



# Java Essentials

## Hoofdstuk 1

### Programmeren in Java

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

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



# Inhoud

1. Introductie
  1. Wat is java?
  2. Java als platform
  3. Soorten java toepassingen
2. Mijn eerste java programma
  1. Kennismaking met IntelliJ
  2. Opbouw van het programma
3. Variabelen
  1. Variabelen
  2. Datatypes
  3. De klasse String
4. Input van de gebruiker



# Inhoud

- 5. Operatoren
- 6. Condities en logische operatoren
- 7. Iteraties
  - 1. While-loop
  - 2. For-loop
- 8. Arrays
- 9. Twee-dimensionale arrays
- 10. String formatering



## HISTORY

Appeared in

**1995**

=



**James Gosling**



```
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
} //The traditional Hello world program can be written in Java.
```



*Duke*  
the Java mascot

## VERSION



Today:  
Java SE 13  
Java EE 8

## JAVA FRAMEWORKS



Java software runs on everything from laptops to data centers, game consoles to scientific supercomputers.

## TOP 10

Most popular programming languages in use



**930 million**

Java Runtime Environment downloads each year



**3 billion**

Mobile phones run Java

**9 million Developers**

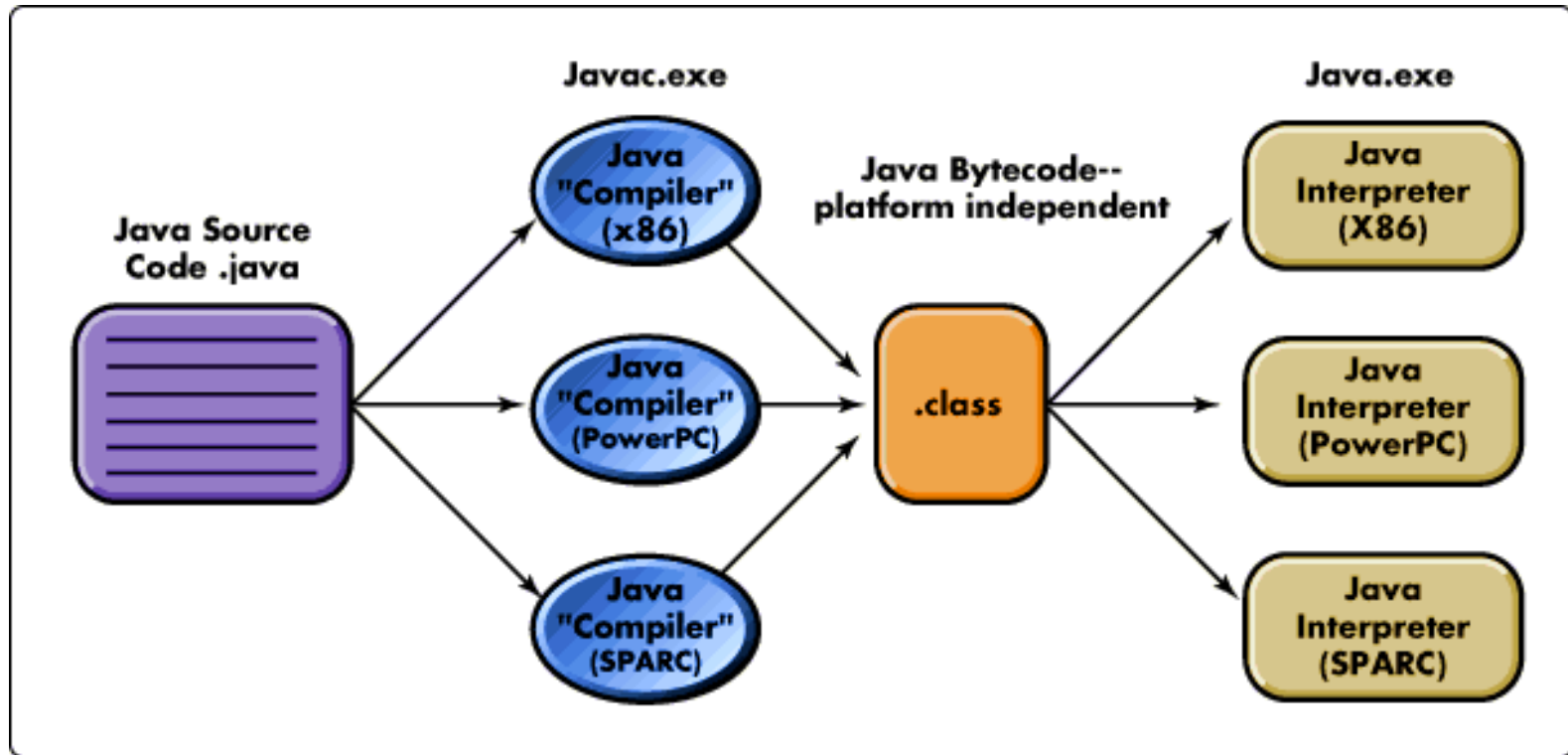
Resources: <http://www.javatpoint.com/>

## IDE



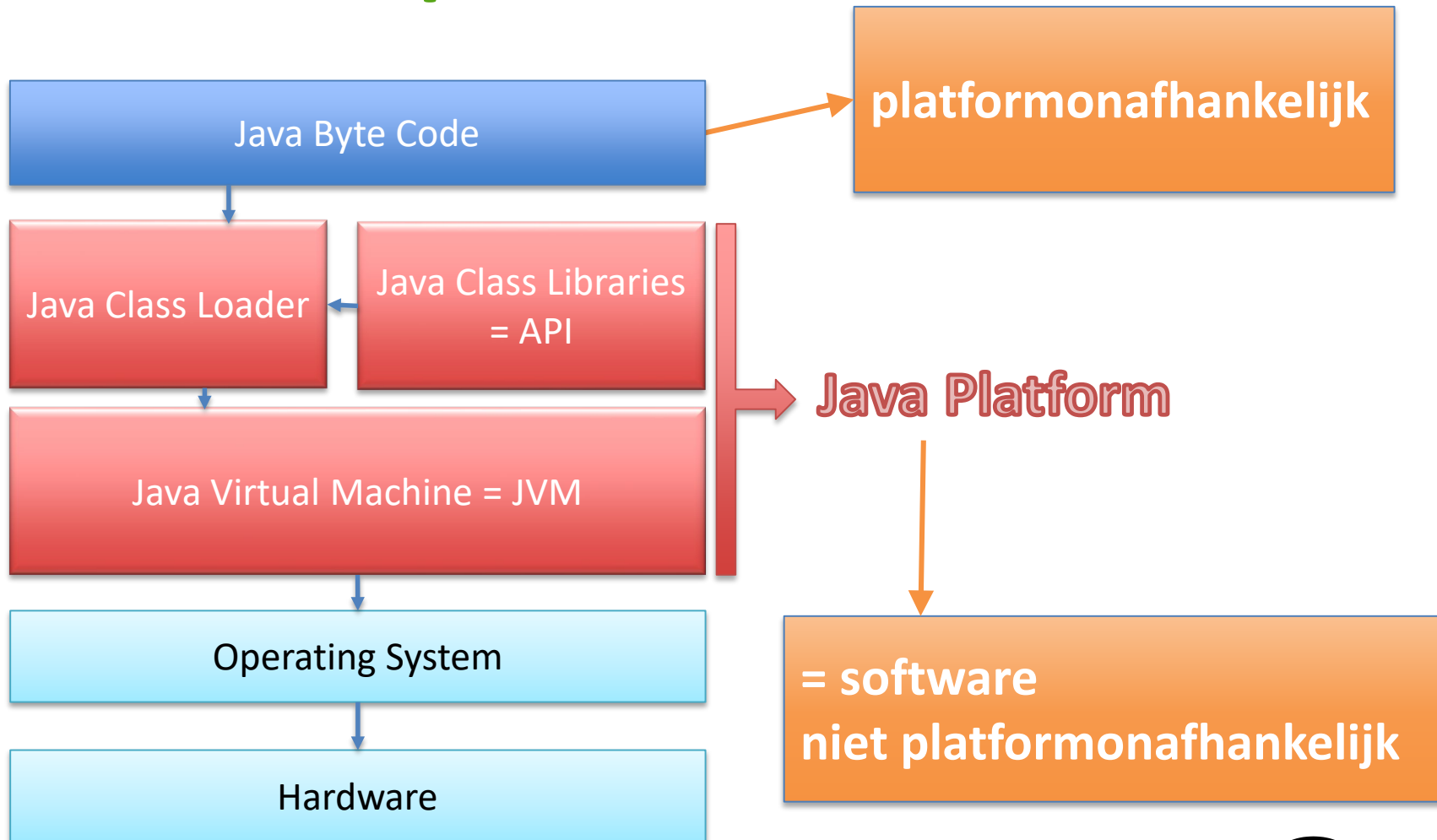
IDE is a software that provides facilities to programmers for software development.

