Rik van Rijn

2-4-2023

Object georienteerd ontwerpen

Inhoud

[Refactoring strategie 2](#_Toc157631192)

[Composing Methods 2](#_Toc157631193)

[Dealing with generalization 2](#_Toc157631194)

[Moving features between objects 2](#_Toc157631195)

[Organizing data 2](#_Toc157631196)

[Overzicht van alle wijzigingen 4](#_Toc157631197)

[Overzicht van wijzigingen voor herkansing 6](#_Toc157631198)

[Architecturale problemen binnen Jabberpoint 7](#_Toc157631199)

[Klassen Diagram voor refactoring 9](#_Toc157631200)

[Klassen Diagram na refactoring 9](#_Toc157631201)

# Refactoring strategie

Voor het refactoren van code zijn een aantal technieken. Tijdens het refactoren van JabberPoint is er gebruik gemaakt van meerdere van deze technieken.

## Composing Methods

De techniek composing methods is bedoelt voor het herstructureren van methodes. Hier is gebruik van gemaakt in de SlideViewerComponent klasse. Hier is de Paintcomponent() opgesplitst in twee verschillende methodes. Dit heeft als reden dat de methode anders te groot zal zijn. Er is als basislijn aangehouden dat een methode meestal niet langer dan 10 regels moet zijn.

In de MenuController heeft hetzelfde plaats gevonden. De MenuController constructor was dusdanig groot dat niet goed leesbaar was wat plaats vond in de code. Er is gekozen om de verschillende opties die aan de dropdown knoppen allemaal in een apparte methode te plaatsen. Op deze manier is het makkelijker in te zien wat waar gebeurd. In de constructor is de code beter te lezen. Ook zorgt het opsplitsen in meerdere methoden er voor dat wanneer er iets aangepast moet worden dit snel gevonden kan worden. Het nadeel is echt wel dat wanneer het menu groter word er steeds meer functies bijkomen.

Ook zijn er op verschillende plaatsen onnodige variabelen weggehaald om zodoende de methode overzichtelijker te maken.

## Dealing with generalization

Bij de methode dealing with generalization gaat het vooral om de abstracte klassen. Er is gebruik gemaakt van deze methode door de abstracte klasse Accesor weg te halen. De abstracte klasse werd in de originele code geïmporteerd door de DemoPresentation en de XMLAccesor. Deze twee klassen dienen voor twee verschillende dingen. De DemoPresentation heeft als enigste functionaliteit om een demo presentatie aan te maken bij het opstarten en de XMLAccesor dient voor het maken en lezen van XML bestanden. Omdat deze twee klassen zoveel van elkaar verschillen is er gekozen de abstracte klasse te verwijderen.

## Moving features between objects

Wanneer een methode eigenlijk bij een andere klasse hoort is dit onderdeel van moving features. Er is gekozen om bij de DemoPresentation klasse alle methoden te verwijderen behalve de loadFile(). Dit heeft als reden dat de andere methoden niks toevoegden aan het programma. Er is vervolgens gekozen om de loadFile functie te hernoemen naar createDemoPresentation en deze verplaatsen naar de Presentation klasse. Dit heeft als reden dat de DemoPresentation bestond uit een enkele methode en beter bij de presentation past. Een andere mogelijkheid was geweest om de demo presentatie op te slaan in xml en deze in te lezen via de XML parser. Hier is niet voor gekozen omdat dit er voor zal zorgen dat de mogelijkheid ontstaat dat de werking van het programma niet correct is wanneer het bestand niet bijgeleverd is.

Een ander onderdeel van moving features between objects is het opsplitsen van een klasse. Zo is er gekozen om de Style klasse op te splitsen in een style en styles klasse. Dit heeft als reden dat de style klasse eerst een array bevatte met daarin instanties van zichzelf. Op deze manier wordt het makkelijker nieuwe styles toe te voegen.

## Organizing data

Organizing data is het organiseren van data in het programma. Hieronder valt dubbele data verwijderen. Zo is er gekozen om de SlideViewerFrame HEIGHT en WIDTH weg te halen en exclusief de HEIGHT en WIDTH van de Slide objecten te gebruiken. Er is gekozen om de HEIGHT en WIDTH uit de SlideViewerFrame weg te halen omdat het mogelijk is in de toekomst dia’s van verschillende grote’s te kunnen hebben.

# Overzicht van alle wijzigingen

Alle uitgevoerde refactoring is hieronder in een overzicht te zien.

AboutBox:

* Nieuwe velden gecreëerd zodat show functie duidelijker is
* Laten bestaan en niet in MenuController gezet ivm uitbreidbaarheid

BitMapItem:

* Hernoemen variabelen g naar graphics voor verbeterde leesbaarheid van de code.
* Niet gebruikte constructor verwijderd.

