



## Java Advanced

# Week 1: Herhaling

### **DE HOGESCHOOL MET HET NETWERK**

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt  
[www.pxl.be](http://www.pxl.be) - [www.pxl.be/facebook](http://www.pxl.be/facebook)



# Wat is er fout?

```
Scanner scanner = new Scanner(System.in);  
int number = scanner.nextInt();  
  
if (number = 1) {  
    System.out.println("Choose 1");  
} else {  
    System.out.println("Choose other");  
}
```



= VS ==

```
Scanner scanner = new Scanner(System.in);  
int number = scanner.nextInt();  
  
if (number == 1) {  
    System.out.println("Choose 1");  
} else {  
    System.out.println("Choose other");  
}
```



# Wat is er fout?

```
public void getMeat(String type) {  
    if (type == "beef") {  
        System.out.print("Choose beef");  
    } else if (type == "pork") {  
        System.out.print("Choose pork");  
    }  
}
```



# Objecten vergelijken: .equals ipv ==

```
public void getMeat(String type) {  
    if (type.equals("beef")) {  
        System.out.print("Choose beef");  
    } else if (type.equals("pork")) {  
        System.out.print("Choose pork");  
    }  
}
```



# Wat is er fout?

```
int[] numbers = {256};  
int firstNumber = numbers[1];
```



# 0-based index

```
int[] numbers = {256};  
int firstNumber = numbers[0];
```



# Exceptions

```
public class Sum {  
    public static void main(String[] args) {  
        int a = 0;  
        int b = 0;  
  
        try {  
            a = Integer.parseInt(args[0]);  
            b = Integer.parseInt(args[1]);  
  
        } catch (NumberFormatException ex) {  
        }  
  
        int sum = a + b;  
  
        System.out.println("Sum = " + sum);  
    }  
}
```





# Exceptions

```
public class Sum {  
    public static void main(String[] args) {  
        int a = 0;  
        int b = 0;  
  
        try {  
            a = Integer.parseInt(args[0]);  
            b = Integer.parseInt(args[1]);  
  
        } catch (NumberFormatException ex) {  
            ex.printStackTrace();  
        }  
  
        int sum = a + b;  
  
        System.out.println("Sum = " + sum);  
    }  
}
```



# Klassen en objecten

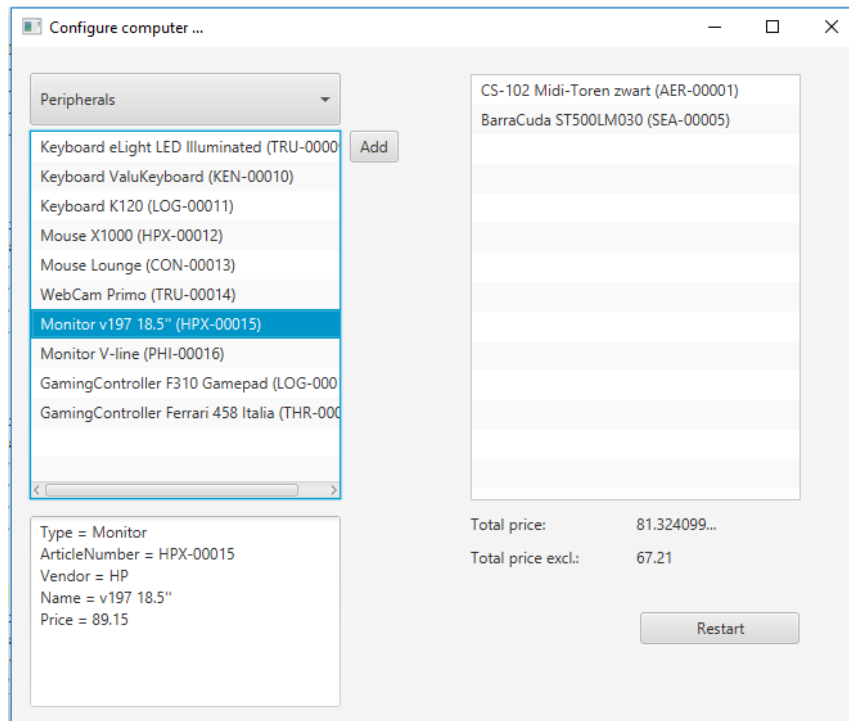
```
public class Person {  
    private int age;  
    private String name;  
  
    public Person(String name) {  
        this.name = name;  
    }  
  
    public int getAge() {  
        return age;  
    }  
  
    public void setAge(int age) {  
        this.age = age;  
    }  
  
    @Override  
    public String toString() {  
        return name;  
    }  
}
```

```
public class PersonUsage {  
    public static void main(String[] args) {  
        Person willie = new Person("Willie Wortel");  
        willie.setAge(40);  
        int value = willie.getAge();  
    }  
}
```



# De opdracht

Ontwikkel een softwarepakket waarmee een computerwinkel een computer kan configureren.



