

OGO Veranderingen

Ik heb ergens een verandering gemaakt waardoor new het niet meer doet.

AboutBox

Geen veranderingen gemaakt.

Accessor

Ongebruikt field weggehaald.

BitMapItem

FILE en NOTFOUND zijn protected.

Verandert naar private.

<https://refactoring.guru/self-encapsulate-field>

Getters en setters gemaakt voor bufferedImage en imageName

In de setter van imageName een check gedaan om te kijken of de string niet null of leeg is.

In de setter van bufferedImage een null check gedaan.

En de getters en setters toepassen in alle documenten

getBoundingBox tussenstaple variabele gemaakt, zodat er betere leesbaarheid is.

Draw niet leesbaar, alle parameters in tussentijdse variabele gezet.

This. Toegevoegd aan method call. In method toString

DemoPresentation

Alle slides worden nu in de loadFile method gemaakt, dit is niet de verantwoordelijkheid van 1 methode. Dus ik ga de methode opsplitsen in meerdere methodes.

<https://refactoring.guru/extract-method>

Methode voor het maken van slide 1,

Methode voor het maken van slide 2.

Methode voor het maken van slide 3.

<https://refactoring.guru/split-temporary-variable>

De variabele slide wordt nu de hele tijd override voor alle slides een nieuwe variabele maken voor elke slide.

Slides worden nu in eigen methode gemaakt. Deze methodes returned die specifieke slide die net gemaakt is. Elke methode heeft eigen locale variabele die gereturned wordt.

JabberPoint

Namen van de fields veranderd zodat ze duidelijker zijn.

KeyController

<https://refactoring.guru/self-encapsulate-field>

Getters en setter gemaakt voor presentation

Null check in de setter gedaan. En getter en setter toegepast.

keyPressed method:

- Switch case aangepast op aanraden van intelij.

MenuController

De onderstaande methodes een eigen methode gemaakt, zodat er niet methodes aangemaakt worden in de constructor. Maar dat iedere methode een eigen verantwoordelijkheid heeft.

- Method gemaakt wat er gebeurt als iemand drukt op open, eigen methode in menucontroller
- Method gemaakt wat er gebeurt als iemand op new drukt, eigen methode in menucontroller
- Method gemaakt wat er gebeurt als iemand op save drukt, eigen methode in menucontroller
- Method gemaakt wat er gebeurt als iemand op exit drukt, eigen methode in menucontroller
- Method gemaakt wat er gebeurt als iemand op next drukt, eigen methode in menucontroller
- Method gemaakt wat er gebeurt als iemand op previous drukt, eigen methode in menucontroller
- Method gemaakt wat er gebeurt als iemand op go to page drukt, eigen methode in menucontroller
- Method gemaakt wat er gebeurt als iemand op about drukt, eigen methode in menucontroller

Method gemaakt waar het menu wordt gemaakt, hierin worden de methodes makeFiletab, makeViewTab en makeHelpTab worden aangeroepen.

Dan om dubbele code te voorkomen de methode addToMenu gemaakt. Hierdoor kun je een nieuw onderdeel aan het menu toevoegen, door de parameters kunnen de naam en het menu verschillen.

<https://refactoring.guru/self-encapsulate-field>

Getter en setter gemaakt voor de fields parent en presentation

Null checks toegevoegd op de setters.

Alle static final fields private gemaakt en paar nieuwe namen gegeven die iets duidelijker zijn.

menultem een field gemaakt, omdat hij gebruikt wordt in meerdere methods.

In de methode goToPage de check gedaan of het ingevulde nummer niet meer is dan het aantal slides van de presentatie.

Factory gemaakt om een menultem te maken.

En ook in MenuControllerFactory een methode gemaakt om een MenuShortCut te maken

Presentation

Added this. Op een voor een paar plekken waar de methodes worden aangeroepen

<https://refactoring.guru/self-encapsulate-field>

Alle getters and setters aangemaakt, en toegepast.

En clear method public gemaakt.

Slide

Naam van de append methode gewijzigd.

SlideItem

Geen veranderingen in SlideItem gemaakt.

SlideViewerComponent

Repaint wordt in update sowieso gedaan of je door de if statement komt ja of nee dus dat kan ook boven de if statement komen te staan. Boven de if statement repaint() gezet, zodat er geen dubbele code daar is.

