

ЛАБОРАТОРНА РОБОТА № 5

Тема: Java Collections Framework

Мета роботи: робота з Java Collections Framework

Завдання на лабораторну роботу

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.

Лістинг програми

```
package com.education.ztu;

public class Product implements Comparable<Product> {
    private String brand;
    private String model;
    private int year;
    private double price;
    public Product(String brand, String model, int year, double price) {
        this.brand = brand;
        this.model = model;
        this.year = year;
        this.price = price;
    }
    public String getBrand() { return brand; }
    public String getModel() { return model; }
    public int getYear() { return year; }
    public double getPrice() { return price; }
    public void setBrand(String brand) { this.brand = brand; }
    public void setModel(String model) { this.model = model; }
    public void setYear(int year) { this.year = year; }
    public void setPrice(double price) { this.price = price; }

    @Override
    public String toString() {
        return "Product{" + brand + " " + model + ", " + year + ", $" + price + "}";
    }
    @Override
    public int compareTo(Product other) {
        return Double.compare(this.price, other.price);
    }
    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (!(o instanceof Product)) return false;
        Product p = (Product) o;
        return brand.equals(p.brand) && model.equals(p.model) && year == p.year && price == p.price;
    }
    @Override
    public int hashCode() {
        return (brand + model + year + price).hashCode();
    }
}
```

| | | | | | | | |
|-------------------------------|-----------------|----------|--------|------|---|------|---------|
| Змн. | Арк. | № докум. | Підпис | Дата | ДУ «Житомирська політехніка».25.121.01.000 – Пр 1 | | |
| Розроб. | Aйсін В.С | | | | Lім. | Арк. | Аркушів |
| Перевір. | Плютківський ВІ | | | | | 1 | 11 |
| Керівник | | | | | | | |
| Н. контр. | | | | | | | |
| Зав. каф. | | | | | | | |
| Звіт з лабораторної роботи | | | | | ФІКТ Гр. ІПЗ-23-1[1] | | |

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)

Лістинг програми

```
package com.education.ztu;

import java.util.*;

public class Main {
    public static void main(String[] args) {
        List<Product> products = new ArrayList<>();
        products.add(new Product("Tesla", "Model S", 2022, 89999));
        products.add(new Product("BMW", "X5", 2020, 60000));
        products.add(new Product("Toyota", "Camry", 2019, 25000));
        List<Product> moreProducts = new ArrayList<>();
        moreProducts.add(new Product("Audi", "A6", 2021, 55000));
        moreProducts.add(new Product("Ford", "Mustang", 2022, 45000));
        products.addAll(moreProducts);
        System.out.println("Перший продукт: " + products.get(0));
        System.out.println("Індекс BMW: " + products.indexOf(products.get(1)));
        System.out.println("\nУсі продукти:");
        Iterator<Product> it = products.iterator();
        while (it.hasNext()) {
            System.out.println(it.next());
        }
        System.out.println("\nУ зворотному порядку:");
        ListIterator<Product> listIt = products.listIterator(products.size());
        while (listIt.hasPrevious()) {
            System.out.println(listIt.previous());
        }
        products.remove(2);
        products.set(1, new Product("Mercedes", "GLE", 2021, 70000));
        products.sort(Comparator.comparingDouble(Product::getPrice));
        System.out.println("\n2 найдешевші продукти:");
        List<Product> top2 = products.subList(0, 2);
        top2.forEach(System.out::println);
        System.out.println("\nМістить Audi A6? " + products.contains(moreProducts.get(0)));
        System.out.println("Список порожній? " + products.isEmpty());
        System.out.println("Кількість продуктів: " + products.size());
        Product[] productArray = products.toArray(new Product[0]);
        System.out.println("\nПродукти у вигляді масиву:");
        for (Product product : productArray) {
            System.out.println(product);
        }
        products.removeAll(moreProducts);
        System.out.println("\nПродукти після removeAll з moreProducts:");
        products.forEach(System.out::println);
        products.clear();
        System.out.println("Список очищено. Порожній? " + products.isEmpty());
    }
}
```

