Ontwerp van Informatiesystemen

Prof. Verelst

Q uickprinter
Koningstraat 13
2000 Antwerpen
www.quickprinter.be

Ontwerp van Informatiesystemen

Table of Contents

    - Subklasse erven methode over maar moeten ze wel schrijven
    - Abstracte klasse kan concrete methode bevatten→ niet andersom
- Interfaces
    - Een pure abstracte klasse
    - Enkel abstracte methode
    - Fungeert als template
    - Definieert welk methoden subklasse moeten bezitten
    - Overerving van meerdere interfaces kan wel
- Overloading
    - 2de voorbeeld van polymorfisme
    - Geld voor methode binnen dezelfde klasse
    - Methode met dezelfde naam, maar verschillende parameterlijst
        - Bv constructen met verschillende parameters (eens naam niet meegeven want niet nodig)
    - Courant gebruikt bij constructor
    - De signature moet verschillen
        - Parameters moet verschillen
        - Type van parameters moet verschillen
    - Handig → let op cohesie!

Statische Attributen en Methoden

- Attributen en methode van klasse zijn mogelijk
    - Bv hoeveel objecten per klasse
    - Methode/attribuut aanroepen op de klasse→ geen object voor nodig
    - Statisch attribuut→ lijkt op globale variabele pascal
        - Maar enkel bruikbaar in de klasse
- Waarom dan Public static void main(String[] args)
    - Public→ moet mainprogramma kunnen opstarten
    - Static→ moet nog geen object hebben
    - Void→ geen returnwaarde

Hoofdstuk 12: Patronen

Ontwerppatronen

- Oplossing voor veelvoorkomend ontwerpprobleem
- Kwaliteit empirisch bewezen
- Eigenschappen
    - Naam

    - Probleem
    - Oplossing
    - Gevolgen
- Basiswerk
    - Gamma→ 23 oplossingen
- Nieuwe patronen voortdurend voorgesteld
- Kennis van patronen is een essentieel onderdeel van de kennis van de programmeur

Voordeel

- Onderhoudbaarheid
- Hergebruikbaarheid
- Correctheid
    - Is al veel gebruikt en is dus correct
- Programmeertaalonafhankelijk
- Communicatie

Voorbeelden

- Creational: aanmaak van objecten
- Structural: objecten samenvoegen
- Behavioral: communicatie en gedrag van objecten

Singleton

- Probleem: 1 en slechts 1 object van een klasse aanmaken
    - Vb. Drie pc willen iets afprinten te gelijk→ printer drukt pagina's willekeurig af
- Oplossing→ singleton
    - 1ste aanroep: creëert object
    - 2de aanroep: krijgt 1ste object terug want er mag maar 1 object zijn
        - vb. Maar 1 printer

Facade

- Probleem: onoverzichtelijke verbanden tussen klassen
    - Vb. Software voor bankautomaat
        - Alle controles moeten uitgevoerd worden
- Oplossing
    - Facadeklasse: Extra klasse
    - Roept alle controles aan
        - Als er extra controles bijkomen→ geen verandering in main
        - Enkel extra aanroep in facadeklasse

Observer

- Objecten die moeten reageren bij statuswijziging aan een ander object
- Oplossing

- Niet elke seconde vragen of er een verandering is
- Wel programma een notify laten sturen als data veranderd
- Koppeling