

# Java Development Kit

and some more goods inside!

A dark blue diagonal gradient bar that starts from the bottom left corner and extends towards the top right corner, covering the lower half of the slide.

# JDK en documentatie.

- ❑ Om **Java-programma's** te kunnen **uitvoeren**, **volstaat** het de **Java Runtime Environment [JRE]** te installeren.
- ❑ Als Java-ontwikkelaar volstaat dit echter niet. We moeten ook Java **code compileren**, **debuggen**, **documentatie** genereren e.d.
- Deze **extra devtools** zijn vervat in de **Java Development Kit [JDK]**.
- ❑ Daarnaast moeten we ook beschikken over de nodige documentatie (raadpleegbaar online | lokaal op pc).
  - De **Java API documentatie**.

# Development IDE.

Nu gaan we twee tekstverwerkers downloaden en installeren. Aan jullie de vrije keuze om te kiezen welke je wil gebruiken.

→ Eclipse

→ **IntelliJ**

# Verschillende commentaren.

## `/* comment */`

- ❑ Dit is de standaard commentaar zoals we die ook kennen in andere talen als C/C++
- ❑ Alles tussen `/*` en `*/` wordt genegeerd door de compiler. (gebruik vooral bij **langere commentaar**)

## `/** documentation */`

- ❑ Dit soort commentaar is bedoeld om achteraf automatisch documentatie te genereren met de JAVADOC-Tool.

## `// comment`

- ❑ Compiler negeert alles dat achter `//` komt tot aan het einde van de regel.

# Het pakket definiëren.

- ❏ Met de regel **package hello;**
  - we geven aan dat deze code tot een bepaald pakket behoort.
  - De code die bij elkaar hoort, wordt vaak in eenzelfde pakket gestopt.
  - Worden ook gebruikt om klassen een unieke naam te geven.
  - De volledige klassenaam is gelijk aan de pakketnaam plus de korte naam van een klasse.
- ❏ Om verwarring omtrent pakketnamen te voorkomen, gebruikt men unieke internet domeinnamen.

# De klasse definiëren.

- ❑ Java is een **objectgeoriënteerde** programmeertaal.
- ❑ In plaats van te werken met **procedures** (zoals **C**) werkt **Java met objecten van een bepaalde klasse**.
- ❑ Een **object** is eigenlijk een **verzameling** van **gegevens** en **methoden** om met die gegevens om te gaan.
- ❑ Klasse is een soort blauwdruk van een object.
- ❑ Later meer hierover (komt nog vaak aan bod!)

# De methode main()

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

- ❑ **public**
  - betekent dat deze methode voor publiek gebruik is.
- ❑ **static**
  - betekent dat deze methode gemeenschappelijk is voor alle objecten van deze klasse.
- ❑ **void**
  - de methode geeft na uitvoering geen waarde terug. (later meer hierover)
- ❑ **String[ ] args**
  - is de parameter die de functie al dan niet meekrijgt als hij wordt aangeroepen.
- ❑ **{ }**
  - instructies die worden uitgevoerd als de methode wordt opgeroepen komen terecht in deze scope

# En hoe toveren we tekst in onze console?

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

In deze regel wordt de methode `println()` aangeroepen van het object **out** dat behoort tot het object **System**.



Almost done!!