# Taak 1

- ☐ Voorzie een klasse `ComputerCase` (behuizing).

Plaats deze klasse in het package `“be.pxl.computerstore.hardware”`.

- ☐ De klasse voorziet volgende member variabelen:

- `vendor` (Leverancier)
- `name`
- `price`
- `dimension` (gebruik hiervoor de bestaande klasse `Dimension`)
- `weight`

- ☐ Voorzie een constructor waarbij je `vendor`, `name` en `price` voorziet als argumenten (volgorde is belangrijk voor de testen).

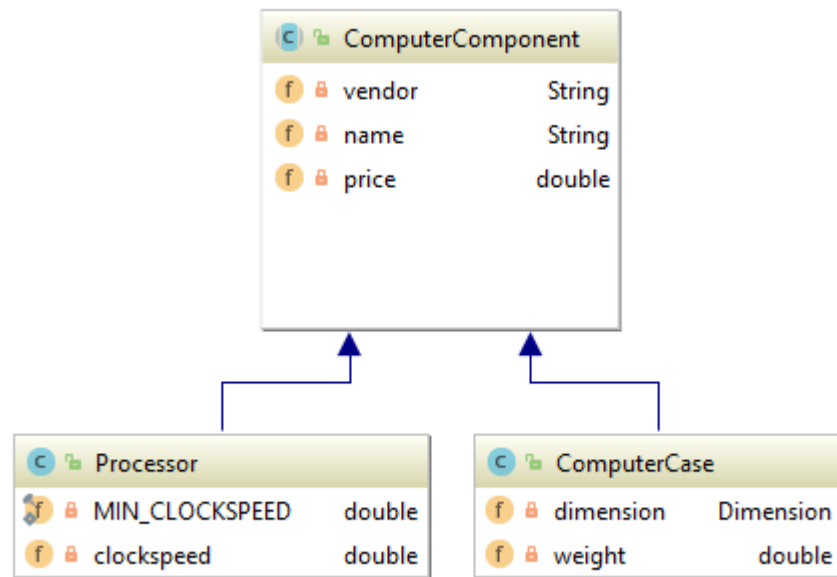
- ☐ Test je klasse door de testen in `ComputerCaseTest` uit te voeren.



## Taak 2

- ☐ Voorzie een klasse Processor in het package “be.pxl.computerstore.hardware”.
- ☐ De klasse voorziet volgende member variabelen:
  - vendor
  - name
  - price
  - clockspeed: deze is minimaal 0,7GHz
- ☐ Voorzie een constructor waarbij je alle eigenschappen voorziet.
- ☐ Hoe vermijd je dubbele code?
- ☐ Test je implementatie door de testklasse ProcessorTest uit te voeren.





## Taak 3

- ☐ Onze computeronderdelen moeten ook voorzien worden van een artikelnummer (articleNumber). Dit genereer je bij de aanmaak van een nieuw object. Dit zijn de spelregels:
  - ☐ Eerste 3 letters van de vendor in uppercase, voor een vender met minder dan 3 karakters voeg je extra “X”-en toe. Vendor “HP” geeft dus “HPX”.
  - ☐ Dan voeg je “-” toe.
  - ☐ Een uniek nummer waaraan je eventueel “0”-en vooraan toevoegt om steeds 5 karakters te hebben. Gebruik een klassevariabele om het unieke getal te genereren.
  - ☐ Opmerking: een artikelnummer kan niet meer veranderen! Je kan dus ook de vendor van computeronderdelen niet aanpassen.
- ☐ Voer de testen in de klasse ComputerCaseArticleNumberTest uit.



## Taak 4

- ❑ Implementeer in de klasse `ComputerComponent` de methode `toString()`. De naam van het onderdeel wordt getoond en tussen haakjes het artikelnummer.

VS4-V ATX (SHA-00002)

Core i9 79000X (INT-00007)

- ❑ Je kan testen met de testklasse `ComputerComponentToStringTest`.





## Taak 5

- ❑ Maak een methode aan met de naam `getFullDescription()` waarbij alle eigenschappen van de computeronderdelen worden opgelijst.
- ❑ Hier is een voorbeeld van het resultaat van `getFullDescription()` voor een `Processor` object en een `ComputerCase` object:

ArticleNumber = AMD-00008-944  
Vendor = AMD  
Name = Ryzen 7 1800X  
Price = 400.27  
Clock speed = 2.4GHz

ArticleNumber = SHA-00002-574  
Vendor = Sharkoon  
Name = VS4-V ATX  
Price = 30.63  
Width = 200mm  
Height = 430mm  
Depth = 445mm  
Weight = 3.5kg

- ❑ Vermijd dubbele code!
- ❑ Je kan testen met de testklasse `ComputerComponentFullDescriptionTest`.



