JAVA + SPRING MVC

שיעור06 : Handler interceptors/Adapters

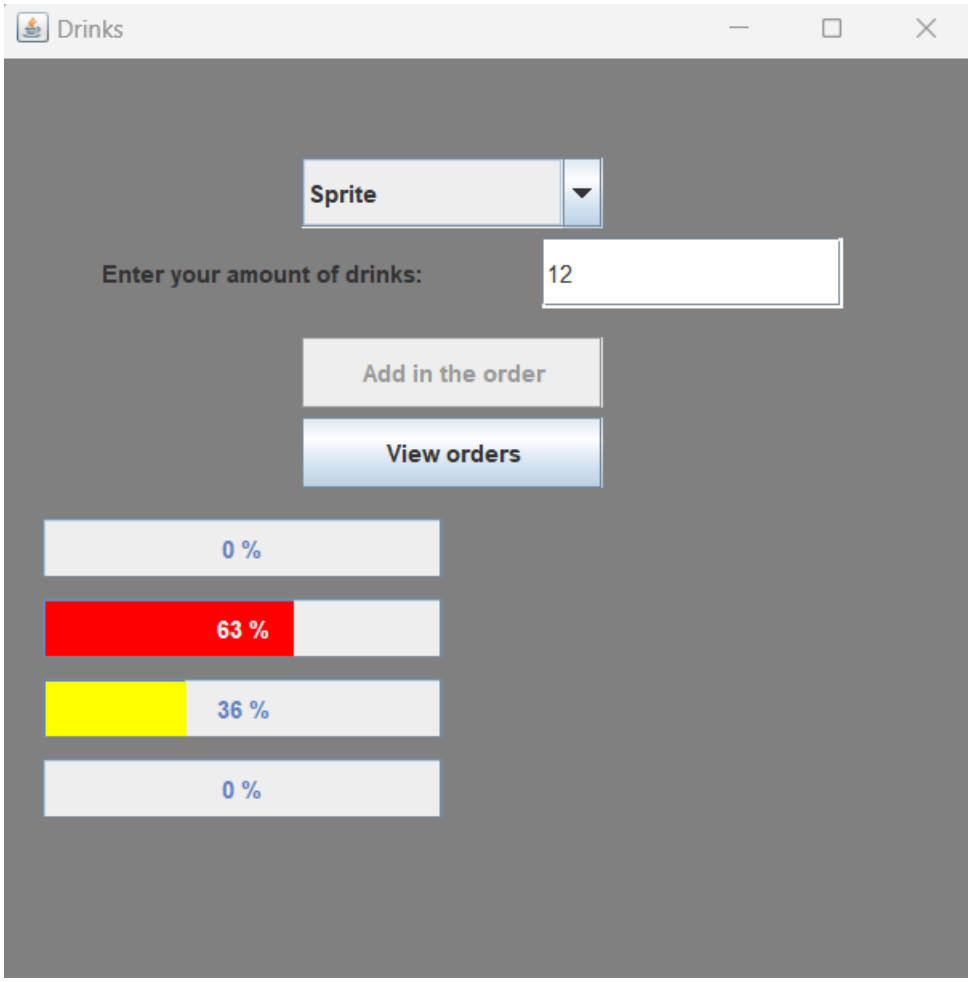
Adapters/Handler interceptors

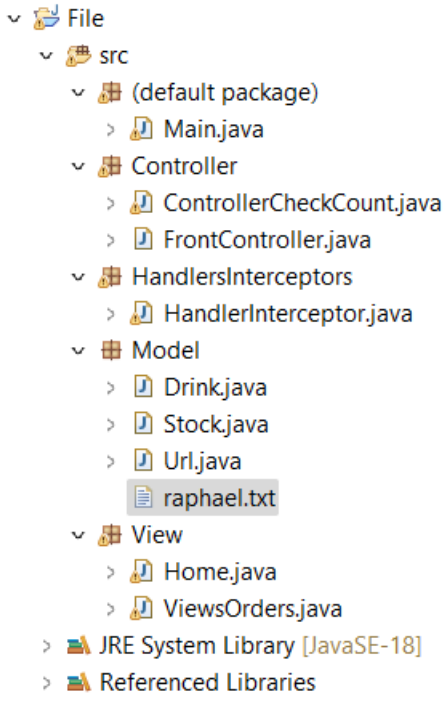
במילים פשוטות Spring Interceptor, הוא מחלקה שמרחיבה את המחלקה HandlerInterceptorAdapter או מיישמת את ממשק HandlerInterceptor.

