# Beschrijving van JabberPoint-code (versie Gerwin en Ewoud)

*(de werking v/h programma moet duidelijk worden nav deze beschrijving…)*

## Start van de applicatie

Het programma JabberPoint start met (in de main-methode in class JabberPoint) het aanmaken van hoofdobject Presenter via de PresenterFactory. Deze factory bepaalt welk concrete class gebruikt wordt als Presenter; de applicatie zelf weet dat niet, die kent alleen de interface ‘Presenter’.

Nadat de Presenter aangemaakt is, wordt er een SlideViewerFrame aangemaakt, waarin (via een SlideViewerComponent) een presentatie slide getoond kan worden. Deze SlideViewerFrame meldt zich aan bij de Presenter als PresentationObserver, waardoor deze bij het switchen naar een ander presentatie de juiste titel kan tonen. De SlideViewerComponent meldt zich ook bij de Presenter aan, maar dan als SlideObserver; de SlideViewerComponent moet namelijk bij een slide-switch de nieuwe slide tonen.

Zodra het frame (en component) aangemaakt zijn, wordt vanuit de main-methode een demo presentatie aangemaakt of een presentatie vanuit een bestand ingelezen (als er een bestandsnaam ingesteld is als programma argument). Van deze presentatie wordt de eerste slide geselecteerd.

De main heeft zijn werkt gedaan; wat rest is een Presenter-object die een huidige/actieve presentatie bevat, en een view die de huidige slide daarvan toont.

## Controllers

Omdat de applicatie natuurlijk door een gebruiker aangestuurd moet kunnen worden, zijn er een tweetal controllers op het SlideViewerFrame actief, een KeyController en een MenuController. De KeyController handelt specifieke keyboard toetsen af, zoals PageUp/PageDown voor respectievelijk vorige en volgende slide van de huidige presentatie. De MenuController handelt de menuitems af, waaronder het selecteren van een andere presentatie, het inlezen van een nieuwe presentaties en het springen naar een specifieke slide (via een ingetoetst slide nummer). Beide controllers communiceren alleen met het Presenter object. De Presenter zal presentatie-handelingen (zoals next/previous slide) doorsluisen naar de actieve presentatie.

## Inlezen van presentaties

Het inlezen van presentaties gaat via de Presenter. Momenteel kan er een (hardcoded) demo presentatie aangemaakt worden en kunnen er presentaties via een xml-bestand worden ingelezen. Het inlezen van presentaties via een xml-bestand verloopt via een ReaderFactory. Deze factory bepaalt aan de hand van de extensie (bv “.xml”) welke concrete reader wordt ingezet voor het laden van een presentatie. De enige opties zijn momemtel een DemoReader en een XMLReader. Deze laatste reader maakt een lijst met Slide-objecten aan, en maakt vervolgens presentaties aan die elk verwijzen naar de aangemaakte Slide-objecten. Op deze manier kunnen er meerdere presentaties (variaties) gemaakt worden die gebruik maken van dezelfde slide, waarbij elke presentatie een andere slide-volgorde kan toepassen, of slides kan overslaan of herhalen. Er wordt vanuit elke presentatie dus verwezen naar de aangemaakte slide-objecten; deze worden dus niet gedupliceerd.

Na net inlezen van nieuwe presentaties wordt de eerste presentatie geselecteerd. De Presenter zal bij elke presentatie-switch een notificatie uitsturen naar alle PresentationObservers, waaronder de eerder genoemde SlideViewerFrame. Verder zal de Presenter bij elke presentatie-switch de eerste slide van deze presentatie selecteren; hierbij wordt een notificatie uitgestuurd naar alle SlideObservers, waaronder de eerder genoemde SlideViewerComponent. Informatie over de nieuwe presentatie en (eerste) slide zal dus meteen door de views worden getoond.

## Tonen van presentatie slide

Het tonen van een presentatie is niets anders dan het tonen van de huidige slide, en het tonen van een slide is op zijn beurt niets anders dan het tonen van al zijn slide-items (de titel van de slide wordt als eerste slide-item getoond, waardoor deze steeds bovenaan op de slide staat). Bij het tekenen van een slide-item wordt een schaalfactor meegegeven; deze factor is de verhouding van de huidige window en de oorspronkelijk grootte. Als de applicatie-window door de gebruiker verkleind/vergroot wordt, dan zullen de items met dezelfde verhouding getekend worden.

De items worden altijd onder elkaar getoond, waarbij de verticale afstand door de hoogte van een “boudingbox” wordt bepaald. Elk specifieke item (tekst, plaatje, …) kan deze boundingbox bepalen. De hoogte is niet alleen afhankelijk van datgene wat getoond wordt (tekst, plaatje, …), maar ook van de ingestelde stijl. Elk item bevat een stijl dat gebruikt wordt bij het tonen, waardoor o.a. de (horizontale) inspringing en (verticale) regelafstand ingesteld kan worden.