# 1.1 Wat is java?



Java broncode (.java) wordt eerst gecompileerd naar platform-onafhankelijke bytecode (.class). Deze bytecode wordt dan geïnterpreteerd door de Java Virtual Machine.

# 1.2 Java als platform



- Java Virtual Machine (**JVM**): interpreteert de bytecode en maakt gebruik van de onderliggende hardware en het onderliggende besturingssysteem om de instructies uit te voeren
- Java Application Programming Interface (**Java API**): verzameling van software-componenten die gebruikt kunnen worden door het Java-programma (gegroepeerd in packages)



# 1.3 Soorten Java toepassingen



## Mobile Phones

If you have an Android phone you use Java every day! Android apps – and indeed the Android operating system! - are written in Java, with Google's API, which is similar to JDK.



## Point of Sale Systems

Java is also used in the creation of PoS systems, helping businesses exchange goods or services for money from their customers.



## Video Games

One of the most popular games of all time, Minecraft, was written in Java by Mojang. Minecraft is a sandbox construction game, where you can build anything you can imagine.



## Trading Applications

Several third-party trading applications use Java. Murex, which is used by many banks for front to back connectivity, is also written in Java.



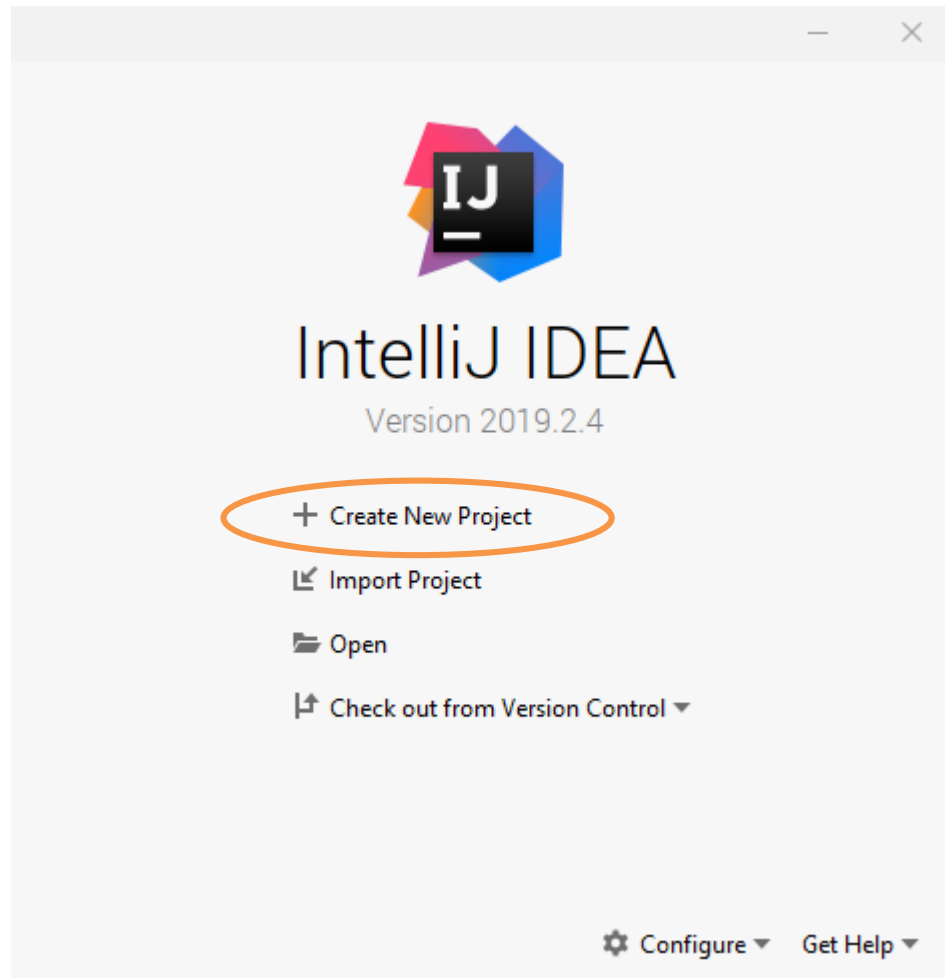
## Big Data Technologies

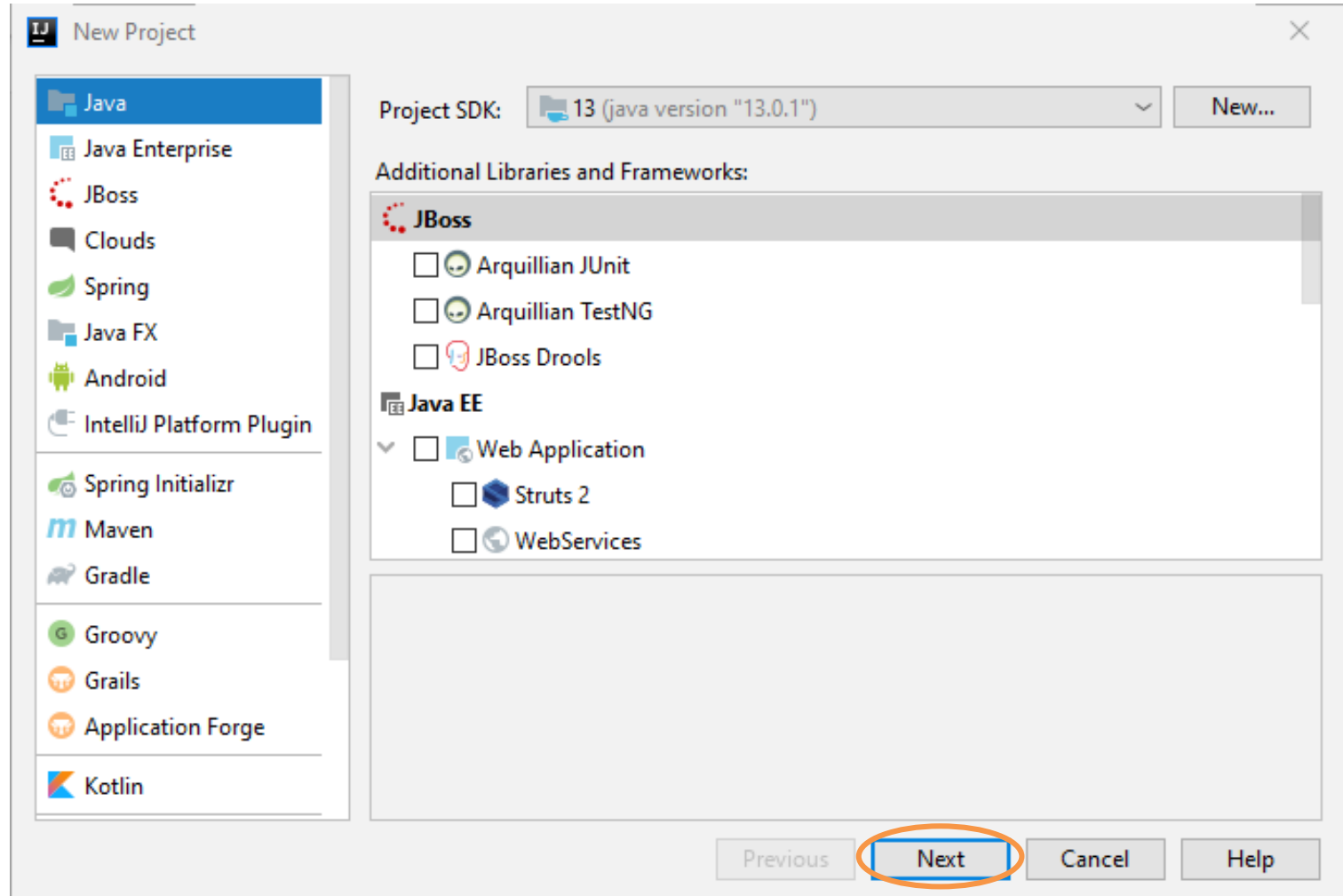
The Java platform is very popular in writing high-performance systems. Hadoop and Elasticsearch are both written in Java and are often used in Big Data projects.

