

Лабораторна робота №8

Лямбда вирази. Функціональні інтерфейси. Посилання на методи. Stream API.

Мета роботи: 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.

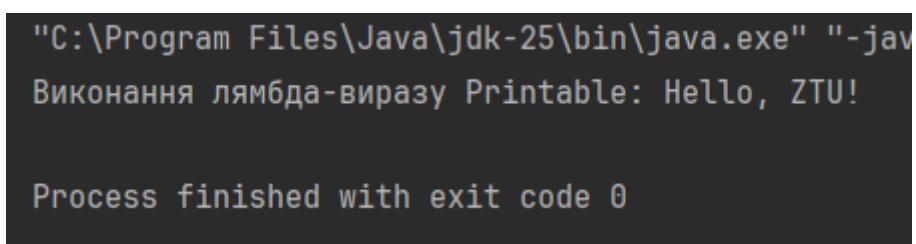
Завдання 2. Описати власний функціональний інтерфейс Printable з методом void print() та написати лямбда вираз цього інтерфейсу.

Лістинг програми:

```
package com.education.ztu;

public class Task_2 {
    public static void main(String[] args) {
        String message = "Виконання лямбда-виразу Printable: Hello, ZTU!";
        Printable printable = () -> System.out.println(message);
        printable.print();
    }
}
```

Результат виконання:



```
"C:\Program Files\Java\jdk-25\bin\java.exe" "-jav
Виконання лямбда-виразу Printable: Hello, ZTU!
Process finished with exit code 0
```

Рис. 1 Результат виконання завдання №2

Завдання 3. Написати лямбда вирази для будованих 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,

Змн.	Арк.	№ докум.	Підпис	Дата	ДУ «Житомирська політехніка».25.121.00.000–Пр8		
Розроб.	Iлуць О.С.						
Перевір.	Піонтківський В.І.						
Керівник							
Н. контр.							
Зав. каф.							
Звіт з лабораторної роботи					Літ.	Арк.	Аркушів
					1	6	
					ФІКТ Гр. ІПЗ-23-1		

використовуючи функціональний інтерфейс Predicate. Написати програму, яка перевіряє, що рядок можна привести до числа, використовуючи метод and() функціонального інтерфейсу Predicate. b) 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. c) Написати лямбда вираз, який виводить в консоль речення в з літерами у верхньому регістрі. Використовуємо функціональний інтерфейс Supplier. d) Написати лямбда вираз, який приймає на вхід рядок з набором чисел через пробіл та повертає добуток цих чисел. Використовуємо функціональний інтерфейс Function.

Лістинг програми:

```
package com.education.ztu;

import java.util.function.Predicate;
import java.util.function.Consumer;
import java.util.function.Supplier;
import java.util.function.Function;

public class Task_3 {
    public static void main(String[] args) {
        // a) Predicate для перевірки, чи рядок можна перетворити на число
        Predicate<String> isNumber = s -> {
            try {
                Double.parseDouble(s);
                return true;
            } catch (NumberFormatException e) {
                return false;
            }
        };

        Predicate<String> notEmpty = s -> !s.isEmpty();
        String testString = "123.45";
        System.out.println("a) Рядок \"" + testString + "\" можна привести до числа: "
                + isNumber.and(notEmpty).test(testString));

        // b) Consumer для виводу повідомлень про початок та кінець пари
        Consumer<String> startLesson = s -> System.out.println("Пара почалася о
8:30");
        Consumer<String> endLesson = s -> System.out.println("Пара закінчилася о
9:50");
        Consumer<String> lesson = startLesson.andThen(endLesson);
        lesson.accept("");

        // c) Supplier для виведення речення з літерами у верхньому регістрі
        Supplier<String> upperCaseSupplier = () -> "це речення у верхньому регіст-
pi".toUpperCase();
        System.out.println(upperCaseSupplier.get());

        // d) Function для обчислення добутку чисел з рядка
        Function<String, Integer> multiplyNumbers = s -> {
```