| | | | | | | |
|------|------|------------------|--------|------|--|------|
| | | Aйсін В.С | | | ДУ «Житомирська політехніка».25.121.01.000 – Пр1 | Арк. |
| | | Плюнгівський В І | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | 2 |

```
Перший продукт: Product{Tesla Model S, 2022, $89999.0}
```

```
Індекс BMW: 1
```

```
Усі продукти:
```

```
Product{Tesla Model S, 2022, $89999.0}
```

```
Product{BMW X5, 2020, $60000.0}
```

```
Product{Toyota Camry, 2019, $25000.0}
```

```
Product{Audi A6, 2021, $55000.0}
```

```
Product{Ford Mustang, 2022, $45000.0}
```

```
У зворотному порядку:
```

```
Product{Ford Mustang, 2022, $45000.0}
```

```
Product{Audi A6, 2021, $55000.0}
```

```
Product{Toyota Camry, 2019, $25000.0}
```

```
Product{BMW X5, 2020, $60000.0}
```

```
Product{Tesla Model S, 2022, $89999.0}
```

```
2 найдешевші продукти:
```

```
Product{Ford Mustang, 2022, $45000.0}
```

```
Product{Audi A6, 2021, $55000.0}
```

```
Містить Audi A6? true
```

```
Список порожній? false
```

```
Кількість продуктів: 4
```

```
Продукти у вигляді масиву:
```

```
Product{Ford Mustang, 2022, $45000.0}
```

```
Product{Audi A6, 2021, $55000.0}
```

```
Product{Mercedes GLE, 2021, $70000.0}
```

```
Product{Tesla Model S, 2022, $89999.0}
```

```
Продукти після retainAll з moreProducts:
```

```
Product{Ford Mustang, 2022, $45000.0}
```

```
Product{Audi A6, 2021, $55000.0}
```

```
Список очищено. Порожній? true
```

| | | | | | | |
|------|------|-------------------|--------|--------|--|------|
| | | Aйсін В.С | | | ДУ «Житомирська політехніка».25.121.01.000 – Пр1 | Арк. |
| Змн. | Арк. | Ліонтківський В І | | Підпис | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | 3 |

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)

Лістинг програми

```
package com.education.ztu;

import java.util.ArrayDeque;
import java.util.Deque;

public class Main2 {
    public static void main(String[] args) {
        Deque<Product> productQueue = new ArrayDeque<>();
        productQueue.push(new Product("Tesla", "Cybertruck", 2024, 79999));
        productQueue.offerLast(new Product("BMW", "i8", 2021, 99999));
        productQueue.offerLast(new Product("Audi", "Q7", 2022, 65000));
        productQueue.push(new Product("Ford", "F-150", 2023, 48000));
        System.out.println("Черга після додавання:");
        productQueue.forEach(System.out::println);
        System.out.println("\nПерший елемент (getFirst): " + productQueue.getFirst());
        System.out.println("Останній елемент (peekLast): " + productQueue.peekLast());
        System.out.println("\nВидаляємо перший елемент (pop): " + productQueue.pop());
        System.out.println("Видаляємо останній елемент (removeLast): " + productQueue.removeLast());
        System.out.println("Видаляємо останній елемент (pollLast): " + productQueue.pollLast());
        System.out.println("\nЧерга після видалення:");
        productQueue.forEach(System.out::println);
        System.out.println("\nРозмір черги: " + productQueue.size());
        System.out.println("Черга порожня? " + productQueue.isEmpty());
        productQueue.clear();
        System.out.println("Після очищення, черга порожня? " + productQueue.isEmpty());
    }
}
```

Черга після додавання:

```
Product{Ford F-150, 2023, $48000.0}
Product{Tesla Cybertruck, 2024, $79999.0}
Product{BMW i8, 2021, $99999.0}
Product{Audi Q7, 2022, $65000.0}
```

Перший елемент (getFirst): Product{Ford F-150, 2023, \$48000.0}

Останній елемент (peekLast): Product{Audi Q7, 2022, \$65000.0}

Видаляємо перший елемент (pop): Product{Ford F-150, 2023, \$48000.0}

Видаляємо останній елемент (removeLast): Product{Audi Q7, 2022, \$65000.0}

Видаляємо останній елемент (pollLast): Product{BMW i8, 2021, \$99999.0}

Черга після видалення:

```
Product{Tesla Cybertruck, 2024, $79999.0}
```

Розмір черги: 1

Черга порожня? false

Після очищення, черга порожня? true

| | | | | | | |
|------|------|-------------------|--------|------|--|------|
| | | Aйсін В.С | | | ДУ «Житомирська політехніка».25.121.01.000 – Пр1 | Арк. |
| | | Лютніковський В І | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | 4 |