KeyController:

* Hernoemen variabel p naar presentation

MenuController:

* Verwijderen Presentatie en Frame veld
* Nieuw functies gemaakt per knop functionaliteit

Presentation:

* createDemoPresentation toegevoegd

Slide:

* verwijderen van getSlideItem
* variabel namen hernoemt

SlideItem:

* overbodige constructor verwijderd
* variabelnamen hernoemd

SlideViewerComponent:

* variabelnamen hernoemd
* Paintcomponent() opgesplitst naar PaintComponent() en setGraphics()

SlideViewerFrame:

* Variabele SERIALVERSIONUID hernoemd
* Variabelen width en height weggehaald
* setUpWindows aangepast en gebruikt nu Slide width en height

Style:

* Opgesplitst in twee verschillende klasses: Style en Styles

TextItem:

* Verwijderen van constructor
* Verwijderen EMPTYTEXT variabele
* Variabelnamen aangepast

XMLAccesor:

* Hernoemd naar XMLParser
* Toevoegen loadSlides functie om loadFile verder op te splitsen
* Stoppen met importeren Accesor

DemoPresentation klasse verwijderd omdat deze overbodig was.

Accesor abstracte klasse verwijderd omdat deze ook overbodig was.

# Overzicht van wijzigingen voor herkansing

Alle uitgevoerde refactoring voor de herkansing is hier uiteengezet:

BitmapItem:

* IMAGENAME final gemaakt omdat deze na het aanmaken niet van de bitmap niet meer gewijzigd hoeft te worden.

Presentation:

* createDemoPresentation functie aangepast zodat deze niet meer een instantie van zichzelf nodig heeft.

Slide:

* append methode verwijderd. De append methode die alleen fungeerde voor het appenden van een textitem is verwijderd zodat in de toekomst wanneer er nieuwe slideitemtypes zijn er niet telkens een nieuwe append functie aangemaakt hoeft te worden.

MenuController:

* goToSlide functie aangepast zodat je alleen nog kan navigeren naar een slide die bestaat. Wanneer men een slide kiest die niet bestaat komt er een pop-up in beeld.
* Methodes public gemaakt in plaats van private zodat deze beter testbaar zijn.

# Architecturale problemen binnen Jabberpoint

|  |  |  |
| --- | --- | --- |
| Probleem: | Oplossing: | Reden: |
| Menucontroller is een erg grote functie. | Methode opsplitsen in meerdere kleine methodes. | De code is slecht leesbaar en daardoor moeilijker te begrijpen. Ook is de methode moeilijk te testen. |
| Aboutbox is moeilijk te lezen. | Variabelen aangemaakt van de verschillende velden in de aboutbox. | De aboutbox code is moeilijk te lezen. Dit maakt het moeilijker deze te wijzigen in het geval dat deze geüpdatete moet worden. |
| Demopresentation klasse is een onlogische klasse. | Verwijderen van de demopresentation klasse en het herschrijven en verplaatsen van de functie createdemopresentation naar de presentation klasse. | Demopresentation klasse is een functie die behoort tot de presentation klasse. |
| Accesor klasse biedt geen extra functionaliteit. | Verwijderen van de klasse. | De accesor klasse biedt geen extra functionaliteit voor het huidige systeem. Het maakt het begrijpen van de code alleen maar moeilijker. |
| Code is slecht leesbaar. | Namen wijzigen van variabalen, niet gebruikte variabelen verwijderen en niet gebruikte functies en constructoren verwijderen. | De code is moeilijker te begrijpen wanneer er veel niet gebruikte en onlogische variabelnamen en niet gebruikte functies en constructoren in staan. |
| Klasse namen zijn onduidelijk. | Xmlaccesor hernoemen naar XMLparser. | Door de klasse een andere naam te geven geeft deze duidelijker aan waar deze klasse voor dient. Accesor zou betekenen dat hij alleen XML kan lezen terwijl parser aangeeft dat de klasse kan lezen en schrijven. |
| Style klasse heeft een standaard style. | Standaard style verplaatsen naar eigen klasse. Eventueel kan deze in de toekomst nog in de XML worden verwerkt. | Door de standaard style naar een eigen locatie te verplaatsen is de makkelijker te vinden en sneller aan te passen. |
| Append methode in de slide klasse is onhandig. | Append methode uit de slide klasse verwijderen. | De append methode is uit de slide klasse verwijdert omdat deze het moeilijker maakt om nieuwe slide items toe te voegen aan een slide wanneer er een nieuw type slide item aangemaakt word zou het met de huidige append methode telkens een extra append methode gemaakt moeten worden. |
| Go to slide gaat naar alle pagina’s ook als deze niet bestaat. | goToSlide functie aangepast zodat deze nu checkt of de target dia bestaat. | De gebruiker kon naar dia’s navigeren die niet bestaan. Dit is niet de bedoeling. |
| Weinig documentatie in de code | Extra documentatie in de code toegevoegd | Door meer documentatie toe te voegen kan het programma makkelijker gelezen worden en begrepen door een programmeur bij eventuele uitbreiding. |
| Dia navigatie is onlogisch. | Pagina omhoog en pagina omlaag gekoppelde functies omgewisseld. | Het is onlogisch om bij het pijltje omhoog een dia terug te gaan en bij een pijltje omlaag een dia verder. |

# Klassen Diagram voor refactoring

Afbeelding met diagram, schematisch

Automatisch gegenereerde beschrijving

# Klassen Diagram na refactoring