		Iлуць О.С.			ДУ «Житомирська політехніка».25.121.00.000 – Пр8	Арк.
		Плюнгіківський В.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

```

        String[] parts = s.split("\\s+");
        int result = 1;
        for (String part : parts) {
            result *= Integer.parseInt(part);
        }
        return result;
    };
    String numbers = "2 3 4";
    System.out.println("Добуток чисел \"" + numbers + "\": " +
multiplyNumbers.apply(numbers));
}
}

```

Результат виконання:

```

а) Рядок "123.45" можна привести до числа: true
Пара почалася о 8:30
Пара закінчилася о 9:50
ЦЕ РЕЧЕННЯ У ВЕРХНЬОМУ РЕГІСТРІ
Добуток чисел "2 3 4": 24

```

Рис. 2 Результат виконання завдання №3

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) • 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 додати функціонал, що використовує інші методи стрімів.

Лістинг програми:

```

package com.education.ztu;

import java.util.*;
import java.util.stream.*;
import java.util.stream.Collectors;

public class Task_4 {
    public static void main(String[] args) {
        Product[] products = ProductData.getProducts();

        System.out.println("1) Всі бренди:");
        Arrays.stream(products)
            .map(Product::getBrand)
            .distinct()
            .forEach(System.out::println);

        System.out.println("\n2) 2 товари з ціною менше 1000:");
        Arrays.stream(products)
            .filter(p -> p.getPrice() < 1000)
            .limit(2)
            .forEach(System.out::println);
    }
}

```

		Iлуцк О.С.			ДУ «Житомирська політехніка».25.121.00.000 – Пр8	Арк.
		Піонітківський В.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		3

```

        int totalCount = Arrays.stream(products)
            .map(Product::getCount)
            .reduce(0, Integer::sum);
        System.out.println("\n3) Сума всіх товарів на складі: " + totalCount);

        System.out.println("\n4) Товари згруповані по бренду:");
        Map<String, List<Product>> groupedByBrand = Arrays.stream(products)
            .collect(Collectors.groupingBy(Product::getBrand));
        groupedByBrand.forEach((brand, list) -> System.out.println(brand + ": " +
list));

        System.out.println("\n5) Товари відсортовані за ціною:");
        Product[] sortedByPrice = Arrays.stream(products)
            .sorted(Comparator.comparingDouble(Product::getPrice))
            .toArray(Product[]::new);
        Arrays.stream(sortedByPrice).forEach(System.out::println);

        System.out.println("\n6) Товари з кількістю більше 10:");
        Arrays.stream(products)
            .filter(p -> p.getCount() > 10)
            .map(Product::getName)
            .forEach(System.out::println);
    }
}

```

Результат виконання:

```

1) Всі бренди:
Lenovo
Samsung
LG
Logitech
Apple
Sony

2) 2 товари з ціною менше 1000:
Смартфон | Samsung | 900.0 | 10
Монітор | LG | 800.0 | 3

3) Сума всіх товарів на складі: 80

4) Товари згруповані по бренду:
Sony: [Навушники | Sony | 200.0 | 15]
Lenovo: [Ноутбук | Lenovo | 1200.0 | 5]
Apple: [Планшет | Apple | 1500.0 | 2]
Logitech: [Клавіатура | Logitech | 100.0 | 20, Миша | Logitech | 50.0 | 25]
LG: [Монітор | LG | 800.0 | 3]
Samsung: [Смартфон | Samsung | 900.0 | 10]

5) Товари відсортовані за ціною:
Миша | Logitech | 50.0 | 25
Клавіатура | Logitech | 100.0 | 20

```

Рис. 3 Результат виконання завдання №4

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

Лістинг програми:

```

package com.education.ztu;

import java.util.*;
import java.util.stream.*;

