

Påskeopgaver



1. TV

Skriv en klasse, der simulerer et TV. Der skal være mulighed for at vælge kanel, volumen og tænde og slukke for TV. Husk at det ikke giver mening at forsøge at ændre lyd/kanal, hvis TV er slukket 😊
Prøv at unit teste følgende adfærd:

```
TV tv1 = new TV();  
tv1.turnOn();  
tv1.setChannel(30);  
tv1.setVolume(3);
```

```
TV tv2 = new TV();  
tv2.channelUp();  
tv2.channelUp();  
tv2.turnOn();  
tv2.volumeUp();
```

2. Cirkel

Skriv en klasse, der repræsenterer en cirkel med en radius som attribut og en set metode.
Hvis metoden modtager en negativ radius som parameter, skal der kastes en `Exception`.

Der skal være en metode til at udregne cirkelens areal.

3. Arv & interfaces

Hvad har dyr og frugt tilfælles? Vi spiser (nogle af) dem 😊
Udgangspunktet for denne opgave er følgende interfaces og abstrakte klasser:

A.

```
public interface Edible {  
    /** describe how to eat */  
    public abstract String howToEat();
```

B.

```
abstract class Animal {
    private double weight;

    public double getWeight() {
        return weight;
    }

    public void setWeight(double weight) {
        this.weight = weight;
    }

    /** Return animal sound */
    public abstract String sound();
}
```

C.

```
abstract class Fruit implements Edible {
    // Data fields, constructors, and methods omitted here
}
```

Du skal nu implementere flg. klasser (find selv på indhold i de metoder, som du selv skal implementere):

```
class Chicken extends Animal implements Edible

class Tiger extends Animal

class Apple extends Fruit

class Orange extends Fruit
```

Skriv en testklasse med følgende main metode:

```
public class TestEdible {
    public static void main(String[] args) {
        Object[] objects = {new Tiger(), new Chicken(), new Apple()};
        for (int i = 0; i < objects.length; i++) {
            if (objects[i] instanceof Edible)
                System.out.println(((Edible)objects[i]).howToEat());

            if (objects[i] instanceof Animal) {
                System.out.println(((Animal)objects[i]).sound());
            }
        }
    }
}
```

Afsluttende spørgsmål:

Hvad gør metoden `instanceof`?

Hvad vil det sige at en metode er abstrakt?

Kan man have en ikke-abstrakt klasse med en abstrakt metode? Hvis ikke, hvorfor ikke?

Hvad betyder `@override`?

.....

Take a Break