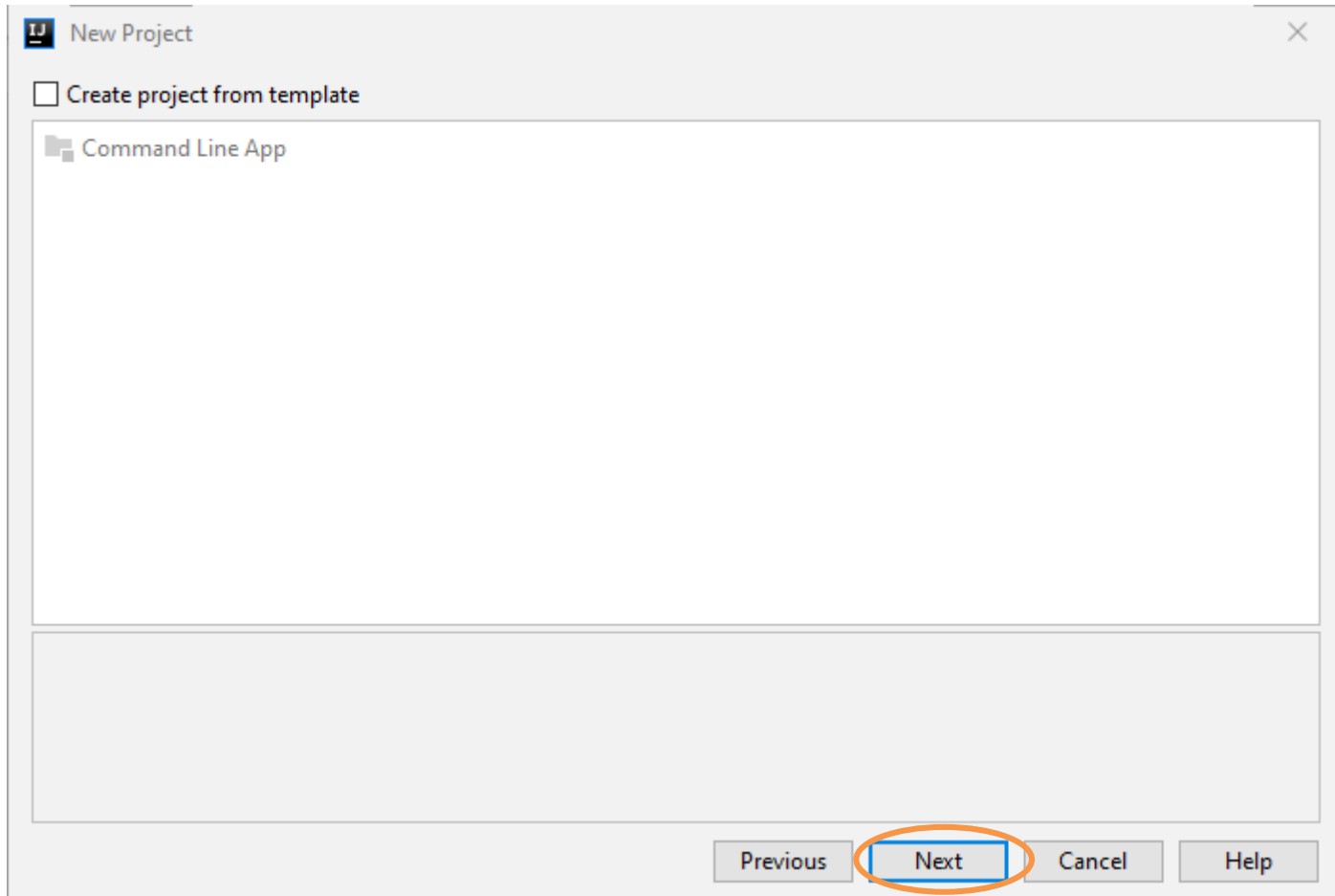



## 2. Mijn eerste java programma

### Opdracht 1: Kennismaking met IntelliJ







 New Project ✕

Project name:

Project location:  ...

▼ More Settings

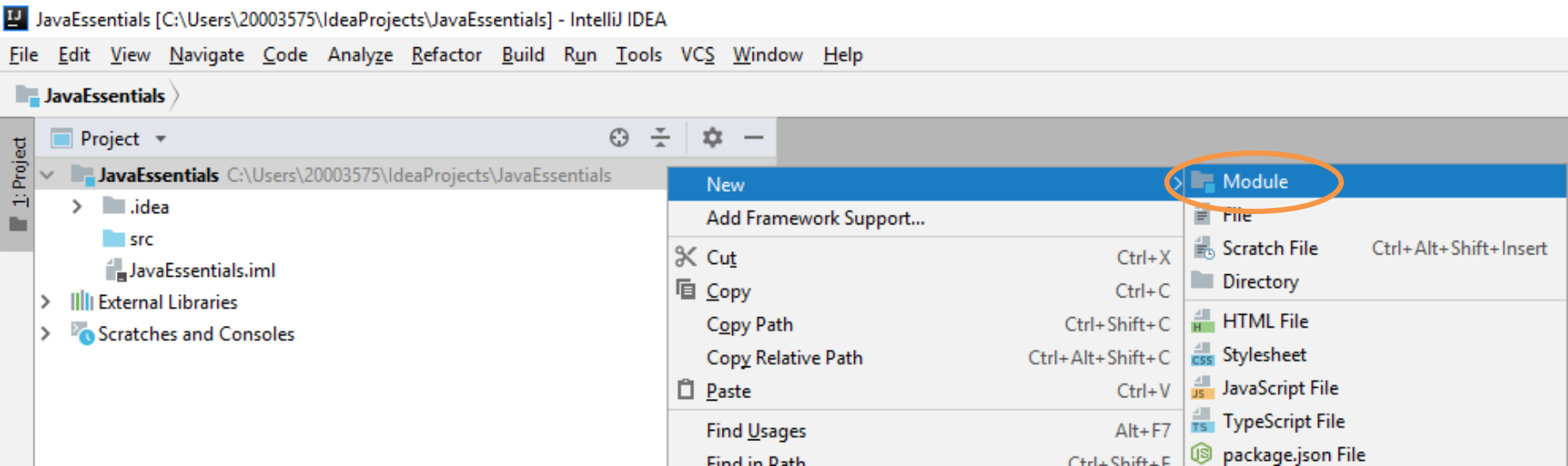
Module name:


