отримати список лише назв страв
- Обчислити загальну суму всіх замовлень