

## ЛАБОРАТОРНА РОБОТА № 5

# Java Collections Framework

**Мета:** робота з Java Collections Framework

**Хід роботи:**

**Завдання 1.** Створити консольний Java проект `java_lab_5` з пакетом `com.education.ztu`

**Завдання 2.** Створити клас `Product` та задати йому поля та методи на власний вибір.

**Завдання 3.** Створити динамічний масив, що містить об'єкти класу `Product`: • Використовуємо клас `ArrayList` або `LinkedList`. • Продемонструвати роботу з масивом використовуючи різні методи (`add`, `addAll`, `get`, `indexOf`, `lastIndexOf`, `iterator`, `listIterator`, `remove`, `set`, `sort`, `subList`, `clear`, `contains`, `isEmpty`, `retainAll`, `size`, `toArray`)

**Завдання 4.** Створити чергу, що містить об'єкти класу `Product`: • Використовуємо клас `ArrayDeque`. • Продемонструвати роботу з чергою використовуючи методи (`push`, `offerLast`, `getFirst`, `peekLast`, `pop`, `removeLast`, `pollLast` та інші)

**Завдання 5.** Створити множину, що містить об'єкти класу `Product`: • Використовуємо клас `TreeSet`. • Продемонструвати роботу з множиною використовуючи методи (`add`, `first`, `last`, `headSet`, `subSet`, `tailSet`, `ceiling`, `floor`, `higher`, `lower`, `pollFirst`, `pollLast`, `descendingSet`)

**Завдання 5.** Створити `Map` що містить пари (ключ, значення) - ім'я продукту та об'єкт продукту (клас `Product`). • Використовуємо клас `HashMap`, • Продемонструвати роботу з `Map` використовуючи методи (`put`, `get`, `containsKey`, `containsValue`, `clear`, `putIfAbsent`, `keySet`, `values`, `putAll`, `remove`, `size`) • Викликати метод `entrySet` та продемонструвати роботу з набором значень, що він поверне (`getKey`, `getValue`, `setValue`)

					ДУ «Житомирська політехніка».25.121.00.000 – Лр5			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Ищук Ол.С.			Звіт з лабораторної роботи		Літ.	Арк.
Перевір.		Піонтківській В.І.						1
Керівник							Аркушів	
Н. контр.							13	
Зав. каф.							ФІКТ Гр. ІПЗ-23-1	

**Завдання 6.** Продемонструвати роботу з класом Collections: • Для роботи використати масив створений через Arrays.asList • Метод Collections.sort() • Метод Collections.binarySearch() • Методи Collections.reverse(), Collections.shuffle() • Метод Collections.fill() • Методи Collections.max(), Collections.min() • Метод Collections.copy() • Метод Collections.rotate() • Метод Collections.checkedCollection() • Метод Collections.frequency()

Лістинг програми:

**Main.java:**

```
package com.education.ztu;

import java.util.*;

public class Main {
    public static void main(String[] args) {

        // =====
        // Завдання 3: Динамічний масив
        // =====
        System.out.println("== ArrayList ==");
        List<Product> products = new ArrayList<>();
        Product p1 = new Product("Apple", 1.2);
        Product p2 = new Product("Banana", 0.8);
        Product p3 = new Product("Orange", 1.5);

        products.add(p1);
        products.add(p2);
        products.add(p3);

        List<Product> moreProducts = Arrays.asList(new Product("Kiwi", 2.0), new
        Product("Mango", 2.5));
        products.addAll(moreProducts);

        System.out.println("ArrayList: " + products);
        System.out.println("Get element at index 1: " + products.get(1));
        System.out.println("Index of Banana: " + products.indexOf(p2));
        System.out.println("Last index of Banana: " + products.lastIndexOf(p2));

        products.remove(p2);
        System.out.println("After removing Banana: " + products);

        products.set(0, new Product("Pineapple", 3.0));
        System.out.println("After set element at index 0: " + products);

        Collections.sort(products);
        System.out.println("After sort: " + products);

        List<Product> sublist = products.subList(1, 3);
        System.out.println("Sublist(1,3): " + sublist);

        System.out.println("Contains Mango? " + products.contains(new
        Product("Mango", 2.5)));
        System.out.println("Is empty? " + products.isEmpty());
        System.out.println("Size: " + products.size());
    }
}
```

		Іщук Ол.С.			ДУ «Житомирська політехніка».25.121.00.000 – Лр5	Арк.
		Піонтківській В.І.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

Object[] array = products.toArray();
System.out.println("ToArray: " + Arrays.toString(array));

System.out.println("\n=== Iterator ===");
Iterator<Product> iterator = products.iterator();
while(iterator.hasNext()){
    System.out.println(iterator.next());
}

System.out.println("\n=== ListIterator ===");
ListIterator<Product> listIterator = products.listIterator();
while(listIterator.hasNext()){
    System.out.println(listIterator.next());
}