Content root:  📁

Module file location:  📁


Project format:  ▼






 New Module ✕

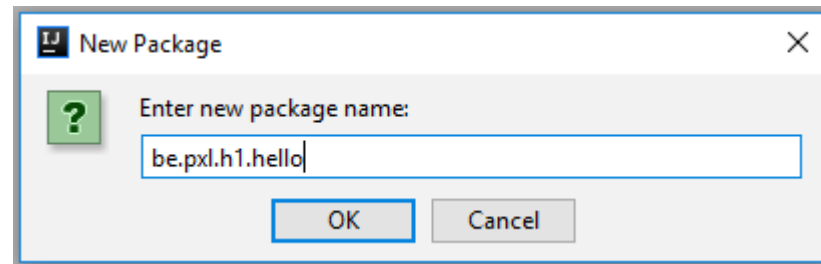
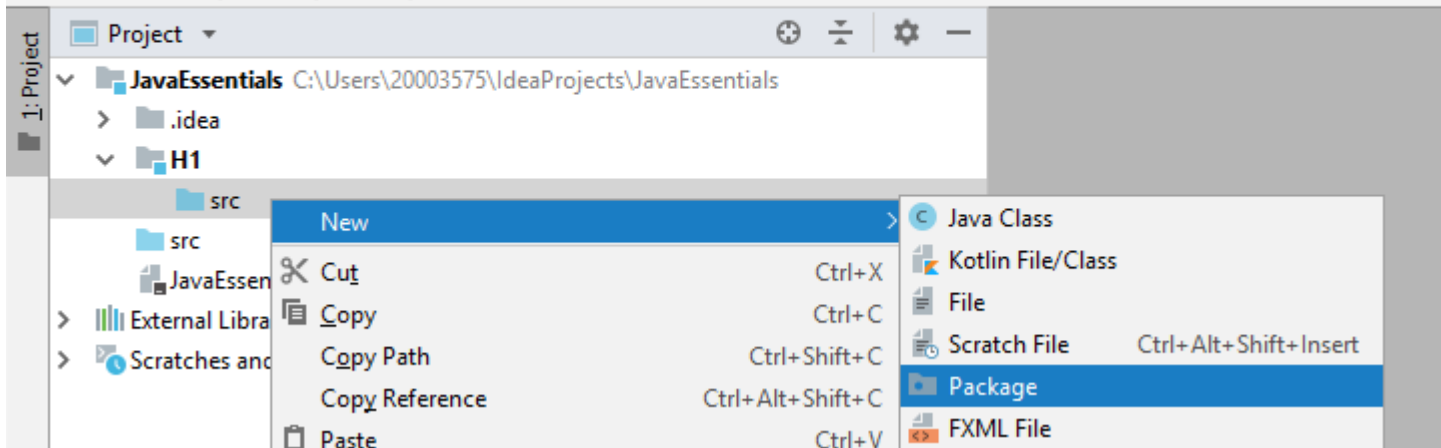
Module name:

Content root:  

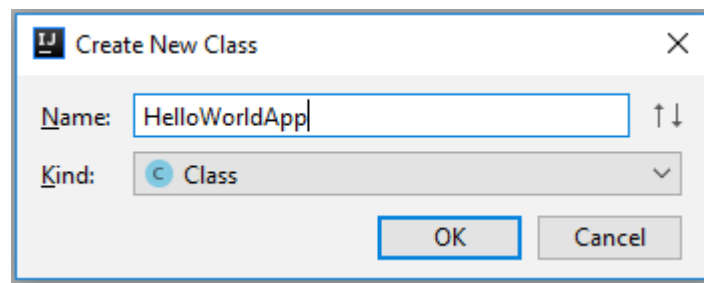
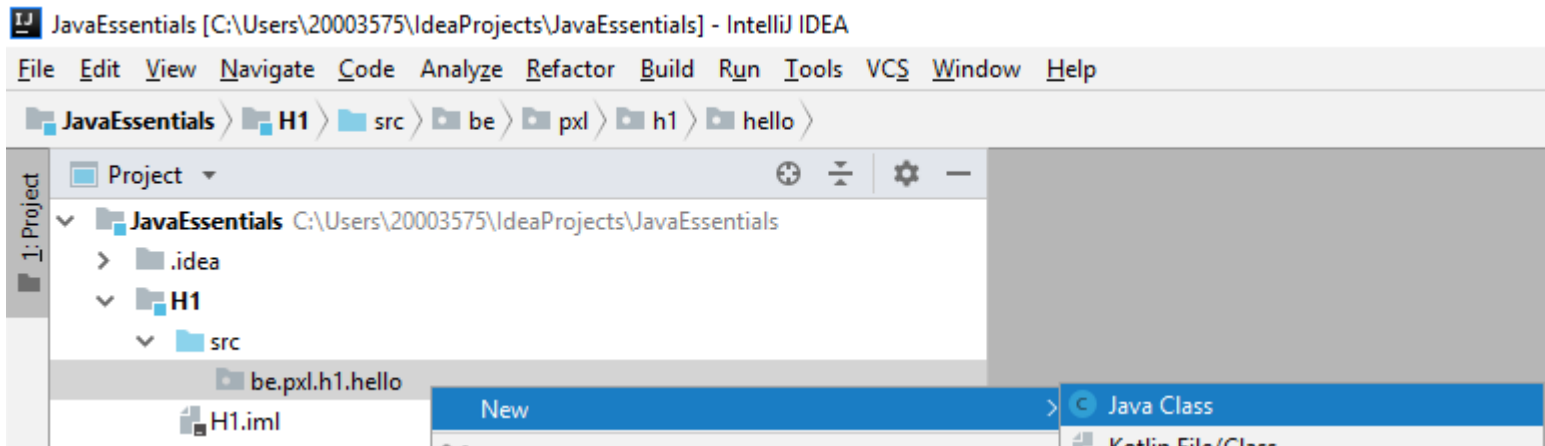
Module file location:  

Previous Finish Cancel Help









JavaEssentials [C:\Users\20003575\IdeaProjects\JavaEssentials] - ... \H1\src\be\pxl\h1\hellc

File Edit View Navigate Code Analyze Refactor Build Run Tools V

JavaEssentials > H1 > src > be > pxl > h1 > hello > HelloWorldA

Project

- JavaEssentials C:\Users\20003575\IdeaProjects\JavaEssentials
  - .idea
  - H1
    - src
      - be.pxl.h1.hello
        - HelloWorldApp
- H1.iml
- src
- JavaEssentials.iml
- External Libraries
- Scratches and Consoles

= Projectnaam

src = source = map met broncode

package (pakket) = submap

klassenaam

➔ C:\projectfolder\JavaEssentials\H1\src\be\pxl\h1\hello\HelloWorldApp.java



```
HelloWorldApp.java x
1 package be.px1.h1.hello;
2
3 public class HelloWorldApp {
4     public static void main(String[] args) {
5         System.out.println("Hello World!");
6     }
7 }
8
```

Punkomma



## 2.2 Opbouw van het programma

### 1. Het pakket definiëren

```
package be.px1.h1.hello;
```

Pakketten gebruiken we om orde / structuur te brengen in de code (cfr. mappenstructuur Verkenner)

### 2. De klasse definiëren

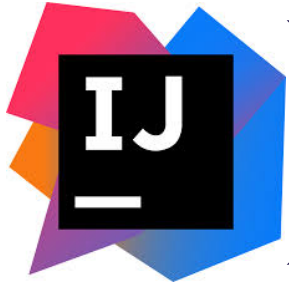
Hierin schrijf je de eigenlijke code.



### 3. De methode main()

```
public static void main(String[] args) {...}
```

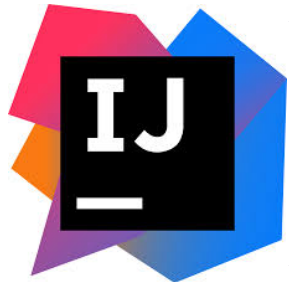
Klasse met methode main() = het hoofdprogramma.



Je kan deze main()-methode genereren door **psvm** te typen en daarna op <Enter> te drukken!

### 4. Het eigenlijke werk

```
System.out.println("Hello World!");
```



Je kan de bovenstaande methode genereren als je **sout** typt en daarna op <Enter> drukt!



## 5. Drie soorten commentaar

`/* indien er meerdere regels commentaar  
in een programma geschreven worden */`

`/** documentation JAVADOC */`

`// een regel commentaar`



## Opdracht 2: Wijzig het programma

- Voeg bovenaan de klasse volgende regel commentaar toe  
author: naam voornaam
- Zorg ervoor dat ook volgende regel tekst wordt afgedrukt:  
Dit is mijn eerste java programma.





