

ЛАБОРАТОРНА РОБОТА № 8

Лямбда вирази. Функціональні інтерфейси. Посилання на методи.

Stream API.

Мета: практика роботи з лямбда виразами, функціональними інтерфейсами; використання посилань на методи та Stream API при розробці програм на Java.

Хід роботи:

Завдання 1. Створити консольний Java проект java_lab_8 з пакетом com.education.ztu.

Завдання 2. Описати власний функціональний інтерфейс Printable з методом void print() та написати лямбда вираз цього інтерфейсу.

Завдання 3. Написати лямбда вирази для вбудованих функціональних інтерфейсів: а) Створити лямбда вираз, який повертає значення true, якщо рядок можна привести до числа, використовуючи функціональний інтерфейс Predicate. Створити вираз лямбда, який перевіряє, що рядок можна привести до числа, використовуючи функціональний інтерфейс Predicate. Написати програму, яка перевіряє, що рядок можна привести до числа, використовуючи метод and() функціонального інтерфейсу Predicate. б) Написати лямбда вираз, який приймає на вхід рядок і виводить на консоль повідомлення "Пара почалася о 8:30", "Пара закінчилася о 9:50". Використовуємо функціональний інтерфейс Consumer і метод за замовчуванням andThen. с) Написати лямбда вираз, який виводить в консоль речення в злітерами у верхньому регістрі. Використовуємо функціональний інтерфейс Supplier. д) Написати лямбда вираз, який приймає на вхід рядок з набором чисел через пробіл та повертає добуток цих чисел. Використовуємо функціональний інтерфейс Function.

Завдання 4. Stream API. • Створити стрім з масиву Product з полями name, brand, price, count. • Отримати всі бренди та вивести в консоль. (map) • Отримати 2 товари ціна яких менше тисячі. (filter, limit)

Змн.	Арк.	№ докум.	Підпис	Дата	ДУ «Житомирська політехніка».25.121.00.000 – Пр8		
Розроб.	Iцук Ол.С.						
Перевір.	Піонтківський В.І.						
Керівник							
Н. контр.							
Зав. каф.							
Звіт з лабораторної роботи					Літ.	Арк.	Аркушів
						1	13
					ФІКТ Гр. ІПЗ-23-1		

- Отримати суму всіх видів товарів, що є на складі. (reduce)
- Згрупувати товари по бренду (Collectors.groupingBy())
- Відсортувати товари за зростанням ціни та повернути масив (sorted, Collectors)
- За бажанням дописати функціонал, що використовує інші методи стрімів.

Завдання 5. Посилання на методи чи конструктори. В попередньому завданні, де це можливо, виклики переробити на посилання на методи чи конструктори

Завдання 6. Використання Optional та його методів. Знайти максимальне значення з масиву чисел, в іншому випадку повернути рядок «Числа відсутні».

Лістинг програми:

Main.java:

```
package com.education.ztu;

import java.util.Arrays;
import java.util.Comparator;
import java.util.List;
import java.util.Optional;
import java.util.function.Consumer;
import java.util.function.Function;
import java.util.function.Predicate;
import java.util.function.Supplier;
import java.util.stream.Collectors;
import java.util.stream.Stream;
import java.util.Map;

public class Main {
    public static void main(String[] args) {
        System.out.println("--- Завдання 2: Printable ---");
        task2();

        System.out.println("\n--- Завдання 3: Вбудовані ФІ ---");
        task3();

        System.out.println("\n--- Завдання 4 & 5: Stream API та Посилання ---");
        task4_5();

        System.out.println("\n--- Завдання 6: Optional ---");
        task6();
    }

    public static void task2() {
        Printable myPrintLambda = () -> System.out.println("Привіт! Лямбда-вираз для Printable.");
        myPrintLambda.print();
    }

    public static void task3() {
        Predicate<String> isNumeric = s -> {
            try {
                Double.parseDouble(s);
                return true;
            } catch (NumberFormatException e) {
                return false;
            }
        };
    }
}
```

