



Programming Expert

# Lambda expressions

## Oefeningen

**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)



# Oefening 1: Inner class

- Test het voorbeeld uit.

```
public class TextApp {  
    public static void main(String[] args) {  
        String s = "hello";  
        Text text = new Text(  
            "Hello this is an example of a sentence containing words");  
        System.out.println("*** Words containing 'e' ***");  
        text.printFilteredWords(new WordFilter() {  
@Override  
        public boolean isValid(String s) {  
            return s.contains("e"); }  
        });  
  
        text.printFilteredWords("hello how are you doing") ; }  
}
```



# Oefening 1: Lambda expression

- Test het voorbeeld uit.

```
public class TextApp {  
    public static void main(String[] args) {  
        final String c = "e";  
        Text text = new Text(  
            "Hello this is an example of a sentence  
containing words");  
        System.out.println("*** Words containing  
'e' ***");  
        text.printFilteredWords((s) -> s.contains(c)) ;  
    }  
}
```



# Oefening 1

- Voeg volgende filters toe:
  - enkel woorden die als tweede letter 'e' hebben.
  - enkel woorden die twee keer de letter 'e' bevatten.



# Oefening 2:

- Maak de functionele interface `WordProcessor` met volgende functionele method:
  - `public String process(String s)`
- Voeg in de klasse `Text` een methode toe die de woorden afdrukt nadat ze verwerkt zijn door een `WordProcessor`:
  - `public void printProcessedWords(WordProcessor wordprocessor)`
- Maak een utility-interface `TextUtil` met volgende methode:
  - `public static String reverse(String s)`
- Gebruik in je hoofdapplicatie een methode-referentie naar de statische methode `reverse()` om de woorden omgekeerd af te drukken.



# Oefening 3: Word Processing

- Maak een nieuwe klasse TextScrambler met volgende methode:
  - `public String scramble(String s)`
  - Deze methode vervangt volgende letters in een woord:
    - a -> @
    - e -> @
    - l -> 1
    - o -> 0
- Gebruik een instantie van deze klasse om de woorden verhaspeld af te drukken.
- Druk de woorden af met kleine letters. Gebruik hiervoor een methode-referentie
- Maak een functionele interface NumberParser met volgende functionele methode:
  - `public Double parse(String s)`
- Voeg aan de klasse Text een methode printSum() toe die de som afdruckt van alle getallen die vervat zijn in de tekst.



# Oefening 4: Hello world

- Maak een hello world programma dat gebruik maakt van volgende lambda expressie:  
() -> "hello world"



# Oefening 5: Filter builder

Maak een klasse FilterBouwer met de volgende methodes:

- WordFilter bouwBevat(String tekst)
- WordFilter bouwMinimumLengte(int minLen)
- WordFilter bouwLetterControle(char letter, int pos)
- WordFilter bouwDubbelCheck(  
WordFilter filter1, WordFilter filter2)

Test je methodes.





Niet kennen/kunnen

**EXTRA:**

**DENK- EN ZOEKVRAGEN**

# Extra 1: Functional Programming

Maak en test de volgende code. Wat doet dit?

```
public class WordProcessorUtil {  
    public WordProcessor combo(  
        WordProcessor daarna, WordProcessor eerst) {  
        return (String w) -> daarna.process(eerst.process(w));  
    }  
}
```

Voorbeeld:

```
WordProcessor w = WordProcessorUtil.combo(  
    w -> "" + w + "", w::toUpperCase);
```



# Extra 2: Text

Splits de showFilteredWords methode van Text in 3:

- Text splits(String teken)
- Text filter(Predicate<String> conditie)
- void show(Consumer<String> shower)
- void reset()

Hou intern een Supplier<String> data bij, die je telkens aanpast in bovenstaande methodes.

Test je methode als volgt:

```
new Text("wat een dag, wat een dag")  
    .splits(" ")  
    .filter(w -> w.charAt(1) == 'a')  
    .show(System.out::println);
```



## Extra 3: Local variable

Maak een outer klasse met een lokale inner klasse. Test de volgende technieken om toch lokale variabelen te kunnen gebruiken/wijzigen:

- Geef de lokale variabelen mee in de constructor van de local inner class
- Maak extra velden in de outer klasse
- Kan je lokale Objecten wijzigen? Probeer van de lokale variabelen arrays met lengte 1 te maken.

Welke techniek werkt het best? Zijn er nog?



## Extra 4: Extended Functional Interface (moeilijk)

Maak IndexedWordFilter.

Maak de interface functioneel.

Zorg dat de interface overerft van WordFilter.

Voorzie de methode test(String woord, int index), die naast het woord ook de index van het woord geeft, het hoeveelste woord het is in de zin.

Zorg dat de interface bruikbaar is zonder de code van Text te wijzigen, de klasse met printFilteredWords.

Hint: default