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)

Лістинг програми

```
package com.education.ztu;

import java.util.*;

public class Main3 {
    public static void main(String[] args) {
        TreeSet<Product> productSet = new TreeSet<>();

        productSet.add(new Product("Tesla", "Model Y", 2023, 52000));
        productSet.add(new Product("BMW", "X5", 2020, 60000));
        productSet.add(new Product("Toyota", "Corolla", 2019, 20000));
        productSet.add(new Product("Audi", "Q5", 2021, 45000));
        productSet.add(new Product("Ford", "Focus", 2018, 18000));

        System.out.println("Всі продукти у множині (за зростанням ціни):");
        productSet.forEach(System.out::println);

        System.out.println("\nПерший (найдешевший): " + productSet.first());
        System.out.println("Останній (найдорожчий): " + productSet.last());

        Product refProduct = new Product("Ref", "Ref", 2020, 45000);

        System.out.println("\nПродукти дешевші за $45000 (headSet):");
        productSet.headSet(refProduct).forEach(System.out::println);

        System.out.println("\nПродукти від $45000 включно (tailSet):");
        productSet.tailSet(refProduct).forEach(System.out::println);

        System.out.println("\nПродукти у діапазоні $20000 - $52000 (subSet):");
        Product low = new Product("L", "L", 0, 20000);
        Product high = new Product("H", "H", 0, 52000);
        productSet.subSet(low, high).forEach(System.out::println);

        System.out.println("\nЕлемент рівний або більший за $45000 (ceiling): " +
productSet.ceiling(refProduct));
        System.out.println("Елемент рівний або менший за $45000 (floor): " + productSet.floor(refProduct));
        System.out.println("Наступний після $45000 (higher): " + productSet.higher(refProduct));
        System.out.println("Попередній перед $45000 (lower): " + productSet.lower(refProduct));

        System.out.println("\nВидаляємо найменший (pollFirst): " + productSet.pollFirst());
        System.out.println("Видаляємо найбільший (pollLast): " + productSet.pollLast());

        System.out.println("\nЗалишилось у множині:");
        productSet.forEach(System.out::println);

        System.out.println("\nУ зворотному порядку:");
        NavigableSet<Product> descending = productSet.descendingSet();
        descending.forEach(System.out::println);
    }
}
```

| | | | | | | |
|------|------|-------------------|--------|------|--|------|
| | | Aїсін В.С | | | ДУ «Житомирська політехніка».25.121.01.000 – Пр1 | Арк. |
| | | Ліонтківський В І | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | 5 |

Всі продукти у множині (за зростанням ціни):

Product{Ford Focus, 2018, \$18000.0}

Product{Toyota Corolla, 2019, \$20000.0}

Product{Audi Q5, 2021, \$45000.0}

Product{Tesla Model Y, 2023, \$52000.0}

Product{BMW X5, 2020, \$60000.0}

Перший (найдешевший): Product{Ford Focus, 2018, \$18000.0}

Останній (найдорожчий): Product{BMW X5, 2020, \$60000.0}

Продукти дешевші за \$45000 (headSet):

Product{Ford Focus, 2018, \$18000.0}

Product{Toyota Corolla, 2019, \$20000.0}

Продукти від \$45000 включно (tailSet):

Product{Audi Q5, 2021, \$45000.0}

Product{Tesla Model Y, 2023, \$52000.0}

Product{BMW X5, 2020, \$60000.0}

Продукти у діапазоні \$20000 - \$52000 (subSet):

Product{Toyota Corolla, 2019, \$20000.0}

Product{Audi Q5, 2021, \$45000.0}

Елемент рівний або більший за \$45000 (ceiling): Product{Audi Q5, 2021, \$45000.0}