# 3. Variabelen / 3.1 Variabelen

```
package be.px1.demo;

public class Sum {
    public static void main(String[] args) {
        int a = 5;
        int b = 6;
        int sum = a + b;
        System.out.println(sum);
    }
}
```

Puntkomma

Gegevenstype van de variabele

a	0..0101
b	0..0110
→ sum	0..1011

## 3.2 Primitieve datatypes

Type	Formaat	Bereik
<b>Gehele getallen</b>		
int	32 bit	-2147483648 ... 2147483647
long	64 bit	-9223372036854775808 ... 9223372036854775807
<b>Reële getallen</b>		
double	64 bit	
<b>Andere types</b>		
boolean	1 bit	
char	16 bit	

## Opdracht 3: *Waarde toekennen aan variabele*

- Maak een programma waarin je variabelen van volgende datatypes een waarde geeft.
  - *boolean*
  - *char*
  - *int*
  - *long*
  - *double*



## Typeconversie

= gegevens van een bepaald datatype omzetten naar een ander datatype

- **Impliciete conversie** (automatisch door de compiler)

```
int anInteger = 5;  
long aLong;  
aLong = anInteger;
```

int  
(32 bit)

past in!!!

long  
(64 bit)



- **Expliciete conversie (= casten)**

```
long aLong = 115L;  
int anInteger;  
anInteger = aLong;
```

long  
(64 bit)

Past NIET  
in!!!

int  
(32 bit)

casten

= verantwoordelijkheid van  
de programmeur!  
Wanneer loopt het mis?



```
anInteger = (int) aLong;
```

# Rangorde van de datatypes

Rangorde	Type
1	double
2	long
3	int

*int + long = long*

*int + double = double*

*int \* long = long*

*double \* double = double*

*int / int = int*

*int / double = double*

!!! Hier maak je snel  
een fout tegen.

Vb.  $10 / 4 = 2$  i.p.v.  $2.5$



Bij wiskundige bewerkingen is het datatype van het resultaat het type van de operand met de hoogste rang.

## Opdracht 4:

*Geef na iedere bewerking de waarde van de variabele*

```
int i;  
int n = 3;  
double d;  
i = n + 4;  
i = n * 4;  
i = 7 + 2 * 3;  
n = n * (n + 2) * 4;  
d = 3.5 / 2;  
n = 7 / 4;
```



# De declaratie van variabelen

```
public class Demo1 {  
    public static void main(String[] args) {  
        int number;  
        number = 5;  
        int sum = number + 2;  
        double salary = 1234.5;  
        long bigNumber = 123456789123456789L;  
        number = (int) salary;  
        System.out.println("Salary: " + number);  
        number = (int) bigNumber;  
        System.out.println("Big number: " + number);  
        char letter = 'a';  
        boolean geslaagd = true;  
        String text = "Goedenacht!";  
    }  
}
```

L of l

Casting

Concatenatie

# Datatype char


```
public class CharVoorbeeld {  
    public static void main(String[] args) {  
        char karakter = 'a';  
        System.out.println(karakter);  
        System.out.println(karakter + 1);  
        karakter = (char) (karakter + 1);  
        System.out.println(karakter);  
    }  
}
```

a  
98  
b



## Opdracht 5: *Het bereik van variabelen*

Waarom geeft de compiler een foutindicatie in lijn 6, 11 en 13?

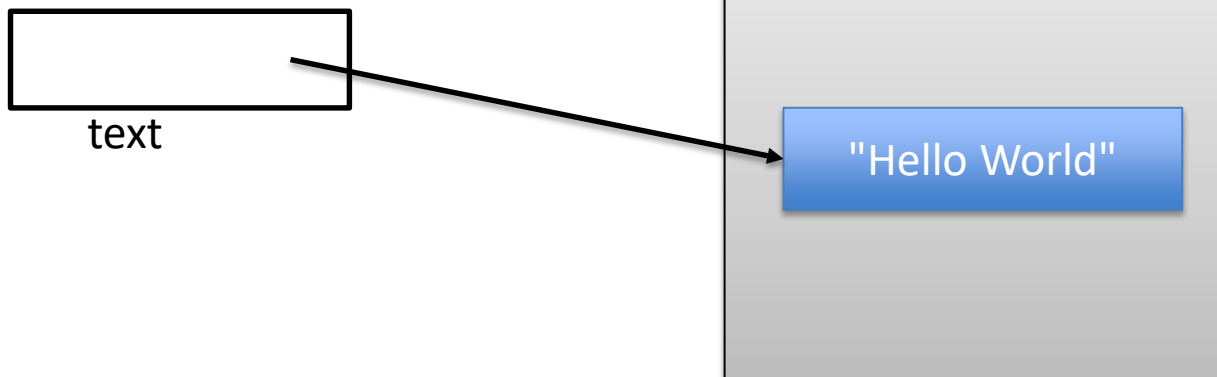
```
3 ▶ public class BereikVanVariabelen {  
4 ▶  public static void main(String[] args) {  
5     int x = 1;  
6     System.out.println("x = " + x + " en y = " + y);  
7     {  
8         int y = 5;  
9         System.out.println("x = " + x + " en y = " + y);  
10    }  
11    System.out.println("x = " + x + " en y = " + y);  
12    }  
13    System.out.println("x = " + x + " en y = " + y);  
14 }
```

## 3.3 De klasse String

- Behoort tot het pakket *java.lang* (wordt standaard geïmporteerd)

```
String text = "Hello World";
```

= objectreferentie



# 4. Input van de gebruiker

```
import java.util.Scanner;

public class Demo {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Geef je naam:");
        String naam = input.next();
        System.out.println("Welkom, " + naam);
        input.close();
    }
}
```

Object van de **Scanner** class aanmaken.

De Scanner class kan je vinden in de bibliotheek (library) **java.util**. Om de Scanner class te gebruiken, moet je deze toevoegen in je code door gebruik te maken van het keyword **import**.

Scanner object moet je steeds sluiten (close()) aan het einde van je programma.



## Opdracht 6: Begroeting

Vraag de gebruiker eerst om zijn voornaam en vervolgens naar zijn achternaam. Voeg beide namen samen, gescheiden door een spatie en druk de volledige naam af.

```
demo1.py x
1 first_name = input("Enter your first name:")
2 last_name = input("Enter your last name:")
3 full_name = first_name + " " + last_name
4 print("Hello, you are " + full_name)
5
```



```

import java.util.Scanner;

public class Demo {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Geef nummer: ");
        int number = scanner.nextInt();
        System.out.println("Geef je favoriete film:");
        String movie = scanner.nextLine();
        System.out.println("Geef je favoriete acteur:");
        String actor = scanner.next();
        System.out.println("Geef je favoriete band:");
        String band = scanner.nextLine();
        System.out.println(number + ", " + movie + ", "
            + actor + ", " + band);
    }
}

```

Geef nummer:

15

Geef je favoriete film:

Geef je favoriete acteur/actrice:

Brad Pitt

Geef je favoriete band:

15, , Brad, Pitt



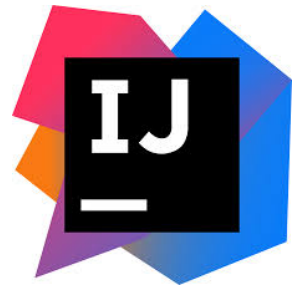


```
import java.util.Scanner;

public class Demo {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Geef nummer: ");
        int number = scanner.nextInt();
        scanner.nextLine();
        System.out.println("Geef je favoriete film:");
        String movie = scanner.nextLine();
        System.out.println("Geef je favoriete acteur:");
        String actor = scanner.nextLine();
        System.out.println("Geef je favoriete band:");
        String band = scanner.nextLine();
        System.out.println(number + ", " + movie + ", "
                           + actor + ", " + band);
    }
}
```



De plaats van de accolades is belangrijk! Zorg verder voor een nette inspruing van de accolades (= een code convention). Dit verhoogt de leesbaarheid van het programma.



