

## Module 2 Softwaresystemen

# IAPConnect-4

DB26, 2014–2015

Martijn Verkleij  
Axel Vugts

28 januari 2015

# Inhoudsopgave

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Ontwerp</b>  | <b>4</b> |
| 1.1      | Klassediagrammen . . . . .  | 4        |
| 1.2      | Welke klasse implementeert welke requirements? . . . . .                    | 4        |
| 1.2.1    | Server . . . . .  | 4        |
| 1.2.2    | Client . . . . .  | 4        |
| 1.3      | MVC en het Observer-patroon . . . . .                                       | 5        |
| 1.4      | Protocollen en manieren van dataopslag . . . . .                            | 5        |
| <b>2</b> | <b>Klasse voor klasse</b>   | <b>6</b> |
| 2.1      | <geen> . . . . .  | 6        |
| 2.1.0.1  | IAPConnect4 . . . . .   | 6        |
| 2.2      | client . . . . .  | 6        |
| 2.2.1    | <geen> . . . . .  | 6        |
| 2.2.1.1  | Client(String playerName, InetAddress hostname, int port) . . . . .         | 7        |
| 2.2.1.2  | ClientController(ClientView cv) . . . . .                                   | 7        |
| 2.2.1.3  | ClientView(ClientController cc) . . . . .                                   | 8        |
| 2.2.1.4  | ServerHandler(Socket sockArg, Client clientArg, String playerArg) . . . . . | 8        |
| 2.2.2    | game . . . . .  | 8        |
| 2.2.2.1  | ComputerPlayer . . . . .  | 9        |
| 2.2.2.2  | RandomComputer . . . . .  | 9        |
| 2.2.2.3  | WinningComputer . . . . .   | 9        |
| 2.3      | core . . . . .  | 9        |
| 2.3.1    | <geen> . . . . .  | 9        |
| 2.3.1.1  | Config . . . . .  | 10       |
| 2.3.2    | game . . . . .  | 10       |
| 2.3.2.1  | Board . . . . .   | 10       |
| 2.3.2.2  | InvalidMoveException . . . . .  | 10       |
| 2.3.2.3  | Player . . . . .  | 11       |
| 2.3.3    | networking . . . . .  | 11       |
| 2.3.3.1  | Command . . . . .   | 11       |
| 2.3.3.2  | InvalidCommandException . . . . .   | 11       |
| 2.3.3.3  | Protocol . . . . .  | 12       |
| 2.3.3.4  | ProtocolError . . . . .   | 12       |
| 2.4      | menu . . . . .  | 12       |
| 2.4.1    | <geen> . . . . .  | 12       |
| 2.4.1.1  | MenuController . . . . .  | 12       |
| 2.4.1.2  | MenuView . . . . .  | 13       |
| 2.5      | server . . . . .  | 13       |
| 2.5.1    | <geen> . . . . .  | 13       |
| 2.5.1.1  | ClientHandler(Socket sockArg, Server serverArg) . . . . .                   | 13       |
| 2.5.1.2  | Server . . . . .  | 14       |

|          |                                 |           |
|----------|---------------------------------|-----------|
| 2.5.1.3  | ServerController . . . . .      | 14        |
| 2.5.1.4  | ServerPlayer . . . . .          | 15        |
| 2.5.1.5  | ServerView . . . . .            | 15        |
| <b>3</b> | <b>Tests</b>                    | <b>16</b> |
| 3.1      | Unit tests . . . . .            | 16        |
| 3.2      | System tests . . . . .          | 16        |
| 3.3      | Metrics . . . . .               | 16        |
| <b>4</b> | <b>Reflectie op de planning</b> | <b>17</b> |