Елемент рівний або менший за \$45000 (floor): Product{Audi Q5, 2021, \$45000.0}

Наступний після \$45000 (higher): Product{Tesla Model Y, 2023, \$52000.0}

Попередній перед \$45000 (lower): Product{Toyota Corolla, 2019, \$20000.0}

Видаляємо найменший (pollFirst): Product{Ford Focus, 2018, \$18000.0}

Видаляємо найбільший (pollLast): Product{BMW X5, 2020, \$60000.0}

Залишилось у множині:

Product{Toyota Corolla, 2019, \$20000.0}

Product{Audi Q5, 2021, \$45000.0}

Product{Tesla Model Y, 2023, \$52000.0}

У зворотному порядку:

Product{Tesla Model Y, 2023, \$52000.0}

Product{Audi Q5, 2021, \$45000.0}

Product{Toyota Corolla, 2019, \$20000.0}

| | | | | | | |
|------|------|------------------|--------|------|--|------|
| | | Aйсін В.С | | | ДУ «Житомирська політехніка».25.121.01.000 – Пр1 | Арк. |
| | | Любомирський В.І | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | 6 |

Завдання 5. Створити Мар що містить пари (ключ, значення) - ім'я продукту та об'єкт продукту (клас Product).

- Використовуємо клас HashMap,
- Продемонструвати роботу з Мар використовуючи методи (put, get, 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)

Лістинг програми

```
package com.education.ztu;

import java.util.*;

public class Main4 {
    public static void main(String[] args) {
        Map<String, Product> productMap = new HashMap<>();
        productMap.put("Tesla Model S", new Product("Tesla", "Model S", 2022, 89999));
        productMap.put("BMW X5", new Product("BMW", "X5", 2020, 60000));
        productMap.put("Toyota Camry", new Product("Toyota", "Camry", 2019, 25000));
        productMap.putIfAbsent("BMW X5", new Product("BMW", "X5", 2021, 62000));
        productMap.putIfAbsent("Audi A6", new Product("Audi", "A6", 2021, 55000));
        System.out.println("Отримати Tesla Model S: " + productMap.get("Tesla Model S"));
        System.out.println("Отримати Audi A6: " + productMap.get("Audi A6"));
        System.out.println("\nМістить ключ 'BMW X5'? " + productMap.containsKey("BMW X5"));
        System.out.println("Містить значення Audi A6? " + productMap.containsValue(new Product("Audi", "A6",
2021, 55000)));
        System.out.println("\nУсі ключі (keySet):");
        for (String key : productMap.keySet()) {
            System.out.println(key);
        }
        System.out.println("\nУсі значення (values):");
        for (Product product : productMap.values()) {
            System.out.println(product);
        }
        System.out.println("\nПари ключ-значення (entrySet):");
        for (Map.Entry<String, Product> entry : productMap.entrySet()) {
            System.out.println("Ключ: " + entry.getKey());
            System.out.println("Значення: " + entry.getValue());
            if (entry.getKey().equals("Toyota Camry")) {
                entry.setValue(new Product("Toyota", "Camry", 2019, 23000)); // змінили ціну
            }
        }
        System.out.println("\nПісля зміни ціни Toyota Camry:");
        productMap.forEach((key, value) -> System.out.println(key + " -> " + value));
        productMap.remove("BMW X5");
        Map<String, Product> newProducts = new HashMap<>();
        newProducts.put("Ford Mustang", new Product("Ford", "Mustang", 2022, 45000));
        productMap.putAll(newProducts);
        System.out.println("\nПісля видалення BMW та додавання Mustang:");
        productMap.forEach((key, value) -> System.out.println(key + " -> " + value));
        System.out.println("\nКількість елементів у Мар: " + productMap.size());
        productMap.clear();
        System.out.println("Мар очищено. Порожня? " + productMap.isEmpty());
    }
}
```

| | | | | | | |
|------|------|------------------|--------|------|--|------|
| | | Aйсін В.С | | | ДУ «Житомирська політехніка».25.121.01.000 – Пр1 | Арк. |
| | | Любомирський В.І | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | 7 |

```
Отримати Tesla Model S: Product{Tesla Model S, 2022, $89999.0}
Отримати Audi A6: Product{Audi A6, 2021, $55000.0}
```

Містить ключ 'BMW X5'? true

Містить значення Audi A6? true

Усі ключі (keySet):