**Code > Reformat code**  
**CTRL + ALT + L**



## Opdracht 7: Omrekening Celsius → Fahrenheit

Vraag de temperatuur in Celsius en reken deze temperatuur om naar Fahrenheit.



```
celsius = int(input("Geef de temperatuur in C: "))  
fahrenheit = 9 / 5 * celsius + 32  
print("De temperatuur in F is: ", round(fahrenheit, 1))
```

$$F = \frac{9}{5} C + 32$$

Celsius to Fahrenheit Formula

Geef de temperatuur in C: 26  
De temperatuur in F is: 78.8



```
import java.util.Scanner;
```

```
public class Demo {
```

```
    public static void main(String[] args) {
```

```
        Scanner input = new Scanner(System.in);
```

```
        System.out.println("Geef de temperatuur in C:");
```

```
        int celsius = input.nextInt();
```

```
        double fahrenheit = 9 / 5 * celsius + 32;
```

```
        fahrenheit = Math.round(fahrenheit * 10) / 10;
```

```
        System.out.println("F:" + fahrenheit);
```

```
        System.out.println();
```

```
        input.close();
```

```
    }
```

```
}
```

1 (omdat: int / int → int)



Geef temperatuur in C:

26

F: 58.0



```
import java.util.Scanner;

public class Demo {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Geef de temperatuur in C:");
        int celsius = input.nextInt();
        double fahrenheit = 9 / 5.0 * celsius + 32;
        fahrenheit = Math.round(fahrenheit * 10) / 10.0;
        System.out.println("F:" + fahrenheit);
        System.out.println();
        input.close();
    }
}
```



Geef de temperatuur in C:

26

F: 78.8



# 5. Operatoren

operator	Betekenis
++	Verhoog var met 1
--	Verlaag met 1
+=	Verhoog met gegeven getal
-=	Verlaag met gegeven getal
/=	Deel door het gegeven getal
*=	Vermenigvuldig variabele met het gegeven getal
%=	Rest bij deling door het gegeven getal

`i++;` // is een kortere notatie voor `i = i + 1;`

`i--;` // is een kortere notatie voor `i = i - 1;`

`i += 3;` // is een kortere notatie voor `i = i + 3;`

`i *= 2;` // is een kortere notatie voor `i = i * 2;`



## Opdracht 8: Controlegetal BBAN bepalen

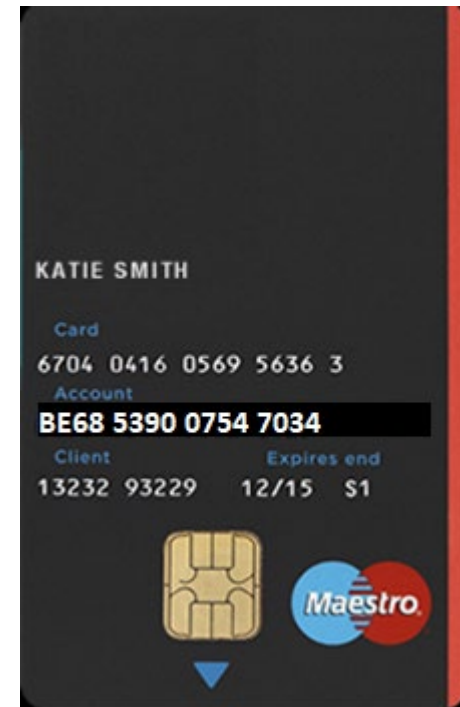
*Neem de 12 laatste cijfers van een bankrekeningnummer en sla deze waarde op in een variabele.*

*Druk de waarde af.*

*Vermenigvuldig dit getal met 1000000.*

*Tel hier 111400 bij op. Vervolgens bereken je de rest van het getal bij deling door 97.  
Het bekomen getal trek je af van 98.*

*Het getal dat je nu uitkomt is het 2-cijferig getal na BE in het bankrekeningnummer.*



# 6. Condities en logische operatoren

## 6.1 Condities

Vraag de leeftijd van de gebruiker. Indien de leeftijd groter of gelijk is aan 18 toon je: "Je bent volwassen" indien de leeftijd lager is dan 18 toon je "Je bent een kind".



```
leeftijd = int(input("Geef je leeftijd:"))  
if leeftijd >= 18:  
    print("Je bent volwassen")  
else:  
    print("Je bent een kind")
```



## 6.2 Relationale en logische operatoren

operator	Betekenis
>	groter dan
>=	groter of gelijk aan
<	kleiner dan
<=	kleiner of gelijk aan
==	gelijk aan
!=	verschillend van

Twee of meer relatiecondities kunnen gecombineerd worden m.b.v. logische operatoren

Logische operator	Betekenis	Resultaat
&&	and	true als beide condities true zijn
	or	true als 1 van beide condities true is
!	not	is het tegenovergestelde



```
import java.util.Scanner;

public class Leeftijd {
    public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);
        System.out.println("Geef je leeftijd:");
        int leeftijd = keyboard.nextInt();
        if (leeftijd >= 18) {
            System.out.println("Je bent volwassen");
        } else {
            System.out.println("Je bent een kind");
        }
        keyboard.close();
    }
}
```

() zijn verplicht



## if .. else if .. else

```
if (leeftijd >= 18) {  
    System.out.println("Je bent volwassen");  
} else if (leeftijd >= 10) {  
    System.out.println("Je bent een tiener");  
} else if (leeftijd >= 2) {  
    System.out.println("Je bent een kind");  
} else {  
    System.out.println("Je bent een baby");  
}
```



## Of geneste if ... else ...

```
if (leeftijd >= 18) {  
    System.out.println("Je bent volwassen");  
} else {  
    if (leeftijd >= 10) {  
        System.out.println("Je bent een tiener");  
    } else {  
        if (leeftijd >= 2) {  
            System.out.println("Je bent een kind");  
        } else {  
            System.out.println("Je bent een baby");  
        }  
    }  
}
```

## Opdracht 9:

Schrijf een programma dat toelaat de leeftijd en het aansluitingsjaar van een lid van een tennisclub in te voeren en dat op basis hiervan zijn bijdrage berekent en afdrukt. Volgende regels worden hierbij gerespecteerd:

- Basisbedrag: 100 euro
- Reductie: 14.5 euro indien het lid nog geen 21 is of ouder is dan 60
- Reductie: 2.5 euro per aangesloten jaar
- Minimum te betalen bedrag is 62.5 euro.



```

public class Demo {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Geef je naam:");
        String naam = input.next();
        System.out.println("Geef je naam nog een keer:");
        String naamNogEens = input.next();
        System.out.println("==" + (naam == naamNogEens));
        System.out.println("equals:" + naam.equals(naamNogEens));
        input.close();
    }
}

```

Geef je naam:

**Bert**

Geef je naam nog een keer:

**Bert**

==:false

equals:true

Als je de inhoud van 2 strings wil vergelijken, moet je de methode ***equals()*** gebruiken.



# 7. Iteraties

## 7.1 Iteraties: while-loop

*Lees een rij getallen in. Zolang het ingelezen getal kleiner is dan 10 druk je het getal samen met het dubbel van dit getal af.*

```
Scanner keyboard = new Scanner(System.in);
int getal = keyboard.nextInt();
while (getal < 10) {
    int dubbel = getal * 2;
    System.out.println("Het dubbel van " + getal + " is " + dubbel);
    getal = keyboard.nextInt();
}
keyboard.close();
```



## Opdracht 10: While-loop

Er wordt door renners een tijdrit gereden over een afstand van 36 km. Van iedere renner wordt het inschrijvingsnummer en zijn tijd in seconden gegeven. Deze gegevens dienen ingegeven te worden. De invoer stopt wanneer er voor het inschrijvingsnummer een negatief getal wordt ingegeven. We veronderstellen dat alle renners er een verschillende tijd over doen. Gevraagd:

- Welke renner is het snelst?
- Het percentage van de renners dat er langer dan 1 uur over doet,

Zorg voor volgende afdruk:

- Snelste renner is de renner met inschrijvingsnummer: .....
- Het percentage van de renners dat er langer dan 1 uur over doet: ...





## 7.2 Iteratives: for-loop



```
for count in range(1, 5):  
    print(count)
```

```
for (int count = 1; count < 5; count++) {  
    System.out.println(count);  
}
```

Output = ?



1  
2  
3  
4



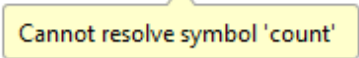


for (*initialisation*; *condition*; *increment*) {  
 statements;  
}



## Bereik van lus-variabele

Omdat de variabele `count` geïnitialiseerd is in de `for`-loop, is deze variabele enkel zichtbaar binnen de `for`-loop.

```
public class Demo3 {  
    public static void main(String[] args) {  
        for (int count = 10; count >= 0; count--) {  
            System.out.println(count);  
        }  
        System.out.println(count);  
    }  
}
```



## Opdracht 11: *For-loop*

1. Toon alle veelvouden van 10 die kleiner of gelijk zijn aan 100.
2. Schrijf een programma volgende schermafdruck te bekomen:

```
R
S T
U V W
X Y Z A
B C D E F
```

De grootte van de driehoek en de beginletter kunnen variëren en wordt als gegeven via het toetsenbord ingegeven. Indien je voorbij Z bent, moet je verdergaan vanaf A.