# 1 Ontwerp

## 1.1 Klassediagrammen

### 5 1.2 Welke klasse implementeert welke requirements?

#### 1.2.1 Server

1. *"When the server is started, a port number should be entered that the server will listen to."*  
nl.utwente.iapc.IAPConnect4.menu.MenuController  
10 Via het menu kan de gebruiker een poort kiezen,
2. *"If the port number already is in use, an appropriate error message is returned, and a new port number can be entered."*  
In server.ServerController wordt gecontroleerd of de poort gebruikt kan worden. Als dit niet zo is wordt de gebruiker teruggebracht naar het hoofdmenu.
- 15 3. *"A server should be able to support multiple instances of the game that are played simultaneously by different clients."*  
nl.utwente.iapc.IAPConnect4.model.server.Server  
Elke client die verbinding maakt met de server, krijgt een eigen Thread middels een ClientHandler.
- 20 4. *"The TUI ensures that all communication messages are written to System.out."*  
Omdat wij een GUI hebben gebouwd, is deze requirement niet van toepassing.
5. *"The server should respect the protocol as defined for the tutorial group during the project session in Week 7, i.e., the server should be able to communicate with all other clients from the tutorial group."*  
25 nl.utwente.iapc.IAPConnect4.core.networking.Protocol  
In de Enum's core.networking.Protocol en core.networking.ProtocolError hebben we het protocol zoals vastgelegd in week 7 opgeslagen.

#### 1.2.2 Client

1. *"The client should have a user friendly TUI, which provides several options to the user (e.g., possibility to enter port number and IP address) to request a game at the server."*  
30 nl.utwente.iapc.IAPConnect4.menu.MenuView



De Swing GUI die wordt gemaakt is gebruiksvriendelijk en kan alle functies van het spel aansturen.

- 35 2. *"The client should support human players, and computer players with (some) artificially intelligent behaviour."*

`nl.utwente.iapc.IAPConnect4.client.ClientController`

Als een client gestart wordt, draait deze in HumanPlayer modus. Met een knop in de View kan eventueel een computerspeler het spel overnemen.

- 40 3. *The client provides a hint functionality. This shows a human player a possible move, as indicated by the computer player. The move may only be proposed, the human player should have the possibility to decide whether to play this move, or make a different one.*

4. *"After the game is finished, the player should be able to start a new game."*

45 `nl.utwente.iapc.IAPConnect4.client.ClientController`

Er wordt na het afronden van een spel een PopUp gemaakt waarin de gebruiker aan kan geven of er aan een nieuw spel deel mag worden genomen of niet.

- 50 5. *"If a player quits the game before it has finished, closes the UI, or the client crashes, the other players should be informed. In this case, the other players should be allowed to register again with the server to play the game."*

`nl.utwente.iapc.IAPConnect4.client.ClientController`

Zodra een speler onverwacht de verbinding verbreekt, stuurt de server een `game_end` commando naar de andere client. Deze zal op zijn beurt het spel afronden als gelijkspel en de gebruiker vragen wat nu te doen.

- 55 6. *"A server might at all times disconnect. The clients should react to this in a decent way, closing all open connections etc."*

`nl.utwente.iapc.IAPConnect4.client.ClientController`

Als de server onverwacht de verbinding sluit, wordt de game afgerond.

- 60 7. *"The client should respect the protocol as defined for the tutorial group during the project session in Week 7, i.e., the client should be able to communicate with all other servers from the tutorial group."*

`nl.utwente.iapc.IAPConnect4.core.networking.Protocol`

Ook in dit geval wordt er van dezelfde Enum's gebruik gemaakt, uit `core.networking`.

## 1.3 MVC en het Observer-patroon

- 65 Om de code overzichtelijk en flexibel te houden, hebben we bij het schrijven van het spel gebruik gemaakt van MVC. Elk scherm van de UI wordt opgebouwd door een View en aangestuurd door een Controller. Het model is hierbij verantwoordelijk voor de onderliggende logica. Ook gebruiken we het Observer-patroon om gebeurtenissen vanuit het model aan een controller door te geven.

