

20251121 Wednesday

- Last class, inner class and anonymous inner class
- Today:
Lambda expressions, animation, and quiz
- Next class: Recursion (Chapter 18)
- Final Exam (Test 3). In class test only. NO online test!
- Examples from W3school, geeksforgeeks

Practice (after class/quiz)

```
interface Greeting {  
    void sayHello();  
}  
  
class MyClass implements Greeting{  
    public void sayHello() {  
        System.out.println("Hello from anonymous class");  
    }  
}
```

```
public class Main {  
    public static void main(String[] args) {  
        Greeting g = new MyClass();  
        g.sayHello();  
    }  
}
```

- Inner class. “**STATIC**”

```
public class Main {  
    public static void main(String[] args) {  
        Greeting g = new MyClass();  
        g.sayHello();  
    }  
  
    static class MyClass implements Greeting{  
        public void sayHello() {  
            System.out.println("Hello from anonymous class");  
        }  
    }  
}  
  
interface Greeting {  
    void sayHello();  
}
```

Anonymous inner class

```
interface Greeting {  
    void sayHello();  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Greeting g = new Greeting() {  
            public void sayHello() {  
                System.out.println("Hello from anonymous class");  
            }  
        };  
        g.sayHello();  
    }  
}
```

Lambda Expression

```
interface Greeting {  
    void sayHello();  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Greeting g = () -> System.out.println("Hello from lambda");  
        g.sayHello();  
    }  
}
```

Lambda Expression in Java

`(int x, int y) → x + y`

Parameter List

Lambda

Body

Simplifying Event Handling Using Lambda Expressions

Lambda expression is a new feature in Java 8. Lambda expressions can be viewed as an anonymous method with a concise syntax.

Concise, functional-style code.

enable passing code as parameters or assigning it to variables,

cleaner and more readable programs.

- Lambda expressions implement a functional interface (An interface with only one abstract function)
- Enable passing code as data (method arguments).
- Allow defining behavior without creating separate classes.

Why Use Lambda Expressions?

- **Concise Code:** Reduce boilerplate compared to anonymous classes.
- **Functional Programming:** Treat functions as first-class citizens.
- **Improved Readability:** Code is easier to read and maintain.
- **Enhanced Collections and Streams:** Simplify operations like filtering, mapping, and iterating

Basic Syntax for a Lambda Expression

The basic syntax for a lambda expression is either

`(type1 param1, type2 param2, ...) -> expression`

or

`(type1 param1, type2 param2, ...) -> { statements; }`

The data type for a parameter may be explicitly declared or implicitly inferred by the compiler. The parentheses can be omitted if there is only one parameter without an explicit data type.

Practice

```
public class Main{  
    public static void main(String[] args){  
        Operation add = (a, b) -> a + b;  
        int result = add.operate(10, 20);  
        System.out.println("result: " + result);  
        Operation sub = (x, y) -> y*y - x + 123;  
        result = sub.operate(10, 20);  
        System.out.println("result: " + result);  
    }  
}  
  
interface Operation{  
    int operate(int a, int b);  
}
```

Animation

- Good website for java swing:
- [Swing - Happy Coding](https://happycoding.io/tutorials/java/swing#)



- javax.swing **Class Timer**

Constructor	Description
<code>Timer(int delay, ActionListener listener)</code>	Creates a <code>Timer</code> and initializes both the initial delay and between-event delay to <code>delay</code> milliseconds.

timer

```
new Timer(16, new ActionListener(){
    public void actionPerformed(ActionEvent e){
        step();
        repaint();
    }
}).start();
```

```
private void step(){
    circleY++;
    if(circleY > getHeight()){
        circleY = 0;
    }
}
```

- Below two slides are one complete animation example from the website <https://happycoding.io/tutorials/java/swing>

```
import javax.swing.JFrame;
public class MyGui{
    public static void main(String[] args){
        JFrame frame = new JFrame("from HappyCoding.io");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        CustomComponent customComponent = new CustomComponent();
        frame.add(customComponent);

        frame.setSize(1000, 1000);
        frame.setVisible(true);
    }
}
```



```

import javax.swing.JPanel;
import javax.swing.Timer;
import java.awt.Graphics;
import java.awt.Color;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
public class CustomComponent extends JPanel{
    private int circleY = 0;
    public CustomComponent(){
        setBackground(new Color(0, 255, 255));
        new Timer(16, new ActionListener(){
            public void actionPerformed(ActionEvent e){
                step();
                repaint();
            }
        }).start();
    }
    private void step(){
        circleY++;
        if(circleY > getHeight()){
            circleY = 0;
        }
    }
    @Override
    public void paintComponent(Graphics g){
        super.paintComponent(g);
        g.setColor(Color.RED);
        g.fillOval(getWidth()/2 - 10, circleY, 20, 20);
    }
}

```

Quiz time

Today, lambda expression and animation briefly
Now, quiz time.

Final exam. NO online option.