		Iлуць Ол.С.		
		Плюнгіківський В.І.		
Змн.	Арк.	№ докум.	Підпис	Дата

```

        }

    System.out.println("a) Predicate:");
    System.out.println("'123' є числом? " + isNumeric.test("123"));

    Predicate<String> isNotEmpty = s -> s != null && !s.trim().isEmpty();
    Predicate<String> isValidNumberString = isNotEmpty.and(isNumeric);
    System.out.println(" '789' є числом і не порожній? " +
isValidNumberString.test("789"));

    System.out.println("-----");

    Consumer<String> startMessage = s -> System.out.println(s + "Пара почалася
о 8:30");
    Consumer<String> endMessage = s -> System.out.println(s + "Пара закінчила-
ся о 9:50");
    Consumer<String> fullLecture = startMessage.andThen(endMessage);

    System.out.println("b) Consumer та andThen():");
    fullLecture.accept("");

    System.out.println("-----");

    Supplier<String> uppercaseSentenceSupplier = () -> "Це речення у верхньому
перистрі.";
    String result = uppercaseSentenceSupplier.get().toUpperCase();

    System.out.println("c) Supplier:");
    System.out.println(" Речення: " + result);

    System.out.println("-----");

    Function<String, Long> productOfNumbers = s -> {
        try {
            return Arrays.stream(s.split("\\s+"))
                .map(String::trim)
                .filter(str -> !str.isEmpty())
                .mapToLong(Long::parseLong)
                .reduce(1L, (a, b) -> a * b);
        } catch (NumberFormatException e) {
            System.err.println("Помилка: Не всі елементи рядка є числами.");
            return 0L;
        }
    };
}

String numbersString = "5 10 2 3";
long product = productOfNumbers.apply(numbersString);

System.out.println("d) Function:");
System.out.println(" Добуток: " + product);
}
}

public static void task4_5() {
    Product[] productsArray = new Product[]{
        new Product("Ноутбук Pro", "Apple", 25000.0, 5),
        new Product("Смартфон X", "Samsung", 15000.0, 10),
        new Product("Клавіатура Механічна", "Logitech", 999.50, 20),
        new Product("Монітор 4K", "Samsung", 12000.0, 3),
        new Product("Веб-камера HD", "Logitech", 450.0, 50),
        new Product("Навушники Беспровідні", "Sony", 1500.0, 15)
    };
}

```

		Ilyuk Ol.C.			Арк.
		Піонірківський В.І.			
Змн.	Арк.	№ докум.	Підпис	Дата	3

```

Stream<Product> productStream = Arrays.stream(productsArray);

System.out.println("\n--- 1. Усі бренди (map, Посилання на метод) ---");
List<String> brands = productStream
    .map(Product::getBrand)
    .distinct()
    .collect(Collectors.toList());
System.out.println(" Бренди: " + brands);

productStream = Arrays.stream(productsArray);

System.out.println("\n--- 2. Два товари ціною < 1000 (filter, limit) ---");
List<Product> cheapProducts = productStream
    .filter(p -> p.getPrice() < 1000.0)
    .limit(2)
    .collect(Collectors.toList());
System.out.println(" Товари: " +
cheapProducts.stream().map(Product::getName).collect(Collectors.joining(", ")));

productStream = Arrays.stream(productsArray);

System.out.println("\n--- 3. Сума всіх кількостей товарів (map, reduce) ---");
Optional<Integer> totalCount = productStream
    .map(Product::getCount)
    .reduce(Integer::sum);

System.out.println(" Загальна кількість: " + totalCount.orElse(0));

productStream = Arrays.stream(productsArray);

System.out.println("\n--- 4. Групування товарів по бренду (groupingBy) ---");
Map<String, List<Product>> productsByBrand = productStream
    .collect(Collectors.groupingBy(Product::getBrand));

productsByBrand.forEach((brand, list) -> {
    System.out.println(" Бренд " + brand + ": " + list.size() + " товари(iв)");
});

productStream = Arrays.stream(productsArray);

System.out.println("\n--- 5. Сортування за ціною (sorted, Посилання на метод) ---");
Product[] sortedProducts = productStream
    .sorted(Comparator.comparing(Product::getPrice))
    .toArray(Product[]::new);

System.out.println(" Відсортовані товари:");
for (Product p : sortedProducts) {
    System.out.printf(" - %s: %.2f\n", p.getName(), p.getPrice());
}

System.out.println("\n--- 6. Додатково: Об'єднання назв товарів з кількістю > 10 ---");
String namesWithHighCount = Arrays.stream(productsArray)
    .filter(p -> p.getCount() > 10)
    .map(Product::getName)
    .collect(Collectors.joining(", "));
System.out.println(" Товари: " + namesWithHighCount);
}