## 70 1.4 Protocollen en manieren van dataopslag

Voor het spelen van een multiplayer-spel maken we gebruik van het protocol zoals dat opgesteld is in samenwerking met de rest van onze projectgroep. Er wordt in onze applicatie geen gebruik gemaakt van data-opslag.



## 2 Klasse voor klasse

75 per package:

### 2.1 <geen>

#### 2.1.0.1 IAPConnect4

|                               |   |
|-------------------------------|---|
| <b>Rol:</b>                   | Controller  |
| <b>Verantwoordelijkheden:</b> | De hoofdklasse van het spel, dit is een <code>Singleton</code> . Deze is verantwoordelijk voor het openen van het menu en vervolgens voor het starten van een client of een server. |
| <b>Gebruikte klassen:</b>     | <code>client.ClientController;</code><br><code>menu.MenuController;</code><br><code>server.ServerController;</code>   |
| <b>Speciale eisen:</b>        | geen  |
| <b>Voorzorgsmaatregelen:</b>  |   |

### 2.2 client

80 2.2.1 <geen>

## 2.2.1.1 Client(String playerName, InetAddress hostname, int port)

|                               |   |
|-------------------------------|---|
| <b>Rol:</b>                   | Model   |
| <b>Verantwoordelijkheden:</b> | Houdt de status van een client-verbinding bij middels een <code>ServerHandler</code> .  |
| <b>Gebruikte klassen:</b>     | <pre> client.ServerHandler; core.game.Board; core.networking.Command; core.networking.InvalidCommandException; core.networking.Protocol; </pre> |
| <b>Speciale eisen:</b>        | <pre> playerName != null; hostname != null; port != 0; </pre>   |
| <b>Voorzorgsmaatregelen:</b>  |   |



## 2.2.1.2 ClientController(ClientView cv)

|                               |   |
|-------------------------------|---|
| <b>Rol:</b>                   | Controller  |
| <b>Verantwoordelijkheden:</b> | Verbindt de ClientView met de Client, stuurt gebeurtenissen van de View naar het Model en andersom.   |
| <b>Gebruikte klassen:</b>     | <pre> client.ClientHandler; client.ClientView; client.Client; core.game.Board; core.networking.Command; core.networking.InvalidCommandException; core.networking.Protocol; </pre> |
| <b>Speciale eisen:</b>        | <pre> cv instanceof ClientView, cv != null </pre>   |
| <b>Voorzorgsmaatregelen:</b>  |   |

## 2 Klasse voor klasse

### 85 2.2.1.3 ClientView(ClientController cc)

|                               |  |
|-------------------------------|--|
| <b>Rol:</b>                   | View   |
| <b>Verantwoordelijkheden:</b> | Bouwt een Swing GUI die wordt aangedreven door de ClientController.  |
| <b>Gebruikte klassen:</b>     | <code>IAPConnect4;</code><br><code>client.ClientController;</code><br><code>core.networking.Command;</code><br><code>core.networking.InvalidCommandException;</code><br><code>core.networking.Protocol;</code> |
| <b>Speciale eisen:</b>        | <code>cc instanceof ClientController;</code><br><code>cc != null</code>  |
| <b>Voorzorgsmaatregelen:</b>  |  |

### 2.2.1.4 ServerHandler(Socket sockArg, Client clientArg, String playerArg)

|                               |  |
|-------------------------------|--|
| <b>Rol:</b>                   | Controller   |
| <b>Verantwoordelijkheden:</b> | Onderhoudt de verbinding met de Server, verwerkt commando's van en naar de server.   |
| <b>Gebruikte klassen:</b>     | <code>client.Client;</code><br><code>core.Config;</code><br><code>core.game.Board;</code><br><code>core.game.InvalidMoveException;</code><br><code>core.networking.Command;</code><br><code>core.networking.InvalidCommandException;</code><br><code>core.networking.Protocol;</code><br><code>core.networking.ProtocolError;</code> |
| <b>Speciale eisen:</b>        | <code>sockArg != null;</code><br><code>clientArg != null;</code><br><code>playerArg != null;</code>  |
| <b>Voorzorgsmaatregelen:</b>  |  |

