

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра ЕОМ



Звіт

Лабораторна робота № 6

З дисципліни «Кросплатформні засоби програмування»

На тему: «ПАРАМЕТРИЗОВАНЕ ПРОГРАМУВАННЯ»

Виконала: ст. гр. КІ-303

Кілик І.Р.

Перевірив:

доцент кафедри ЕОМ

Іванов Ю.С.

Львів 2025

Мета: оволодіти навиками параметризованого програмування мовою Java.

ЗАВДАННЯ

1. Створити параметризований клас, що реалізує предметну область задану варіантом. Клас має містити мінімум 4 методи опрацювання даних включаючи розміщення та видалення елементів. Парні варіанти реалізують пошук мінімального елементу, непарні – максимального. Написати на мові Java та налагодити програму-драйвер для розробленого класу, яка мстить мінімум 2 різні класи екземпляри яких розміщуються у екземплярі розробленого класу-контейнеру. Програма має розміщуватися в пакеті Група.Прізвище.Lab6 та володіти коментарями, які дозволяють автоматично згенерувати документацію до розробленого пакету.
2. Автоматично згенерувати документацію до розробленого пакету.
3. Завантажити код на GitHub згідно методичних вказівок по роботі з GitHub.
4. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації та завантажити його у ВНС.
5. Дати відповідь на контрольні запитання.

ВАРИАНТ ЗАВДАННЯ

10. Пенал

The screenshot shows a Java documentation interface. At the top, there's a navigation bar with links for PACKAGE, TREE, INDEX, SEARCH, and HELP. Below that, the package name 'ki303kilyk.lab6' is displayed. On the left, there's a sidebar with 'Contents' and 'Description' sections, and a link to 'Classes and Interfaces'. The main content area is titled 'Package ki303kilyk.lab6' and contains the following information:

Classes	
Class	Description
Main	Клас Main для перевірки роботи пенала
Pen	Клас Pen представляє ручку
PenCase<T extends Comparable<T>>	Клас PenCase представляє пенал, який може зберігати об'єкти будь-якого типу, що реалізують Comparable.
Pencil	Клас Pencil представляє олівець

A search bar at the top right says 'Search documentation (type /)'.

Висновок: Завдяки виконанню даної лабораторної роботи, я оволоділа навиками параметризованого програмування мовою Java.

Додатки

```
package ki303kilyk.lab6;

/**
 * Клас Main для перевірки роботи пеналу
 */
public class Main {
    public static void main(String[] args) {
        // Створюємо пенал для ручок
        PenCase<Pen> penPenCase = new PenCase<>();
        penPenCase.addItem(new Pen("Синя", 140));
        penPenCase.addItem(new Pen("Червона", 150));
        penPenCase.addItem(new Pen("Чорна", 130));

        penPenCase.displayItems();
        System.out.println("Найдовша ручка: " + penPenCase.getMaxItem());

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

        // Створюємо пенал для олівців
        PenCase<Pencil> pencilPenCase = new PenCase<>();
        pencilPenCase.addItem(new Pencil("HB", 160));
        pencilPenCase.addItem(new Pencil("2B", 155));

        pencilPenCase.displayItems();
        System.out.println("Найдовший олівець: " + pencilPenCase.getMaxItem());
    }
}
```

```
package ki303kilyk.lab6;

/**
 * Клас Pen представляє ручку
 */
public class Pen implements Comparable<Pen> {
    private String color;
    private int length; // довжина ручки в мм

    public Pen(String color, int length) {
        this.color = color;
        this.length = length;
    }

    public String getColor() {
        return color;
    }

    public int getLength() {
        return length;
    }
}
```

```

@Override
public int compareTo(Pen other) {
    return Integer.compare(this.length, other.length); // порівнюємо за
довжиною
}

@Override
public String toString() {
    return "Ручка {колір=" + color + ", довжина=" + length + "мм}";
}

```

```

package ki303kilyk.lab6;

import java.util.ArrayList;
import java.util.List;

/**
 * Клас PenCase представляє пенал, який може зберігати об'єкти будь-якого типу, що
реалізують Comparable.
 * @param <T> Тип елементів у пеналі, який повинен бути порівнюваним
 */
public class PenCase<T extends Comparable<T>> {

    private List<T> items;

    /**
     * Конструктор, створює порожній пенал
     */
    public PenCase() {
        items = new ArrayList<>();
    }

    /**
     * Додає елемент у пенал
     * @param item елемент для додавання
     */
    public void addItem(T item) {
        items.add(item);
    }

    /**
     * Видаляє елемент з пеналу
     * @param item елемент для видалення
     * @return true, якщо елемент був видалений, false якщо елемент не знайдено
     */
    public boolean removeItem(T item) {
        return items.remove(item);
    }

    /**
     * Повертає максимальний елемент у пеналі
     */

```

```

    * @return максимальний елемент або null, якщо пенал порожній
    */
    public T getMaxItem() {
        if (items.isEmpty()) return null;
        T max = items.get(0);
        for (T item : items) {
            if (item.compareTo(max) > 0) {
                max = item;
            }
        }
        return max;
    }

    /**
     * Виводить всі елементи пеналу
     */
    public void displayItems() {
        if (items.isEmpty()) {
            System.out.println("Пенал порожній.");
            return;
        }
        System.out.println("Елементи пеналу:");
        for (T item : items) {
            System.out.println(item);
        }
    }

    /**
     * Повертає кількість елементів у пеналі
     * @return кількість елементів
     */
    public int getSize() {
        return items.size();
    }
}

```

```

package ki303kilyk.lab6;

/**
 * Клас Pencil представляє олівець
 */
public class Pencil implements Comparable<Pencil> {
    private String hardness;
    private int length; // довжина олівця в мм

    public Pencil(String hardness, int length) {
        this.hardness = hardness;
        this.length = length;
    }

    public String getHardness() {

```

```
        return hardness;
    }

    public int getLength() {
        return length;
    }

    @Override
    public int compareTo(Pencil other) {
        return Integer.compare(this.length, other.length); // порівнюємо за
довжиною
    }

    @Override
    public String toString() {
        return "Олівець {твердість=" + hardness + ", довжина=" + length + "мм}";
    }
}
```