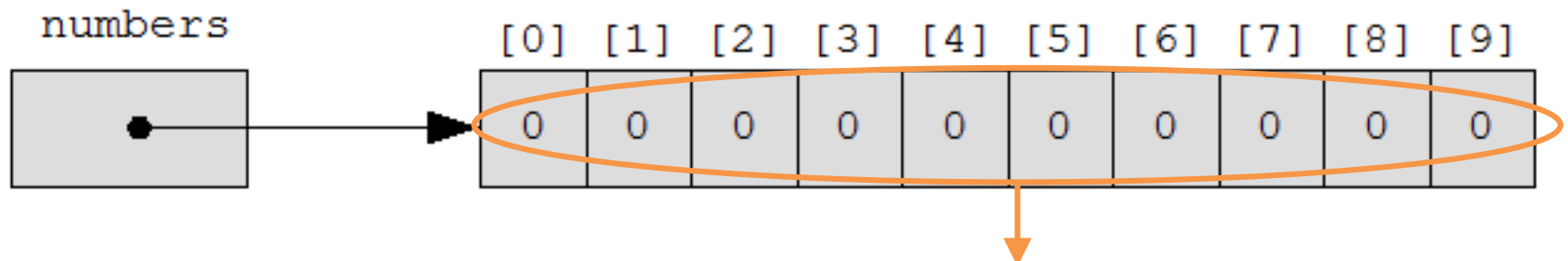
# 8. Arrays

array = verzameling van elementen van hetzelfde type

```
int[] numbers = new int[10];
```

Mag variabele zijn.

Arrays hebben een vaste lengte en kunnen nadien niet meer gewijzigd worden.



Wordt automatisch geïnitieerd op 0 (null voor referentietypen/false voor booleans).

## Declaratie en initialisatie :

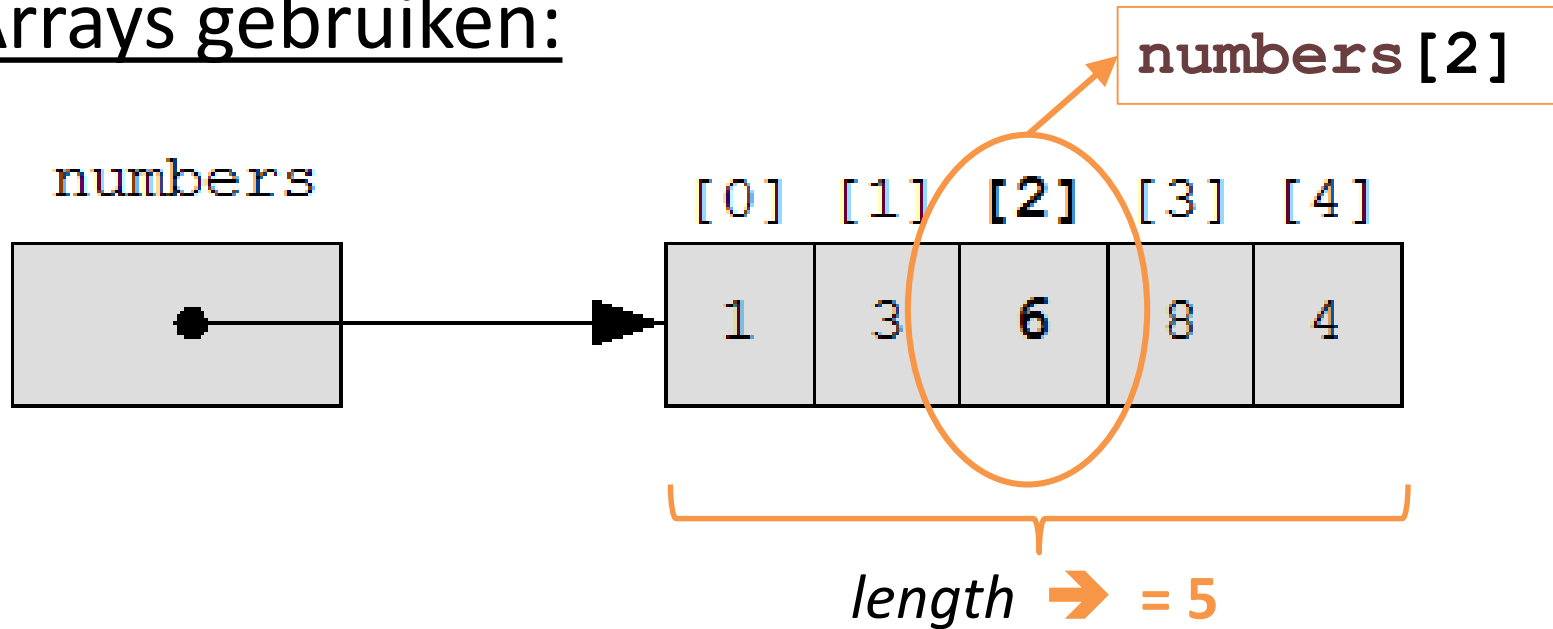
```
int[] numbers = {1, 3, 6, 8, 4};
```

||

```
int[] numbers;  
numbers = new int[] {1, 3, 6, 8, 4};
```

```
Scanner scanner = new Scanner(System.in);  
int[] numbers = new int[5];  
for (int i = 0; i < numbers.length; i++) {  
    System.out.println("Nr. " + (i + 1) + ":");  
    numbers[i] = scanner.nextInt();  
}
```

## Arrays gebruiken:



```
for (int i = 0; i < numbers.length; i++) {  
    System.out.println(numbers[i]);  
}
```

Waarom '<'?



## Een waarde aan een array-element toekennen:

```
numbers[2] = 12;  
numbers[4] = getal;  
numbers[5] = -4; ➔ java.lang.ArrayIndexOutOfBoundsException
```



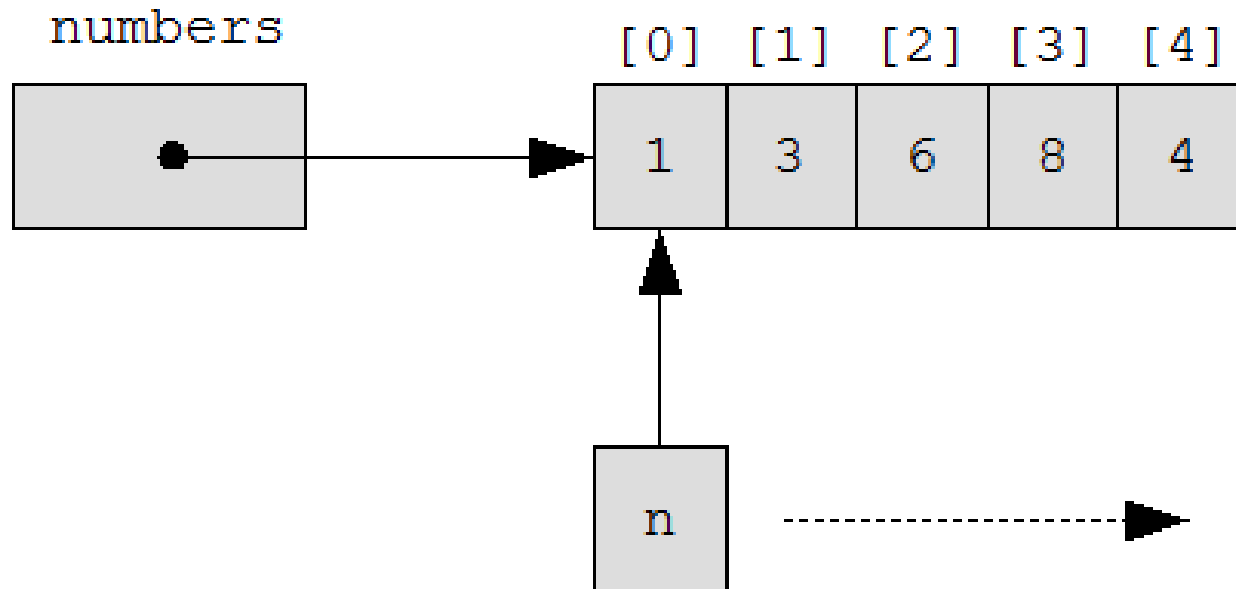
## For each-loop:

```
int[] numbers = { 1, 3, 6, 8, 4 };  
  
for (int n : numbers) {  
    System.out.println(n);  
}
```

De variabele *n* neemt tijdens de iteratie één voor één de waarde aan van de elementen uit de *array*.







