**Leden**: Wouter Bollaert

Maarten Hoste

Stijn Kuppens

Tim Leenaers

Marco Tordoor

|  |  |
| --- | --- |
| **Academiejaar**: 2012 – 2013 | **Opleidingsonderdeel**:  Geïntegreerd softwareproject |
| **Opleiding**: Toegepaste Informatica | **Jaar**: 2 |
| **Groep**: 9 |  |

**CODE CONVENTIONS**

**Professoren**: Jan Willekens

Kerstin Nys

Erik Ulrichts

Marijke Sporen

# Waarom gebruiken we code conventies?

We hebben verschillende redenen om te werken met code conventies:

* 80% van de kosten van een softwareprogramma gaat naar de onderhoud ervan
* Code conventies maken het makkelijker om de code van een programma te begrijpen
* Makkelijker werken in groep

# Bestandsnamen

|  |  |
| --- | --- |
| File Type | Suffix |
| Java source | .java |
| Java bytecode | .class |

# Organisatie van bestanden

## Begincommentaar

Een bestand bestaat uit verschillende secties, gescheiden van elkaar door een witregel en een optionele regel commentaar die uitlegt wat de volgende sectie doet.

Elk bestand moet beginnen met commentaar waarin volgende informatie te vinden is:

* De naam van de klasse
* De versie
* De datum
* Rechten in verband met copyright

Commentaar wordt altijd als volgt geschreven:

/\*\* … \*/

## Package en import statements

De eerste regel na de begincommentaar is een package statement. Hierin staat in welk package gewerkt zal worden binnen de gebruikte klasse.

Hieronder komen de import statements. Hiermee kunnen we voorgeprogrammeerde java-klassen gebruiken.

## Klasse- en interfacedeclaraties

*Vorm van een klasse (na de begincommentaar en de package en import statements):*

* Variabelen:

1. Public variabelen
2. Protected variabelen
3. Private variabelen

* Constructor(s)
* Methoden
* Eindigen met commentaar, vb. /\*\* Einde bestand oef1.java \*/