Toyota Camry

Audi A6

BMW X5

Tesla Model S

Усі значення (values):

```
Product{Toyota Camry, 2019, $25000.0}
```

```
Product{Audi A6, 2021, $55000.0}
```

```
Product{BMW X5, 2020, $60000.0}
```

```
Product{Tesla Model S, 2022, $89999.0}
```

Пари ключ-значення (entrySet):

Ключ: Toyota Camry

Значення: Product{Toyota Camry, 2019, \$25000.0}

Ключ: Audi A6

Значення: Product{Audi A6, 2021, \$55000.0}

Ключ: BMW X5

Значення: Product{BMW X5, 2020, \$60000.0}

Ключ: Tesla Model S

Значення: Product{Tesla Model S, 2022, \$89999.0}

Після зміни ціни Toyota Camry:

```
Toyota Camry -> Product{Toyota Camry, 2019, $23000.0}
```

```
Audi A6 -> Product{Audi A6, 2021, $55000.0}
```

```
BMW X5 -> Product{BMW X5, 2020, $60000.0}
```

```
Tesla Model S -> Product{Tesla Model S, 2022, $89999.0}
```

Після видалення BMW та додавання Mustang:

```
Toyota Camry -> Product{Toyota Camry, 2019, $23000.0}
```

```
Audi A6 -> Product{Audi A6, 2021, $55000.0}
```

```
Tesla Model S -> Product{Tesla Model S, 2022, $89999.0}
```

```
Ford Mustang -> Product{Ford Mustang, 2022, $45000.0}
```

Кількість елементів у Мар: 4

Мар очищено. Порожня? true

| Змн. | Арк. | Ліцензія | № докум. | Підпис | Дата | ДУ «Житомирська політехніка».25.121.01.000 – Пр1 | Арк. |
|------|------|-----------------|----------|--------|------|--|------|
| | | Піонерський В І | | | | | 8 |

Завдання 6. Продемонструвати роботу з класом Collections:

- Для роботи використати масив створений через Arrays.asList
- Метод Collections.sort()
- Метод Collections.binarySearch()
- Методы Collections.reverse(), Collections.shuffle()

- Метод Collections.fill()
- Методы Collections.max(), Collections.min()
- Метод Collections.copy()
- Метод Collections.rotate()
- Метод Collections.checkedCollection()
- Метод Collections.frequency()

Лістинг програми

```
package com.education.ztu;

import java.util.*;

public class Main5 {
    public static void main(String[] args) {
        List<Product> productList = new ArrayList<>(Arrays.asList(
            new Product("Tesla", "Model S", 2022, 89999),
            new Product("BMW", "X5", 2020, 60000),
            new Product("Toyota", "Camry", 2019, 25000),
            new Product("Ford", "Focus", 2018, 18000),
            new Product("Audi", "A6", 2021, 55000)
        ));
        System.out.println("Оригінальний список:");
        productList.forEach(System.out::println);
        Collections.sort(productList);
        System.out.println("\nСписок після Collections.sort() за ціною:");
        productList.forEach(System.out::println);
        Product key = new Product("Toyota", "Camry", 2019, 25000);
        int index = Collections.binarySearch(productList, key);
        System.out.println("\nІндекс Toyota Camry після binarySearch: " + index);
        Collections.reverse(productList);
        System.out.println("\nСписок після Collections.reverse():");
        productList.forEach(System.out::println);
        Collections.shuffle(productList);
        System.out.println("\nСписок після Collections.shuffle():");
        productList.forEach(System.out::println);
        Collections.fill(productList, new Product("Filled", "Product", 2025, 999));
        System.out.println("\nСписок після Collections.fill():");
        productList.forEach(System.out::println);
        List<Product> newList = new ArrayList<>(Arrays.asList(
            new Product("Tesla", "Model 3", 2023, 45000),
            new Product("BMW", "i4", 2021, 48000),
            new Product("Toyota", "Corolla", 2020, 22000),
            new Product("Ford", "Mondeo", 2022, 27000),
            new Product("Audi", "A4", 2021, 40000)
        ));
        System.out.println("\nМаксимальний (найдорожчий): " + Collections.max(newList));
        System.out.println("Мінімальний (найдешевший): " + Collections.min(newList));
        List<Product> copyList = new ArrayList<>(Arrays.asList(
            null, null, null, null, null
        ));
        Collections.copy(copyList, newList);
        System.out.println("\nСписок після Collections.copy():");
        copyList.forEach(System.out::println);
        Collections.rotate(copyList, 2);
        System.out.println("\nСписок після Collections.rotate(2):");
        copyList.forEach(System.out::println);
        Collection<Product> checked = Collections.checkedCollection(new ArrayList<>(), Product.class);
        checked.add(new Product("Skoda", "Octavia", 2020, 23000));
        System.out.println("\nПеревірена колекція (checkedCollection):");
        checked.forEach(System.out::println);
        Product duplicate = new Product("Ford", "Mondeo", 2022, 27000);
        newList.add(duplicate);
        newList.add(duplicate);
        int count = Collections.frequency(newList, duplicate);
        System.out.println("\nКількість входжень Ford Mondeo: " + count);
    }
}
```