## 2.2.2 game



## 90 2.2.2.1 ComputerPlayer

|                               |   |
|-------------------------------|---|
| <b>Rol:</b>                   | Model   |
| <b>Verantwoordelijkheden:</b> | Een abstracte klasse gemaakt als basis voor computerspelers.    |
| <b>Gebruikte klassen:</b>     | <code>core.game.Board;</code><br><code>core.game.Player;</code> |
| <b>Speciale eisen:</b>        | geen  |
| <b>Voorzorgsmaatregelen:</b>  |   |

## 2.2.2.2 RandomComputer

|                               |   |
|-------------------------------|---|
| <b>Rol:</b>                   | Model   |
| <b>Verantwoordelijkheden:</b> | Een computerspeler met een laag niveau: zal een volledig willekeurige Move bereken. |
| <b>Gebruikte klassen:</b>     | <code>client.game.ComputerPlayer;</code>  |
| <b>Speciale eisen:</b>        | geen  |
| <b>Voorzorgsmaatregelen:</b>  |   |

## 2.2.2.3 WinningComputer

|                               |   |
|-------------------------------|---|
| <b>Rol:</b>                   | Model   |
| <b>Verantwoordelijkheden:</b> | Een slimmere computerspeler. Deze zal met een iets betere strategie een minder-random Move berekenen. |
| <b>Gebruikte klassen:</b>     | <code>core.game.ComputerPlayer;</code>  |
| <b>Speciale eisen:</b>        | geen  |
| <b>Voorzorgsmaatregelen:</b>  |   |

## 2.3 core

## 2.3.1 &lt;geen&gt;

## 2 Klasse voor klasse

### 2.3.1.1 Config

|                               |   |
|-------------------------------|---|
| <b>Rol:</b>                   | Model   |
| <b>Verantwoordelijkheden:</b> | Bevat constanten voor hergebruik door de code |
| <b>Gebruikte klassen:</b>     | geen  |
| <b>Speciale eisen:</b>        | geen  |
| <b>Voorzorgsmaatregelen:</b>  |   |

### 100 2.3.2 game

#### 2.3.2.1 Board

|                               |   |
|-------------------------------|---|
| <b>Rol:</b>                   | Model   |
| <b>Verantwoordelijkheden:</b> | Model van een bord: bewaart de staat van alle cellen op een bord. Bevat ook spellogica voor het controleren van de geldigheid van zetten. |
| <b>Gebruikte klassen:</b>     | geen  |
| <b>Speciale eisen:</b>        | geen  |
| <b>Voorzorgsmaatregelen:</b>  |   |



#### 2.3.2.2 InvalidMoveException

|                               |   |
|-------------------------------|---|
| <b>Rol:</b>                   | Model   |
| <b>Verantwoordelijkheden:</b> | Een exceptie voor het aangeven van een ongeldige zet, bijvoorbeeld omdat de kolom op het bord al gevuld is. |
| <b>Gebruikte klassen:</b>     | <code>core.game.Player</code>   |
| <b>Speciale eisen:</b>        | geen  |
| <b>Voorzorgsmaatregelen:</b>  |   |

## 105 2.3.2.3 Player

|                               |  |
|-------------------------------|--|
| <b>Rol:</b>                   | Model  |
| <b>Verantwoordelijkheden:</b> | Interface voor het modelleren van verschillende soorten spelers, waaronder <code>ComputerPlayers</code> en <code>HumanPlayers</code> |
| <b>Gebruikte klassen:</b>     | geen   |
| <b>Speciale eisen:</b>        | geen   |
| <b>Voorzorgsmaatregelen:</b>  |  |