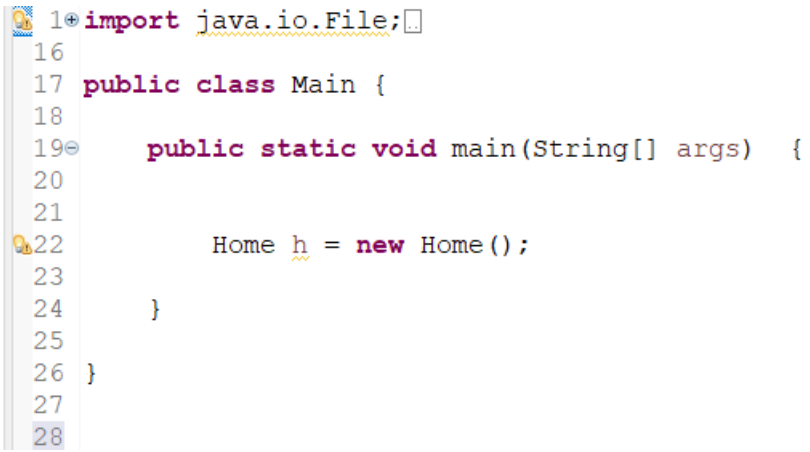
HandlerInterceptorמכיל שלוש שיטות עיקריות:

- **prehandle()** נקרא לפני ביצוע המטפל בפועל
- **postHandle()** נקרא לאחר ביצוע המטפל
- **afterCompletion()** נקרא לאחר סיום הבקשה המלאה והתצוגה נוצרת

שלוש השיטות הללו מספקות גמישות לביצוע כל מיני עיבודים לפני ואחרי.


Drinks
Sprite
Enter your amount of drinks:
12
Add in the order
View orders
0 %
63 %
36 %
0 %


File
src
(default package)
Main.java
Controller
ControllerCheckCount.java
FrontController.java
HandlersInterceptors
HandlerInterceptor.java
Model
Drink.java
Stock.java
Url.java
raphael.txt
View
Home.java
ViewsOrders.java
JRE System Library [JavaSE-18]
Referenced Libraries



```
import java.io.File;

public class Main {

    public static void main(String[] args)  {

        Home h = new Home();

    }

}
```

```
package Controller;

import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Scanner;

public class ControllerCheckCount {

    private String count;
    private String drink;

    public ControllerCheckCount(String drink , String count) {
        this.drink=drink;
        this.count = count;
    }
```

```
	public void check() throws IOException {

		int count = Integer.valueOf(this.count).intValue();
		String myFile="";
		if(count>=0 && count<=30) {

			File file = new
File("C:\\Users\\RAPHAEL\\Desktop\\File\\src\\Model\\raphael.txt");
			Scanner scanner = new Scanner(file);
			while(scanner.hasNextLine()) {
				myFile = myFile+scanner.nextLine()+"\n";
			}
			FileWriter fw = new FileWriter(file);
			PrintWriter pw = new PrintWriter(fw);
			pw.println(myFile);
			String fileContent="";
			fileContent = fileContent+this.drink+" - "+this.count;
			pw.println(fileContent);

			pw.close();
		}


	}
```

```
	public String getCount() {
		return count;
	}

	public void setCount(String count) {
		this.count = count;
	}









}

package Controller;

import java.io.IOException;

import Model.Url;

public class FrontController {

	private Url url;
```

```
	public FrontController(Url url) {


		this.url = url;

	}


	public void getUrl1() throws IOException {

		if(this.url.getPart1().contains("Coca-Cola") ||
this.url.getPart1().contains("Fanta") || this.url.getPart1().contains("Sprite") ||
				this.url.getPart1().contains("Zero")) {


			ControllerCheckCount ch = new
ControllerCheckCount(this.url.getPart1(),this.url.getPart2());
		   ch.check();
		}


	}


	public Url getUrl() {
		return url;
	}


	public void setUrl(Url url) {
		this.url = url;
	}
```