Parameter namen duidelijker gemaakt.

Removed = null from fields was overbodig.

<https://refactoring.guru/self-encapsulate-field>

Getters en setters toegevoegd. En null checks in de setters gedaan.

Presentation als field verwijderd, zodat er geen loop van dependencies meer is. Hiervoor in de plaats de fields slideCount en CurrentSlideNumber

De constante namen veranderd.

SlideViewerFrame

Geen veranderingen in SlideViewerFrame

Style

Een klasse gemaakt waar de styles array wordt gemaakt, met daarin de 5 styles. Dit is verplaatst naar eigen klasse omdat het niet de verantwoordelijkheid van de style is om een array van styles te maken. Is een statische methode zodat de hele package erbij kan.

<https://refactoring.guru/encapsulate-field>

Int indent private gemaakt, ook een getter en setter voor gemaakt, zodat de andere klasse niet zomaar aan het veld kunnen komen, alleen door een getter of setter.

Ook voor Color color, Font font, int fontSize en int leading

Moved getStyle to Stylemaker.java is nu de verantwoordelijkheid van dit document, hier staat namelijk de array in die getStyle gebruikt.

<https://refactoring.guru/self-encapsulate-field>

getters en setters gemaakt voor alle fields null checks gedaan.

StyleMaker

Aangemaakt om style array te maken

Dit was niet de verantwoordelijkheid van Style dus nieuwe klasse aangemaakt.

Methodes gemaakt zodat iedere methode maar 1 verantwoordelijkheid heeft.

- Een method gemaakt om een styleArray te maken
- En een methode gemaakt om een nieuwe style te maken.

TextItem

<https://refactoring.guru/self-encapsulate-field>

Voor tekst een setter gemaakt. En de setter toegepast in constructor

Een field gemaakt voor layouts. Hier ook een getter en setter voor gemaakt.

In de methode getBoundingBox een aantal veranderingen gedaan:

- De parameter g vernoemd naar graphics: dit is duidelijker.
- De while loop vervangen voor een for loop.
- De layouts arraylist wordt geset in de getLayouts methode
- Ik heb xsize en ysize op verschillende regels aangemaakt. Dit voor verbeterde leesbaarheid.
- Een tijdelijke variable aangemaakt voor de breedte en de hoogte. Deze waardes worden 2x gebruikt dus dan kan er een variable van gemaakt worden, dit verbeterd de leesbaarheid van de code ook.

In de methode draw een aantal veranderingen gedaan:

- Tussentijdse variable voor x van het punt en y van het punt gemaakt. Dit verbetert de leesbaarheid van de code
- De layouts arraylist wordt geset in de getLayouts methode
- Omdat deze variabele er nu zijn, hoeft het punt niet meer aangemaakt te worden en kun je in de for loop gewoon deze tussentijdse variabelen aanroepen.
- De while loop is een for loop geworden: op aanraden van intellij

In de methode getLayouts een aantal veranderingen gemaakt:

- Het field layouts een nieuwe arraylist maken.
- De parameters een duidelijkere naam gegeven, voor de leesbaarheid.

<https://refactoring.guru/extract-variable>

XMLAccessor

Een stuk code verplaatst naar een eigen methode waar een document wordt gebuild.

Een stuk code verplaatst naar een eigen methode waar het document wordt gemaakt.

Stuk code verplaatst waar de titleSlide wordt gesaved, dit een eigen methode gegeven zodat het ook nog ergens anders gebruikt zou kunnen worden

Stuk code verplaatst waar een textItem gesaved wordt, dit een eigen methode gegeven zodat het ook nog ergens anders gebruikt kan worden.

Stuk code verplaatst waar een BitmapItem gesaved wordt, dit een eigen methode gegeven zodat het ook nog ergens anders gebruikt kan worden.

Stuk code verplaatst waar een saveStartXMLFile gesaved wordt, dit een eigen methode gegeven zodat het ook nog ergens anders gebruikt kan worden.

Stuk code verplaatst waar een saveTitlePresentation gesaved wordt, dit een eigen methode gegeven zodat het ook nog ergens anders gebruikt kan worden.

De methodes hierboven gemaakt zodat iedere methode maar 1 verantwoordelijkheid heeft.

Niet gebruikte field verwijderd.

Alle static fields private gemaakt.