| | | | | | | |
|------|------|------------------|--------|------|--|------|
| | | Aїсін В.С | | | ДУ «Житомирська політехніка».25.121.01.000 – Пр1 | Арк. |
| | | Плюнгівський В І | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | 9 |

```
} }

Оригінальний список:  
Product{Tesla Model S, 2022, $89999.0}  
Product{BMW X5, 2020, $60000.0}  
Product{Toyota Camry, 2019, $25000.0}  
Product{Ford Focus, 2018, $18000.0}  
Product{Audi A6, 2021, $55000.0}

Список після Collections.sort() за ціною:  
Product{Ford Focus, 2018, $18000.0}  
Product{Toyota Camry, 2019, $25000.0}  
Product{Audi A6, 2021, $55000.0}  
Product{BMW X5, 2020, $60000.0}  
Product{Tesla Model S, 2022, $89999.0}

Індекс Toyota Camry після binarySearch: 1

Список після Collections.reverse():  
Product{Tesla Model S, 2022, $89999.0}  
Product{BMW X5, 2020, $60000.0}  
Product{Audi A6, 2021, $55000.0}  
Product{Toyota Camry, 2019, $25000.0}  
Product{Ford Focus, 2018, $18000.0}

Список після Collections.shuffle():  
Product{Audi A6, 2021, $55000.0}  
Product{BMW X5, 2020, $60000.0}  
Product{Toyota Camry, 2019, $25000.0}  
Product{Tesla Model S, 2022, $89999.0}  
Product{Ford Focus, 2018, $18000.0}

Список після Collections.fill():  
Product{Filled Product, 2025, $999.0}  
Product{Filled Product, 2025, $999.0}  
Product{Filled Product, 2025, $999.0}  
Product{Filled Product, 2025, $999.0}  
Product{Filled Product, 2025, $999.0}

Максимальний (найдорожчий): Product{BMW i4
```

```
Список після Collections.copy():
Product{Tesla Model 3, 2023, $45000.0}
Product{BMW i4, 2021, $48000.0}
Product{Toyota Corolla, 2020, $22000.0}
Product{Ford Mondeo, 2022, $27000.0}
Product{Audi A4, 2021, $40000.0}
```

| | | | | | | |
|------|------|--------------------------|---------------|-------------|---|------------|
| | | <i>Аїсін В.С</i> | | | | |
| | | <i>Ліонтківський В І</i> | | | | |
| Змн. | Арк. | № докум. | <i>Підпис</i> | <i>Дата</i> | ДУ «Житомирська політехніка».25.121.01.000 – Лр 1 | Арк. 10 |

```
Список після Collections.rotate(2):  
Product{Ford Mondeo, 2022, $27000.0}  
Product{Audi A4, 2021, $40000.0}  
Product{Tesla Model 3, 2023, $45000.0}  
Product{BMW i4, 2021, $48000.0}  
Product{Toyota Corolla, 2020, $22000.0}
```

Перевірена колекція (checkedCollection):

Product{Skoda Octavia, 2020, \$23000.0}

Кількість входжень Ford Mondeo: 3

Висновки: ознайомився з Java Collections Framework.

| | | | | | | |
|------|------|---------------------------|--------|------|---|------------|
| | | <i>Аїсін В.С</i> | | | | |
| | | <i>Піонерківський В І</i> | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | ДУ «Житомирська політехніка».25.121.01.000 – Лр 1 | Арк. 11 |