



Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра інформаційні систем та технологій

**ЛАБОРАТОРНА РОБОТА №10**  
з дисципліни «Основи програмування - 2»  
Тема: «Функціональні інтерфейси та лямбда-вирази»

**Виконали:**

студенти групи ІА-31  
Клим'юк В.Л, Самелюк А.С,  
Дук М.Д, Сакун Д.С

**Перевірив:**

асистент кафедри ІСТ  
Степанов А. С.

**Тема:** Функціональні інтерфейси та лямбда-вирази

**Мета:** Мета цієї лабораторної роботи полягає в отриманні практичного досвіду використання функціональних інтерфейсів та лямбда-виразів в мові програмування Java. Це дозволить нам ознайомитися з основними концепціями функціонального програмування, такими як передача функцій як параметрів і використання анонімних функцій для скорочення коду.

### Хід роботи

#### Хід роботи:

1. Пригадати як використовувати функціональні інтерфейси та лямбда-вирази.
2. Для свого варіанту з л/р №2 першого семестру:
  - 2.1. Створити власний функціональний інтерфейс;
  - 2.2. За допомогою лямбда-виразу задати функцію свого варіанту;
  - 2.3. Передати функцію та параметри в метод, який буде викликати цю функцію, та виводити на консоль значення параметрів та результат обчислення функції.
3. Для свого варіанту з л/р №8 першого семестру:
  - 3.1. Переконатися, що у Вашому класі (класах) присутня реалізація методу **toString()**;
  - 3.2. Створити список об'єктів для свого варіанту (3-4 штуки);
  - 3.3. Вивести список об'єктів на консоль за допомогою передачі відповідного лямбда-виразу у наступний метод інтерфейсу **Iterable<T>**:

**default void forEach(Consumer<? super T> action).**

Примітка 1: інтерфейс **Iterable** є суперінтерфейсом для інтерфейсу **Collection**, тобто даний метод **forEach** присутній у будь-якій колекції типу **Collection**, **List** або **Set**.

Примітка 2: якщо Ваша IDE запропонує Вам замінити лямбда-вираз посиланням на метод (**method reference** або **member reference**), спробуйте (посилання на методи – тема наступної лекції).

Примітка 3: якщо все ще плутаєтесь при застосуванні вайлдкардів (**<? super T>**, **<? extends T>**), ознайомтесь зі статтею:

<https://stackoverflow.com/questions/4343202/difference-between-super-t-and-extends-t-in-java>

```

import java.util.Comparator;

public class ColorPixel extends Pixel{
    public int getR() {
        return r;
    }

    public void setR(int r) {
        this.r = r;
    }

    public int getG() {
        return g;
    }

    public void setG(int g) {
        this.g = g;
    }

    public int getB() {
        return b;
    }

    public void setB(int b) {
        this.b = b;
    }

    protected int r;
    protected int g;
    protected int b;

    public ColorPixel(int x, int y, int r, int g, int b){
        super(x, y);

        if ((r < 0 || r > 255) || (g < 0 || g > 255) || (b < 0 || b > 255))
            throw new IllegalArgumentException("RGB values must be in range from 0 to
255");

        this.r = r;
        this.g = g;
        this.b = b;
    }

    public ColorPixel(int r, int g, int b){
        super(0,0);

        if ((r < 0 || r > 255) || (g < 0 || g > 255) || (b < 0 || b > 255))
            throw new IllegalArgumentException("RGB values must be in range from 0 to
255");

        this.r = r;
        this.g = g;
        this.b = b;
    }
}

```

Приклад коду 1.1

```

import java.util.ArrayList;
import static java.lang.Math.*;

public class Main {
    public static void main(String[] args) {
        task1();
        System.out.println();
        task2();
    }
}

```

```

    }

    public static void task1(){
        System.out.println("Task 1");

        IFunction func1 = (double a, double b, double c, double d) -> {
            return sqrt(abs(sin(a) - 4.0 * log(b) / pow(c,d)));
        };
        IFunction func2 = (double a, double b, double c, double d) -> {
            return abs(atan(d)) * (exp(a) + 3.0 * log10(c)) / sqrt(pow(b, c));
        };
        IFunction func3 = (double a, double b, double c, double d) -> {
            return pow((2.0 * sin(a) + cos(abs(b * sqrt(c)))), d);
        };

        double[][] params = new double[3][4];
        params[0][0] = -1.49; params[0][1] = 23.4; params[0][2] = 1.23; params[0][3] =
2.542;
        params[1][0] = 2.34; params[1][1] = 0.756; params[1][2] = 2.23; params[1][3] = -
1.653;
        params[2][0] = 1.234; params[2][1] = -3.12; params[2][2] = 5.45; params[2][3] =
2.0;

        for(int i = 0; i < params.length; i++){
            System.out.println("Function " + (i + 1) + "; res:");
            switch (i){
                case 0:{ System.out.println(func1.calculate(params[i][0], params[i][1],
params[i][2], params[i][3])); break;}
                case 1:{ System.out.println(func2.calculate(params[i][0], params[i][1],
params[i][2], params[i][3])); break;}
                default:{ System.out.println(func3.calculate(params[i][0], params[i][1],
params[i][2], params[i][3]));}
            }
        }
    }

    public static void task2(){
        System.out.println("Task 2");

        ArrayList<Pixel> pixels = new ArrayList<>();
        pixels.add(new Pixel(1,1));
        pixels.add(new Pixel(2,2));
        pixels.add(new Pixel(3,3));
        pixels.add(new Pixel(4,4));

        pixels.forEach((i) -> {
            System.out.println(i);
        });
    }
}

```

## Приклад коду 1.2

```

import java.util.Comparator;

public class Pixel {
    public int getX() {
        return x;
    }

    public void setX(int x) {
        this.x = x;
    }
}

```

```

public int getY() {
    return y;
}

public void setY(int y) {
    this.y = y;
}

protected int x;
protected int y;

public Pixel(int x, int y) {
    this.x = x;
    this.y = y;
}

@Override
public boolean equals(Object o) {
    if (!(o instanceof Pixel other))
        return false;

    if (this.x == other.x && this.y == other.y)
        return true;
    return false;
}

@Override
public String toString() {
    return String.format("[X:%d ; Y:%d]", this.x, this.y);
}
}

```

### Приклад коду 1.3

```

@FunctionalInterface
public interface IFunction {
    double calculate(double a, double b, double c, double d);
}

```

### Приклад коду 1.4

```

Task 1
Function 1; res:
2.9064822366992433
Function 2; res:
16.025348462088733
Function 3; res:
5.892728703404708

Task 2
[X:1 ; Y:1]
[X:2 ; Y:2]
[X:3 ; Y:3]
[X:4 ; Y:4]

Process finished with exit code 0
|

```

Результат виконання програми.

**Висновки:** Під час виконання лабораторної роботи наша група отримала практичний досвід використання функціональних інтерфейсів та лямбда-виразів в мові програмування Java. Ми ознайомилися з основними концепціями функціонального програмування, такими як передача функцій як параметрів і використання анонімних функцій для скорочення коду. Набуті нами навички сприятимуть здатності писати чистий, ефективний і зрозумілий код у майбутньому.