```

		Iлуць О.С.			ДУ «Житомирська політехніка».25.121.00.000 – Пр8	Арк.
		Піонітківський В.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

```

import java.util.stream.Collectors;

public class Task_5 {
    public static void main(String[] args) {
        Product[] products = ProductData.getProducts();

        System.out.println("1) Всі бренди:");
        Arrays.stream(products)
            .map(Product::getBrand)
            .distinct()
            .forEach(System.out::println);

        System.out.println("\n2) 2 товари з ціною менше 1000:");
        Arrays.stream(products)
            .filter(p -> p.getPrice() < 1000)
            .limit(2)
            .forEach(System.out::println);

        int totalCount = Arrays.stream(products)
            .map(Product::getCount)
            .reduce(0, Integer::sum);
        System.out.println("\n3) Сума всіх товарів на складі: " + totalCount);

        System.out.println("\n4) Товари згруповані по бренду:");
        Map<String, List<Product>> groupedByBrand = Arrays.stream(products)
            .collect(Collectors.groupingBy(Product::getBrand));
        groupedByBrand.forEach((brand, list) -> System.out.println(brand + ": " +
list));

        System.out.println("\n5) Товари відсортовані за ціною:");
        Product[] sortedByPrice = Arrays.stream(products)
            .sorted(Comparator.comparingDouble(Product::getPrice))
            .toArray(Product[]::new);
        Arrays.stream(sortedByPrice).forEach(System.out::println);

        System.out.println("\n6) Товари з кількістю більше 10:");
        Arrays.stream(products)
            .filter(p -> p.getCount() > 10)
            .map(Product::getName)
            .forEach(System.out::println);
    }
}

```

Результат виконання:

		Iлуцк О.С.			ДУ «Житомирська політехніка».25.121.00.000 – Пр8	Арк.
		Піоніївський В.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		5

```

1) Всі бренди:
Lenovo
Samsung
LG
Logitech
Apple
Sony

2) 2 товари з ціною менше 1000:
Смартфон | Samsung | 900.0 | 10
Монітор | LG | 800.0 | 3

3) Сума всіх товарів на складі: 80

4) Товари згруповані по бренду:
Sony: [Навушники | Sony | 200.0 | 15]
Lenovo: [Ноутбук | Lenovo | 1200.0 | 5]
Apple: [Планшет | Apple | 1500.0 | 2]
Logitech: [Клавіатура | Logitech | 100.0 | 20, Миша | Logitech | 50.0 | 25]
LG: [Монітор | LG | 800.0 | 3]
Samsung: [Смартфон | Samsung | 900.0 | 10]

5) Товари відсортовані за ціною:
Миша | Logitech | 50.0 | 25
Клавіатура | Logitech | 100.0 | 20

```

Рис. 4 Результат виконання завдання №5

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

Лістинг програми:

```

package com.education.ztu;

import java.util.*;
import java.util.stream.*;

public class Task_6 {
    public static void main(String[] args) {
        Integer[] numbers = {5, 12, 7, 3, 18, 9};

        Optional<Integer> max = Arrays.stream(numbers)
            .max(Integer::compareTo);

        String result = max.map(String::valueOf)
            .orElse("Числа відсутні");

        System.out.println("Максимальне значення: " + result);

        // З порожнім масивом
        Integer[] emptyArray = {};
        Optional<Integer> maxEmpty = Arrays.stream(emptyArray)
            .max(Integer::compareTo);
        String resultEmpty = maxEmpty.map(String::valueOf)
            .orElse("Числа відсутні");
        System.out.println("Максимальне значення порожнього масиву: " +
resultEmpty);
    }
}

```

Результат виконання:

		Iлуцк О.С.			ДУ «Житомирська політехніка».25.121.00.000 – Пр8	Арк.
		Плюнчіківський В.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

Максимальне значення: 18

Максимальне значення порожнього масиву: Числа відсутні

Рис. 5 Результат виконання завдання №6

Посилання на репозиторій: <https://github.com/Oleg-Ischuk/Java-IPZ-23-1>

Висновок: попрактикувався працювати з лямбда виразами, функціональними інтерфейсами; навчився використовувати посилання на методи та Stream API при розробці програм на Java.

		Iuцук O.C.			ДУ «Житомирська політехніка».25.121.00.000 – Пр8	Арк.
		Піоніківський В.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		