}

```
package HandlersInterceptors;

import javax.naming.spi.DirStateFactory.Result;

public class HandlerInterceptor implements Runnable {

	private String request;

	public HandlerInterceptor(String request) {
		this.request = request;

	}

	public void prehandle() {
		String result = " ** "+request;
		System.out.println(result);


	}

	public void treatRequest() {
		System.out.println("I treat the request ");

	}

	public void posthandle() {
		String result = request+" $$ ";
		System.out.println(result);

	}


	public void afterCompletion() {
		System.out.println("Finished");
	}

	@Override
	public void run() {
		prehandle();
		treatRequest();
		posthandle();
		afterCompletion();

	}

}
```

```java
package Model;

public class Drink {

	private String name;
	private int count;

	public Drink(String name, int count) {
		this.name = name;
		this.count = count;
	}

	public String getName() {
		return name;
	}

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

	public int getCount() {
		return count;
	}

	public void setCount(int count) {
		this.count = count;
	}




}
```

```java
package Model;

import java.util.ArrayList;

public class Stock {

	private ArrayList<Drink> list;

	public Stock(ArrayList<Drink> list) {
		this.list = list;
	}

	public void addDrink(Drink d) {
		this.list.add(d);
	}

	public ArrayList<Drink> getList() {
		return list;
	}

	public void setList(ArrayList<Drink> list) {
```

```
            this.list = list;
      }



}
```

```
package Model;

public class Url {

      private String part1;
      private String part2;

      public Url(String part1, String part2) {
            this.part1 = part1;
            this.part2 = part2;
      }

      public String getPart1() {
            return part1;
      }

      public void setPart1(String part1) {
            this.part1 = part1;
      }

      public String getPart2() {
            return part2;
      }

      public void setPart2(String part2) {
            this.part2 = part2;
      }




}
```

```
package View;


import java.awt.Color;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;
```

```
import java.awt.event.KeyEvent;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.Random;

import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JProgressBar;
import javax.swing.JTextField;
import javax.swing.UIManager;

import Controller.FrontController;
import Model.Url;

public class Home extends JFrame{

	private JPanel panel;
	private JComboBox combo;
	private JTextField jtf;
	private JLabel label;
	public static  JButton button1;
	public static JButton button2;
	public static int countDrink = 0;
```

```
private JProgressBar bar1;
private JProgressBar bar2;
private JProgressBar bar3;
private JProgressBar bar4;
public static int val = 350;
private int countCoca = 0;
private int countFanta = 0;
private int countSprite= 0;
private int countZero = 0;
private double Ratio1 , Ratio2 , Ratio3 , Ratio4;
private int total = 0;



public Home() {
        this.setTitle("Drinks");
        this.setSize(500, 500);
        this.setLocationRelativeTo(null);
        panel = new JPanel();
        panel.setLayout(null);
        panel.setBackground(Color.GRAY);
        combo = new JComboBox();
        combo.addItem("Coca-Cola");
        combo.addItem("Fanta");
        combo.addItem("Sprite");
        combo.addItem("Zero");
```

```
combo.setBounds(150, 50, 150, 35);

label = new JLabel("Enter your amount of drinks:");
label.setBounds(50, 90, 220, 35);



jtf = new JTextField();
jtf.setBounds(270, 90, 150, 35);
jtf.addKeyListener(new KeyListener());

button1 = new JButton();
button1.setText("Add in the order");
button1.setBounds(150, 140, 150, 35);
button1.addActionListener(new ButtonListener());

button2 = new JButton();
button2.setText("View orders");
button2.setBounds(150, 180, 150, 35);
button2.addActionListener(new Button2Listener());
button2.setEnabled(false);

bar1 = new JProgressBar();
bar1.setMaximum(100);
bar1.setMinimum(0);
bar1.setValue(0);
bar1.setStringPainted(true);
```

```
        bar1.setBounds(20, 230, 200, 30);
        bar1.setVisible(true);
        bar1.setStringPainted(true);
        bar1.setForeground(Color.blue);

bar2 = new JProgressBar();
bar2.setMaximum(100);
bar2.setMinimum(0);
bar2.setValue(0);
bar2.setStringPainted(true);
bar2.setBounds(20, 270, 200, 30);
bar2.setVisible(true);
bar2.setStringPainted(true);
        bar2.setForeground(Color.red);


bar3 = new JProgressBar();
bar3.setMaximum(100);
bar3.setMinimum(0);
bar3.setValue(0);
bar3.setStringPainted(true);
bar3.setBounds(20, 310, 200, 30);
bar3.setVisible(true);
bar3.setStringPainted(true);
        bar3.setForeground(Color.yellow);
```

```
    bar4 = new JProgressBar();
    bar4.setMaximum(100);
    bar4.setMinimum(0);
    bar4.setValue(0);
    bar4.setStringPainted(true);
    bar4.setBounds(20, 350, 200, 30);
    bar4.setVisible(true);
    bar4.setStringPainted(true);
        bar4.setForeground(Color.green);

        panel.add(combo);
        panel.add(label);