// =====
// Завдання 4: Черга ArrayDeque
// =====
System.out.println("\n=== ArrayDeque ===");
ArrayDeque<Product> queue = new ArrayDeque<>();
queue.push(p1);
queue.offerLast(p2);
queue.offerLast(p3);

System.out.println("Queue: " + queue);
System.out.println("First element: " + queue.getFirst());
System.out.println("Peek last: " + queue.peekLast());
System.out.println("Pop: " + queue.pop());
System.out.println("Remove last: " + queue.removeLast());
System.out.println("Poll last: " + queue.pollLast());
System.out.println("Queue after operations: " + queue);

// =====
// Завдання 5: Множина TreeSet
// =====
System.out.println("\n=== TreeSet ===");
TreeSet<Product> set = new TreeSet<>();
set.add(p1);
set.add(p2);
set.add(p3);

System.out.println("TreeSet: " + set);
System.out.println("First: " + set.first());
System.out.println("Last: " + set.last());
System.out.println("HeadSet(Orange): " + set.headSet(new Product("Orange",
0)));
System.out.println("SubSet(Apple, Orange): " + set.subSet(new
Product("Apple",0), new Product("Orange",0)));
System.out.println("TailSet(Banana): " + set.tailSet(new
Product("Banana",0)));
System.out.println("Ceiling(Apple): " + set.ceiling(new
Product("Apple",0)));
System.out.println("Floor(Mango): " + set.floor(new Product("Mango",0)));
System.out.println("Higher(Apple): " + set.higher(new
Product("Apple",0)));
System.out.println("Lower(Orange): " + set.lower(new
Product("Orange",0)));
System.out.println("Poll first: " + set.pollFirst());
System.out.println("Poll last: " + set.pollLast());
System.out.println("Descending set: " + set.descendingSet());

// =====
// Завдання 6: Map HashMap

```

		Іщук Ол.С.			ДУ «Житомирська політехніка».25.121.00.000 – Лр5	Арк.
		Піонтківській В.І.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```
// =====
System.out.println("\n=== HashMap ===");
Map<String, Product> map = new HashMap<>();
map.put("Apple", p1);
map.put("Banana", p2);
map.put("Orange", p3);

System.out.println("Map: " + map);
System.out.println("Get Apple: " + map.get("Apple"));
System.out.println("Contains key Banana? " + map.containsKey("Banana"));
System.out.println("Contains value p2? " + map.containsValue(p2));
map.putIfAbsent("Kiwi", new Product("Kiwi", 2));
System.out.println("After putIfAbsent: " + map);

Set<Map.Entry<String, Product>> entries = map.entrySet();
for (Map.Entry<String, Product> entry : entries) {
    System.out.println(entry.getKey() + " => " + entry.getValue());
}

// =====
// Завдання 7: Collections
// =====
System.out.println("\n=== Collections ===");
List<Integer> numbers = new ArrayList<>(Arrays.asList(5, 3, 8, 1, 9));
System.out.println("Original list: " + numbers);

Collections.sort(numbers);
System.out.println("Sorted: " + numbers);

System.out.println("Binary search for 3: " +
Collections.binarySearch(numbers, 3));

Collections.reverse(numbers);
System.out.println("Reversed: " + numbers);

Collections.shuffle(numbers);
System.out.println("Shuffled: " + numbers);

Collections.fill(numbers, 0);
System.out.println("Fill with 0: " + numbers);

List<Integer> numbers2 = new ArrayList<>(Arrays.asList(1, 2, 3, 4, 5));
Collections.copy(numbers2, Arrays.asList(9, 9, 9, 9, 9));
System.out.println("After copy: " + numbers2);

Collections.rotate(numbers2, 2);
System.out.println("After rotate by 2: " + numbers2);

Collection<Integer> checked = Collections.checkedCollection(numbers2,
Integer.class);
System.out.println("Checked collection: " + checked);

System.out.println("Max: " + Collections.max(numbers2));
System.out.println("Min: " + Collections.min(numbers2));
System.out.println("Frequency of 9: " + Collections.frequency(numbers2,
9));
}
}
```

## Product.java:

```
package com.education.ztu;

import java.util.Objects;
```

		Іщук Ол.С.			ДУ «Житомирська політехніка».25.121.00.000 – Лр5	Арк.
		Піонтиківській В.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

```

public class Product implements Comparable<Product> {
    private String name;
    private double price;

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

    public String getName() { return name; }
    public void setName(String name) { this.name = name; }

    public double getPrice() { return price; }
    public void setPrice(double price) { this.price = price; }

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

    @Override
    public int compareTo(Product other) {
        return this.name.compareTo(other.name);
    }

    @Override
    public boolean equals(Object o) {
        if(this == o) return true;
        if(o == null || getClass() != o.getClass()) return false;
        Product product = (Product) o;
        return Double.compare(product.price, price) == 0 &&
name.equals(product.name);
    }

    @Override
    public int hashCode() {
        return Objects.hash(name, price);
    }
}

