**DD1339 Introduktion till datalogi 2013/2014**

**Uppgift nummer: Hemuppgift 3**

**Namn: Marcus Larsson**

**Grupp nummer: 5**

**Övningsledare: Marcus Dicander**

**Betyg: ..... Datum: ............. Rättad av: .....................................**

## Resterande uppkgift från hemuppgift 2:

Jag missade att det var en extra uppgift att göra i förra hemuppgiften och blev ombedd att skriva den denna gång. Därav bifogar jag koden av endast den metoden.

Implement a method `setPrice` that sets the price of tickets to a new value. The new price should be passed as a parameter to the method. Test your new method by creating a machine, showing the price of tickets, changing the price, and then showing the new price.

```
/**
     * This method will change the prive of the tickets.
     * Not possible to enter negative value. Then price will automatically be set to 0.
     * @param price Enter new price of the tickets.
     */
    public void setPrice(int price){
        if(price>0){
            this.price = price;
        } else {
            this.price=0;
        }
    }
```

## Exercise 2.92 och 2.93

### Nedan följer källkod från klassen Heater:

```
/**
 * This is a student project that simulates a heating control unit where you can increase or decrease the temperature.
 *
 * @author (Marcus Larsson)
 * @version 2013-09-14
 */
public class Heater
{
    private double temperature, min, max, increment;

    /**
     * Constructor for objects of class Heater.
     * You have to set the minimun and maximun allowed temperature upon creation.
     * @param inMin Enter the minimum allowed temperature.
     * @param inMax Enter the maximum allowed temperature.
     */
    public Heater(double inMin, double inMax)
    {
        temperature = 15.0;
        min = inMin;
        max = inMax;
        increment = 5.0;
    }

    /**
```

```
     * This method will increase the temperature.
     * If trying to increase above maximun allowed temperature,
     * the temperature will be set to the maximun allowed value instead.
     */
    public void warmer()
    {
        double newTemperature = temperature + increment;
        if(newTemperature<=max)
        {
            if(newTemperature>temperature){
                setTemperature(newTemperature);
            }
        }else{
            System.out.println("Too high!!! Maximum allowed temperature is: "+max);
            setTemperature(max);
        }
    }

    /**
     * This method will decrease the temperature.
     * If trying to decrease below minimum allowed temperature,
     * the temperature will be set to the minimum allowed value instead.
     */
    public void cooler()
    {
        double newTemperature = temperature - increment;
        if(newTemperature>=min)
        {
            if(newTemperature<temperature){
                setTemperature(newTemperature);
            }
        }else{
            System.out.println("Too low!!! Minimum allowed temperature is: "+min);
            setTemperature(min);
        }
    }

    /**
     * This is a private method that supports the methods warmer and cooler.
     * It only sets the temperature to what the other methods are requesting and prints out a message
about the new temperature.
     */
    private void setTemperature(double newTemperature){
        temperature=newTemperature;
        System.out.println("Temperature was set to: "+temperature);
    }

    /**
     * This method will set the value of which the methods cooler and warmer will decrease or
increase with.
     * The value cannot be set to negative. If trying to set a negative number, no new value will be set
and an error will be printed to the screen.
```

```
     * @param newIncrement Enter a positive number.
     */
    public void setIncrement(double newIncrement)
    {
        if(newIncrement>0)
        {
            increment = newIncrement;
        }else{
            System.out.println("An increment cannot be less than 0...");
        }

    }

    /**
     * This method will return the current temperature
     * @return The current temperature of this Heater object.
     */
    public double getTemperature(){
        return temperature;
    }
}
```