```

		Ilyuk Ol.C.		
		Піоніївський В.І.		
Змн.	Арк.	№ докум.	Підпис	Дата

```

public static void task6() {
    int[] numbers1 = {10, 5, 20, 15, 30};
    int[] numbers2 = {};

    String max1 = findMax(numbers1);
    System.out.println(" Масив 1: " + Arrays.toString(numbers1) + " -> Результат: " + max1);

    String max2 = findMax(numbers2);
    System.out.println(" Масив 2: " + Arrays.toString(numbers2) + " -> Результат: " + max2);
}

public static String findMax(int[] array) {
    return Arrays.stream(array)
        .boxed()
        .max(Comparator.naturalOrder())
        .map(Object::toString)
        .orElse("Числа відсутні");
}
}

```

Printable.java:

```

package com.education.ztu;

@FunctionalInterface
public interface Printable {
    void print();
}

```

Product.java:

```

package com.education.ztu;

public class Product {
    private String name;
    private String brand;
    private double price;
    private int count;

    public Product(String name, String brand, double price, int count) {
        this.name = name;
        this.brand = brand;
        this.price = price;
        this.count = count;
    }

    public String getName() { return name; }
    public String getBrand() { return brand; }
    public double getPrice() { return price; }
    public int getCount() { return count; }

    @Override
    public String toString() {
        return "Product{" +
            "name='" + name + '\'' +
            ", brand='" + brand + '\'' +
            ", price=" + price +
            ", count=" + count +
            '}';
    }
}

```

Результат програми:

		Iлуц Ол.С.			Арк.
		Плюнгівський В.І.			
Змн.	Арк.	№ докум.	Підпис	Дата	ДУ «Житомирська політехніка».25.121.00.000 – Пр8

```
--- Завдання 2: Printable ---
Привіт! Лямбда-вираз для Printable.

--- Завдання 3: Вбудовані ФІ ---
a) Predicate:
'123' є числом? true
'789' є числом І не порожній? true

-----
b) Consumer та andThen():
Пара почалася о 8:30
Пара закінчилася о 9:50

-----
c) Supplier:
Речення: ЦЕ РЕЧЕННЯ У ВЕРХНЬОМУ РЕГІСТРІ.

-----
d) Function:
Добуток: 300
```

Рис. 1 Завдання 2 та 3

```
--- 1. Усі бренди (map, Посилання на метод) ---
Бренды: [Apple, Samsung, Logitech, Sony]

--- 2. Два товари ціною < 1000 (filter, limit) ---
Товари: Клавіатура Механічна, Веб-камера HD

--- 3. Сума всіх кількостей товарів (map, reduce) ---
Загальна кількість: 103

--- 4. Групування товарів по бренду (groupingBy) ---
Бренд Sony: 1 товари(iv)
Бренд Apple: 1 товари(iv)
Бренд Logitech: 2 товари(iv)
Бренд Samsung: 2 товари(iv)

--- 5. Сортування за ціною (sorted, Посилання на метод) ---
Відсортовані товари:
- Веб-камера HD: 450,00
- Клавіатура Механічна: 999,50
- Навушники Бездротові: 1500,00
- Монітор 4K: 12000,00
- Смартфон X: 15000,00
- Ноутбук Pro: 25000,00

--- 6. Додатково: Об'єднання назв товарів з кількістю > 10 ---
Товари: Клавіатура Механічна, Веб-камера HD, Навушники Бездротові
```

Рис. 2 Завдання 4 та 5

		<i>Іщук Ол.С.</i>				
		<i>Піонтківський В.І.</i>				
Змн.	Арк.	№ докум.	<i>Підпис</i>	<i>Дата</i>	ДУ «Житомирська політехніка».25.121.00.000 – Пр8	Арк. 6

```
--- Завдання 6: Optional ---
Масив 1: [10, 5, 20, 15, 30] -> Результат: 30
Масив 2: [] -> Результат: Числа відсутні

Process finished with exit code 0
```

Рис. 3 Завдання 6

Посилання на репозиторій: <https://github.com/Sasha1845/Java>

Висновок: Я практикував роботи з лямбда виразами, функціональними інтерфейсами; використання посилань на методи та Stream API при розробці програм на Java.

		Ilyuk Ol.C.			ДУ «Житомирська політехніка».25.121.00.000 – Пр8	Арк.
		Люшко В.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		7