Voordeel: korte notatie

Nadeel:

- Je beschikt niet over de indexwaarde.
- Enkel geschikt als je alle waarden van de tabel wil doorlopen.

## Opdracht 12: *Arrays gebruiken*

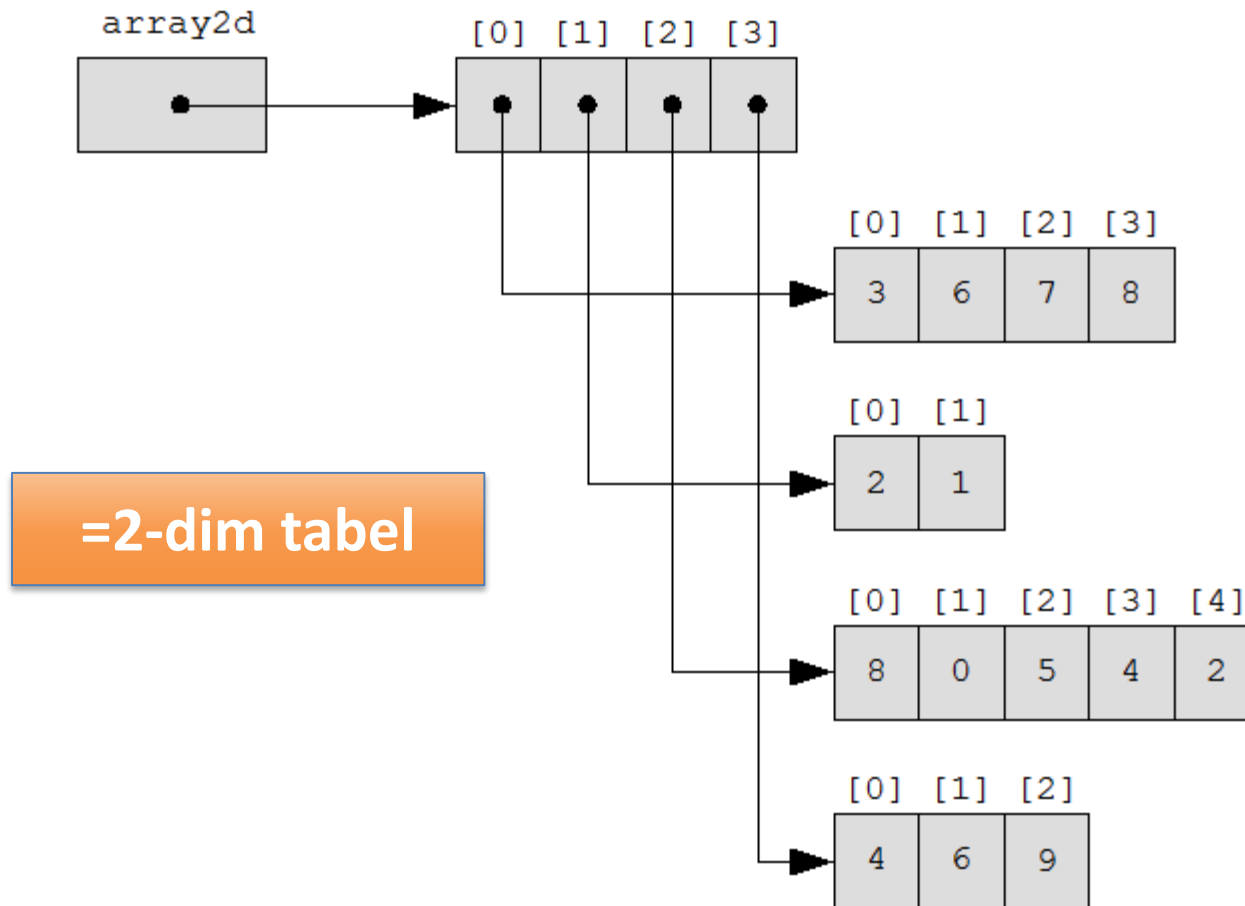
- Maak een *array* van 20 getallen en vul deze met veelvouden van 7.
- Druk de *array* in volgorde af op het scherm met een *for each*-lus.
- Maak vervolgens een tabel van lengte 10 waarin je de volgende waarden plaatst 1, 3, 5, 7, 9, 11, 13, 15, 17, 19.

Druk deze waarden af door gebruik te maken van de *for each*-lus.  
Druk daarna de waarden in omgekeerde volgorde af.



# 9. Arrays van arrays (= meerdimensionale array)

```
int[][] array2d = {{3, 6, 7, 8}, {2, 1},  
                  {8, 0, 5, 4, 2}, {4, 6, 9}};
```



Of: `int[][] array2d = new int[4][];` → Creatie van array van arrays

```
array2d[0] = new int[4];  
array2d[1] = new int[2];  
array2d[2] = new int[5];  
array2d[3] = new int[3];
```

Voor elk element uit de array wordt een array van integers gecreëerd.

```
array2d[0][0] = 3;  
array2d[0][1] = 6;  
array2d[0][2] = 7;  
array2d[0][3] = 8;  
array2d[1][0] = 2;  
array2d[1][1] = 1;  
array2d[2][0] = 8;  
array2d[2][1] = 0;  
array2d[2][2] = 5;  
array2d[2][3] = 4;  
array2d[2][4] = 2;  
array2d[3][0] = 4;
```

Elementen krijgen een waarde

```
array2d[3][1] = 6;  
array2d[3][2] = 9;
```

1 element

## Afdruk 2-dim. array:

```
for (int i = 0; i < array2d.length; i++) {  
    for (int j = 0; j < array2d[i].length; j++) {  
        System.out.print(array2d[i][j] + "\t");  
    }  
    System.out.println();  
}
```

Output?

## Of met *for each*:

```
for (int[] row : array2d)  
    for (int el : row) {  
        System.out.print(el + "\t");  
    }  
    System.out.println();  
}
```

3	6	7	8	
2	1			
8	0	5	4	2
4	6	9		



Indien de *arrays* allemaal dezelfde lengte hebben, kan je een 2-dim. tabel als volgt declareren:

```
int[][] table = new int[4][7];
```

= 4 rijen en 7 kolommen

Zowel de hoofd-*array* als de sub-*array* worden tegelijkertijd gecreëerd. De hoofd-*array* wordt ingevuld met referenties naar de sub-*arrays*.

## Opdracht 13: *2-dimensionale arrays*

Maak een tabel van 4 rijen en 6 kolommen. Vul de tabel met het product van het rijnummer en kolomnummer. Druk de matrix af op het scherm en gebruik hierbij een *for each*-lus.



# 10. String formatering

```
public class Demo {  
    public static void main(String[] args) {  
        System.out.printf("My name is: %s%n", "Joe");  
        System.out.printf("My age is: %d%n", 28);  
    }  
}
```

SPECIFIER	APPLIES TO	OUTPUT
%c	character	Unicode character
%d	integer (int, long)	integer
%f	floating point	decimal number
%n	none	Platform-specific line separator
%s	any type	String value
%S	any type	Upper-case variant van %s





```

public class Demo {
    public static void main(String[] args) {
        System.out.printf("|%20d|\n", 93);
        System.out.printf("|%-20d|\n", 93);
        System.out.printf("|%020d|\n", 93);
        System.out.printf("%.2f", 2.3568888);
    }
}

```

```

|                               93|
|93                               |
|00000000000000000000000093|
2,36

```



# String.format()

```
public class Demo {  
    public static void main(String[] args) {  
        String output = String.format("%s = %d", "Joe", 35);  
        System.out.println(output);  
    }  
}
```