        panel.add(jtf);
        panel.add(button1);
        panel.add(button2);
        panel.add(bar1);
        panel.add(bar2);
        panel.add(bar3);
        panel.add(bar4);

        this.setContentPane(panel);
        this.setVisible(true);

}

public static void ratio(JProgressBar bar , double x) {
```

```
        bar.setValue(((int)x*100)/100);
}

public class KeyListener implements java.awt.event.KeyListener{

        @Override
        public void keyTyped(KeyEvent e) {
                // TODO Auto-generated method stub

        }

        @Override
        public void keyPressed(KeyEvent e) {
                jtf.setBackground(Color.white);
        }

        @Override
        public void keyReleased(KeyEvent e) {
                // TODO Auto-generated method stub

        }


}

public class Button2Listener implements ActionListener{
```

```
		@Override
		public void actionPerformed(ActionEvent e) {



				try {
					ViewsOrders orders = new ViewsOrders();
				} catch (FileNotFoundException e1) {
					// TODO Auto-generated catch block
					e1.printStackTrace();
				}




	}
	}

	public class ButtonListener implements ActionListener{

		@Override
		public void actionPerformed(ActionEvent e) {
		total = total + Integer.valueOf(jtf.getText()).intValue();

			if(combo.getSelectedItem().toString().equals("Coca-Cola")) {
				countCoca = countCoca +
Integer.valueOf(jtf.getText()).intValue();
			}
```

```
                else if(combo.getSelectedItem().toString().equals("Fanta")) {
                        countFanta = countFanta +
Integer.valueOf(jtf.getText()).intValue();

                }
                else if(combo.getSelectedItem().toString().equals("Sprite"))
{
                        countSprite = countSprite +
Integer.valueOf(jtf.getText()).intValue();

                }
                else if(combo.getSelectedItem().toString().equals("Zero")) {
                        countZero = countZero +
Integer.valueOf(jtf.getText()).intValue();

                }


                 Ratio1 =((double)countCoca/total)*100;
                 Ratio2 =((double)countFanta/total)*100;
                 Ratio3 =((double)countSprite/total)*100;
                 Ratio4 = ((double)countZero/total)*100;

                System.out.println(" - "+countCoca);
                System.out.println(" - "+countFanta);
                System.out.println(" - "+countSprite);
                System.out.println(" - "+countZero);
```

```
                    System.out.println("** - "+Ratio1);
                    System.out.println("** - "+Ratio2);
                    System.out.println("** - "+Ratio3);
                    System.out.println("** - "+Ratio4);



                    ratio(bar1,Ratio1);
                    ratio(bar2,Ratio2);
                    ratio(bar3,Ratio3);
                    ratio(bar4,Ratio4);


                    if(Integer.valueOf(jtf.getText()).intValue()>30){
                            jtf.setBackground(Color.red);


                    }else {
                            countDrink = countDrink +
Integer.valueOf(jtf.getText()).intValue();

                            if(countDrink >30) {
                                    button1.setEnabled(false);
                                    button2.setEnabled(true);
                            }else {
```

```
                                    Url url = new
Url(combo.getSelectedItem().toString(),jtf.getText());

                                    FrontController fc = new FrontController(url);
                                    try {
                                        fc.getUrl1();
                                    } catch (IOException e1) {
                                        // TODO Auto-generated catch block
                                        e1.printStackTrace();
                                    }
                                }
                            }



                        }

                }

                }
```

```
package View;

import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

import javax.swing.*;

public class ViewsOrders extends JFrame{

	private JTable jt;
	private JPanel panel;
	private String textFile="";
	public static int count=0;

	public ViewsOrders() throws FileNotFoundException {
		this.setTitle("ViewsOrders");
		this.setSize(500, 500);
		this.setLocationRelativeTo(null);
		panel = new JPanel();
		panel.setLayout(null);

		File file = new
File("C:\\Users\\RAPHAEL\\Desktop\\File\\src\\Model\\raphael.txt");
		Scanner scanner = new Scanner(file);
```

```
            String data[][] = new String[10][1];
            int row=0;

            while(scanner.hasNextLine()) {
                        data[row][0]=scanner.nextLine()+"\n";
                        row++;
                  }

      String column[]={"Orders"};
      jt=new JTable(data,column);
      jt.setBounds(30,40,200,300);
      JScrollPane sp=new JScrollPane(jt);

      this.add(sp);
      this.setVisible(true);

   }
}
```