## 2.3.3 networking

## 2.3.3.1 Command

|                               |   |
|-------------------------------|---|
| <b>Rol:</b>                   | Model   |
| <b>Verantwoordelijkheden:</b> | Klasse die zowel door de <code>ClientHandler</code> als de <code>ServerHandler</code> wordt gebruikt om ontvangen en verzonden commando's te interpreteren. |
| <b>Gebruikte klassen:</b>     | <code>core.networking.Protocol</code>   |
| <b>Speciale eisen:</b>        | geen  |
| <b>Voorzorgsmaatregelen:</b>  | Moet geconstrueerd worden met minstens een string onderdeel van <code>Protocol.getAllValues()</code> en eventueel met enige Strings.                        |

## 110 2.3.3.2 InvalidCommandException

|                               |   |
|-------------------------------|---|
| <b>Rol:</b>                   | Model   |
| <b>Verantwoordelijkheden:</b> | Exceptie die door de <code>Server</code> en de <code>Client</code> wordt gebruikt als er een invalide commando wordt ontvangen. |
| <b>Gebruikte klassen:</b>     | <code>core.networking.Protocol</code>   |
| <b>Speciale eisen:</b>        | geen  |
| <b>Voorzorgsmaatregelen:</b>  |   |

## 2 Klasse voor klasse

### 2.3.3.3 Protocol

|                               |   |
|-------------------------------|---|
| <b>Rol:</b>                   | Model   |
| <b>Verantwoordelijkheden:</b> | Een Enum die de door onze projectgroep opgestelde protocolregels bevat. |
| <b>Gebruikte klassen:</b>     | geen  |
| <b>Speciale eisen:</b>        | geen  |
| <b>Voorzorgsmaatregelen:</b>  |   |

### 2.3.3.4 ProtocolError

|                               |  |
|-------------------------------|--|
| <b>Rol:</b>                   | Model  |
| <b>Verantwoordelijkheden:</b> | Een Enum die de door onze projectgroep opgestelde foutcodes bevat. |
| <b>Gebruikte klassen:</b>     | geen   |
| <b>Speciale eisen:</b>        | geen   |
| <b>Voorzorgsmaatregelen:</b>  |  |

115



## 2.4 menu

### 2.4.1 <geen>

#### 2.4.1.1 MenuController

|                               |   |
|-------------------------------|---|
| <b>Rol:</b>                   | Controller  |
| <b>Verantwoordelijkheden:</b> | Beheert de MenuView en verbindt deze met de IAPConnect4-klasse. |
| <b>Gebruikte klassen:</b>     | IAPConnect4<br>menu.MenuView                                    |
| <b>Speciale eisen:</b>        | pgeen   |
| <b>Voorzorgsmaatregelen:</b>  |   |

## 120 2.4.1.2 MenuView

|                               |   |
|-------------------------------|---|
| <b>Rol:</b>                   | View  |
| <b>Verantwoordelijkheden:</b> | Bouwt de Swing GUI voor het menu en geeft evenementen aan de MenuController door. |
| <b>Gebruikte klassen:</b>     | IAPConnect4<br>menu.MenuView  |
| <b>Speciale eisen:</b>        | geen  |
| <b>Voorzorgsmaatregelen:</b>  |   |

## 2.5 server

## 2.5.1 &lt;geen&gt;

## 2.5.1.1 ClientHandler(Socket sockArg, Server serverArg)

|                               |  |
|-------------------------------|--|
| <b>Rol:</b>                   | Controller   |
| <b>Verantwoordelijkheden:</b> | Onderhoudt de verbinding met de Server, verwerkt commando's van en naar de server.   |
| <b>Gebruikte klassen:</b>     | core.Config;<br>core.networking.Command;<br>core.networking.InvalidCommandException;<br>core.networking.Protocol;<br>core.networking.ProtocolError;<br>server.Server<br>server.ServerPlayer<br>server.Game |
| <b>Speciale eisen:</b>        | sockArg != null;<br>serverArg != null;   |
| <b>Voorzorgsmaatregelen:</b>  | sockArg != null;   |