## Taak 6

- ☐ Voorzie een klasse Peripheral (=randapparaat). Hiervan kan je geen objecten maken.
- ☐ Alle randapparaten hebben de volgende eigenschappen:
  - ☐ Vendor
  - ☐ Name
  - ☐ Price
- ☐ Voorzie een constructor waarmee je alle eigenschappen een waarde kan geven.
- ☐ Nu kan je de klassen Monitor, Mouse, WebCam, Keyboard en GamingController laten overerven van de klasse Peripheral.
- ☐ Overschrijf de toString() methode, zodat eerst de klassenaam getoond wordt en vervolgens naam en artikelnummers:  
bv. Mouse X100 (HPX-00012) of Monitor V-Line 2000 (HPI-00125)
- ☐ Voorzie in de klasse WebCam de eigenschap frameRate (int).
- ☐ Voorzie voor de klasse Keyboard nog een extra eigenschap KeyboardLayout met als mogelijke waarden QWERTY of AZERTY. Voeg deze eigenschap ook toe aan de parameterlijst van de constructor.



## Taak 7

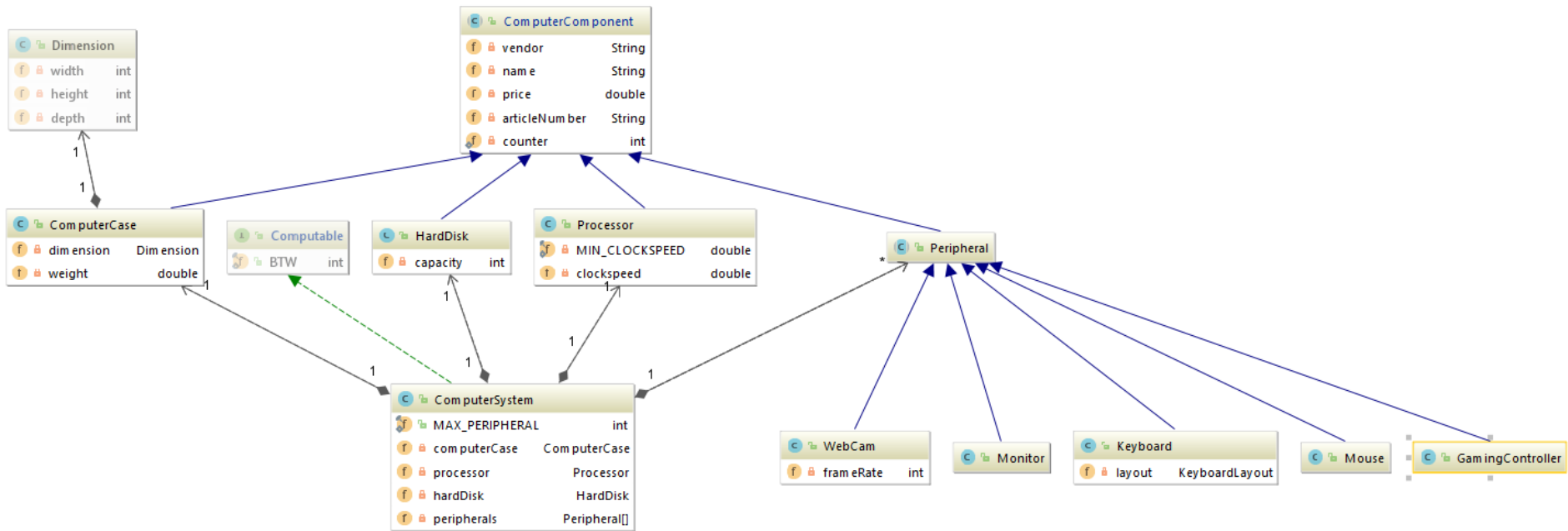
- ☐ Maak nu een klasse `ComputerSystem` met een default constructor.
- ☐ Voorzie de volgende eigenschappen met hun getters en setters:
  - ☐ `processor`
  - ☐ `hardDisk`
  - ☐ `computerCase`
- ☐ Voorzie een array waar je maximum 3 `Peripherals` aan toe kan voegen.  
Voorzie de methoden:
  - ☐ `addPeripheral(Peripheral peripheral)`: om een `Peripheral` object aan de array toe te voegen. Throw de `TooManyPeripheralsException` wanneer het toegelaten aantal `Peripheral` objecten wordt overschreden.
  - ☐ `getNumberOfPheripherals()`: om het aantal randapparaten op te vragen
- ☐ Test alles met de testklasse `ComputerSystemTest`.



## Taak 8

- ☐ Neem de interface `Computable` en implementeer de default methode `totalPriceIncl`.
- ☐ De klasse `ComputerSystem` implementeert de interface `Computable`. De prijs van een computer is de som van de prijs van alle onderdelen.
- ☐ Test alles mbv de testklasse `ComputerSystemComputableTest`.





## Taak 9

- ❑ Wanneer je de klasse ComputerSystemConfiguratorApp nu start, wordt een gui getoond waar je een computersysteem kan configureren.