```

## Результат програми:

```

"C:\Program Files\Java\jdk-25\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.2.2\lib\idea_rt.jar=59956" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8
=== ArrayList ===
ArrayList: [Product{name='Apple', price=1.2}, Product{name='Banana', price=0.6}, Product{name='Orange', price=1.5}, Product{name='Kiwi', price=2.0}, Product{name='Mango', price=2.5}]
Get element at index 1: Product{name='Banana', price=0.6}
Index of Banana: 1
Last index of Banana: 1
After removing Banana: [Product{name='Apple', price=1.2}, Product{name='Orange', price=1.5}, Product{name='Kiwi', price=2.0}, Product{name='Mango', price=2.5}]
After set element at index 0: [Product{name='Pineapple', price=3.0}, Product{name='Orange', price=1.5}, Product{name='Kiwi', price=2.0}, Product{name='Mango', price=2.5}]
After sort: [Product{name='Kiwi', price=2.0}, Product{name='Mango', price=2.5}, Product{name='Orange', price=1.5}, Product{name='Pineapple', price=3.0}]
Sublist(1,3): [Product{name='Mango', price=2.5}, Product{name='Orange', price=1.5}]
Contains Mango? true
Is empty? false
Size: 4
ToArray: [Product{name='Kiwi', price=2.0}, Product{name='Mango', price=2.5}, Product{name='Orange', price=1.5}, Product{name='Pineapple', price=3.0}]

=== Iterator ===
Product{name='Kiwi', price=2.0}
Product{name='Mango', price=2.5}
Product{name='Orange', price=1.5}
Product{name='Pineapple', price=3.0}

=== ListIterator ===
Product{name='Kiwi', price=2.0}
Product{name='Mango', price=2.5}
Product{name='Orange', price=1.5}
Product{name='Pineapple', price=3.0}

```

Рис 1. Результат програми

		Іщук Ол.С.			ДУ «Житомирська політехніка».25.121.00.000 – Лр5	Арк.
		Піонтківській В.І.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

=== ArrayDeque ===
Queue: [Product{name='Apple', price=1.2}, Product{name='Banana', price=0.8}, Product{name='Orange', price=1.5}]
First element: Product{name='Apple', price=1.2}
Peak last: Product{name='Orange', price=1.5}
Pop: Product{name='Apple', price=1.2}
Remove last: Product{name='Orange', price=1.5}
Poll last: Product{name='Banana', price=0.8}
Queue after operations: []

=== TreeSet ===
TreeSet: [Product{name='Apple', price=1.2}, Product{name='Banana', price=0.8}, Product{name='Orange', price=1.5}]
First: Product{name='Apple', price=1.2}
Last: Product{name='Orange', price=1.5}
HeadSet(Orange): [Product{name='Apple', price=1.2}, Product{name='Banana', price=0.8}]
SubSet(Apple, Orange): [Product{name='Apple', price=1.2}, Product{name='Banana', price=0.8}]
TailSet(Banana): [Product{name='Banana', price=0.8}, Product{name='Orange', price=1.5}]
Ceiling(Apple): Product{name='Apple', price=1.2}
Floor(Mango): Product{name='Banana', price=0.8}
Higher(Apple): Product{name='Banana', price=0.8}
Lower(Orange): Product{name='Banana', price=0.8}
Poll first: Product{name='Apple', price=1.2}
Poll last: Product{name='Orange', price=1.5}
Descending set: [Product{name='Banana', price=0.8}]

```

**Рис 2. Результат програми**

```

=== HashMap ===
Map: {Apple=Product{name='Apple', price=1.2}, Orange=Product{name='Orange', price=1.5}, Banana=Product{name='Banana', price=0.8}}
Get Apple: Product{name='Apple', price=1.2}
Contains key Banana? true
Contains value #2? true
After putIfAbsent: {Apple=Product{name='Apple', price=1.2}, Kiwi=Product{name='Kiwi', price=2.0}, Orange=Product{name='Orange', price=1.5}, Banana=Product{name='Banana', price=0.8}}
Apple => Product{name='Apple', price=1.2}
Kiwi => Product{name='Kiwi', price=2.0}
Orange => Product{name='Orange', price=1.5}
Banana => Product{name='Banana', price=0.8}

=== Collections ===
Original list: [5, 3, 0, 3, 9]
Sorted: [0, 3, 3, 5, 9]
Binary search for 3: 1
Reversed: [9, 0, 3, 3, 5]
Shuffled: [3, 0, 5, 3, 9]
Fill with 0: [0, 0, 0, 0, 0]
After copy: [9, 9, 9, 9, 9]
After rotate by 2: [9, 9, 9, 9, 9]
Checked collection: [9, 9, 9, 9, 9]
Max: 9
Min: 9
Frequency of 9: 5

```

**Рис 3. Результат програми**

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

**Висновок:** Я навчився працювати з Java Collections Framework

		Іщук Ол.С.			ДУ «Житомирська політехніка».25.121.00.000 – Лр5	Арк.
		Піонтківській В.І.				6
Змн.	Арк.	№ докум.	Підпис	Дата		