125



## 2 Klasse voor klasse

### 2.5.1.2 Server

|                               |  |
|-------------------------------|--|
| <b>Rol:</b>                   | Model  |
| <b>Verantwoordelijkheden:</b> | Handelt nieuwe en huidige clients af door middel van <code>ClientHandler</code> , en stopt er in dien mogelijk twee in een nieuwe <code>Game</code> .  |
| <b>Gebruikte klassen:</b>     | <code>core.networking.Command</code> ;<br><code>core.networking.InvalidCommandException</code> ;<br><code>core.networking.Protocol</code> ;<br><code>server.ClientHandler</code> ;<br><code>server.Game</code> |
| <b>Speciale eisen:</b>        | geen   |
| <b>Voorzorgsmaatregelen:</b>  | Valide poort, binnen 0 en 65535.   |



### 2.5.1.3 ServerController

|                               |  |
|-------------------------------|--|
| <b>Rol:</b>                   | Controller   |
| <b>Verantwoordelijkheden:</b> | Verbindt de <code>ServerView</code> met de <code>Server</code> , stuurt gebeurtenissen van de View naar het Model en andersom. |
| <b>Gebruikte klassen:</b>     | <code>IAPConnect4</code><br><code>server.Server</code><br><code>server.ServerView</code>                                       |
| <b>Speciale eisen:</b>        | geen   |
| <b>Voorzorgsmaatregelen:</b>  |  |

## 130 2.5.1.4 ServerPlayer

|                               |   |
|-------------------------------|---|
| <b>Rol:</b>                   | Model   |
| <b>Verantwoordelijkheden:</b> | Een netwerkspeeler die om een zet gevraagd kan worden, en een zet van een client ontvangt.  |
| <b>Gebruikte klassen:</b>     | <code>core.game.InvalidMoveException;</code><br><code>core.game.Player;</code><br><code>core.networking.Command;</code><br><code>core.networking.Protocol;</code> |
| <b>Speciale eisen:</b>        | geen  |
| <b>Voorzorgsmaatregelen:</b>  |   |

## 2.5.1.5 ServerView

|                               |   |
|-------------------------------|---|
| <b>Rol:</b>                   | View  |
| <b>Verantwoordelijkheden:</b> | Bouwt een Swing GUI die wordt aangedreven door de <code>ServerController</code> . Biedt de gebruiker de mogelijkheid om de server te stoppen. |
| <b>Gebruikte klassen:</b>     | geen  |
| <b>Speciale eisen:</b>        | <code>sc instanceof ServerController;</code><br><code>sc != null</code>   |
| <b>Voorzorgsmaatregelen:</b>  |   |



## 3 Tests

### 135 3.1 Unit tests

We hebben tests geschreven voor Board en voor Server.

Board is getest met een testklasse die testte of `invalidMove()` en `isLegalMove()` werkt.

Server is een systeemtest voor geschreven die een heel simpel spel speelt op de server. Deze is uitgereider beschreven in System Tests.

### 140 3.2 System tests

Om de server te testen hadden wij een J-unit-test geschreven die alles voor het inloggen en het spelen van een heel simpel spel tot het valideren van de `game_end`-message testte. Dit heeft ons geholpen bij het repareren van regressies tijdens het implementeren van nieuwe features en bugfixes.

### 145 3.3 Metrics

Binnen Metrics bleek dat onze code het meest complex was in Server, ClientHandler en Command. Dit betekende dat deze klassen zijn beschreven met Javadoc en JML-statements.





## 4 Reflectie op de planning

150 Nu we het project hebben afgerond kunnen we terugkijken op onze oorspronkelijke planning. Wat gelijk opvalt is dat we in het begin een stuk minder voortgang boekent dan geplanned. Deels kwam dit door het niet besteden van de geplande tijd aan het project. Maar ook opstartproblemen gooiden roet in onze planning.

Hieruit leren we dat we onze planning eigenlijk te vroeg hebben opgesteld: al toen het project op gang begon te komen, bleek dat onze planning niet realistisch zou zijn.

