PatternsEditor

Release Report

JED-i

Dimitris Liakopoulos 2088
Ilias Mourtos 2302
Giannis Vasileiou 2647

VERSIONS HISTORY

Date	Version	Description	Author
19/5/2018	2.0	<2nd version of the software development pattern >	JED-i team

Introduction

This document provides information concerning the **<1st>** release of the project.

Purpose

A software development pattern defines a general reusable solution to a commonly occurring software development problem within a particular context. Patterns constitute a significant asset of the software engineering community. Amongst the very first approaches we have the GoF design patterns catalog that concerns best OO development practices. Then, there are also regular conferences (e.g. PLoP, EuroPLoP) that take place for more than 20 years and whose main topic is the identification of new patterns and pattern

languages (the term pattern language is typically used to refer to a set of related patterns). Patterns are formally specified in terms of pattern templates. So far, several pattern templates have been proposed in the literature.

The main goal of this project is to develop a PatternsEditor, an application that makes pattern writting easier, especially for young inexperienced pattern writers. At a glance, PatternsEditor shall allow a patterns writer to prepare a new pattern based on well known templates change the structure of an existing pattern by switching between these templates, and generate actual pattern documents in well known formats (simple text, Latex), and so on.

Document Structure

The rest of this document is structured as follows. Section 2 specifies the acceptance tests that have been employed for this release of the project. Section 3 specifies the main design concepts for this release of the project.

Acceptance Tests

<For the user stories included in this releases specify below corresponding tests using a typical tabular form.>

Tests for User Story **<1>**

Test ID	*<TestUserStory1>*
Class	*<Play ->PatternComposite.Class>*
Test Class	*<PatternComponent.class>*
Test Method	*<Play.createPatternLanguage()>*
Description	*<The test shows how a pattern language is created >*

Tests for User Story **<2>**

Test ID	*<TestUserStory2>*
Class	*<Play ->PatternComposite.Class>&<Play ->TemplateFactory.class>*
Test Class	*<PatternComponent.class>*
Test Method	*<Play.createPattern()>*
Description	*<The test shows how a pattern language add a pattern>*

Tests for User Story **<3>**

Test ID	*<TestUserStory3>*
Class	*<Play ->PatternComposite.Class>*

Test Class	*<PatternComponent.class>*
Test Method	*<Play.removePattern()>*
Description	*<The test shows how a pattern language remove a pattern>*

Tests for User Story **<4>**

Test ID	*<TestUserStory4>*
Class	*<CompositePattern.class>*
Test Class	*<PatternComponent.class>*
Test Method	*<Play.setContents()>*
Description	*<The test shows how a pattern update its contents>*

Tests for User Story **<5>**

Test ID	*<TestUserStory5>*
Class	*<Play ->CompositePattern.class>*
Test Class	*<PatternComponent.class>*
Test Method	*<Play.savePatternLanguage()>*
Description	*<The user save a pattern language in a file with txt format>*

Tests for User Story **<6>**

Test ID	*<TestUserStory6>*
Class	*<Play ->CompositePattern.class>*
Test Class	*<PatternComponent.class>*
Test Method	*<Play.loadPatternLanguage()>*
Description	*<The user load a pattern language from the disk>*

Tests for User Story **<7>**

Test ID	*<TestUserStory7>*
Class	*<Play ->CompositePattern.class>*
Test Class	*<PatternComponent.class>*
Test Method	*<Play.decoratePatternLanguage()>*
Description	*<The user decorate a pattern language as Latex>*

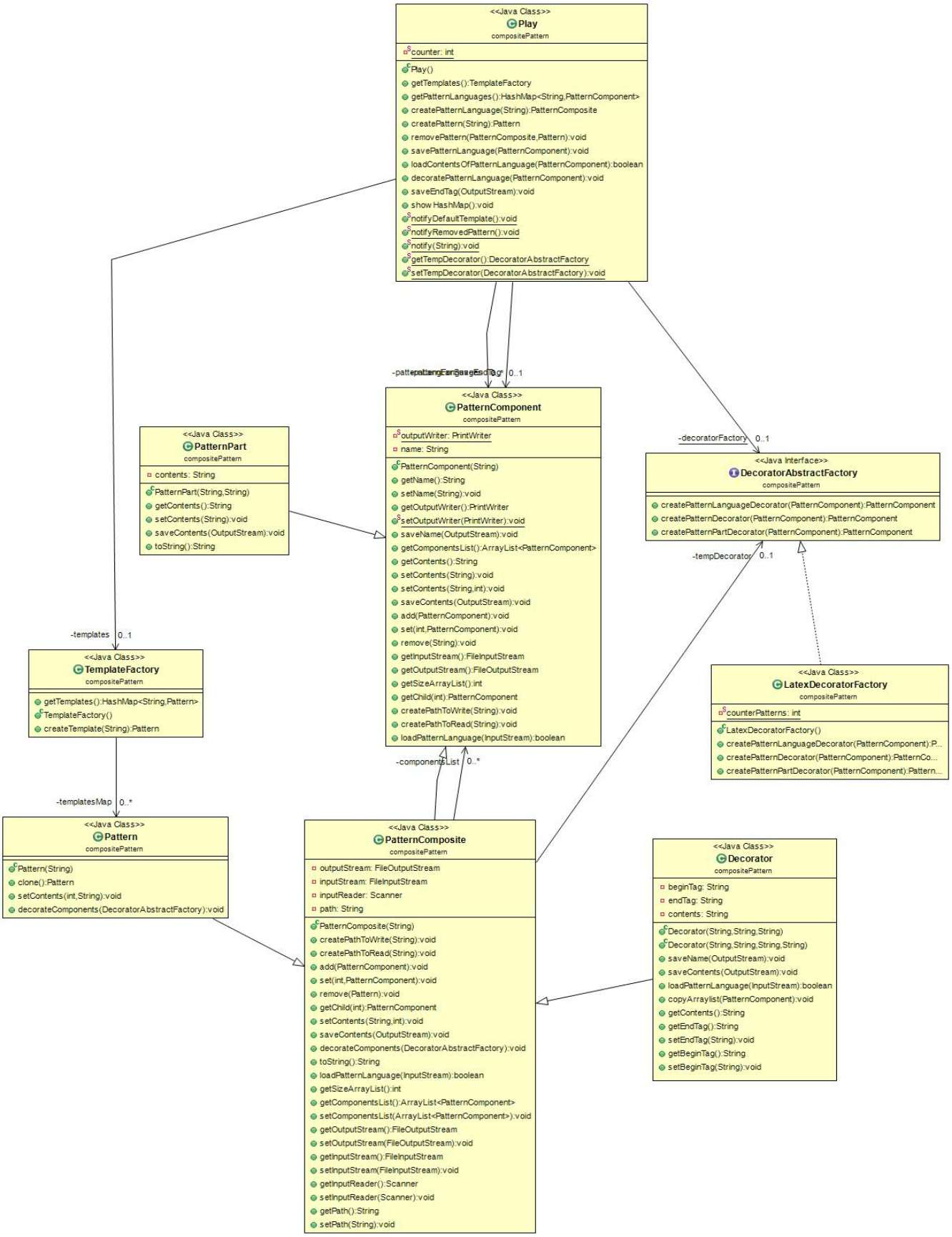
Tests for User Story **<8>**

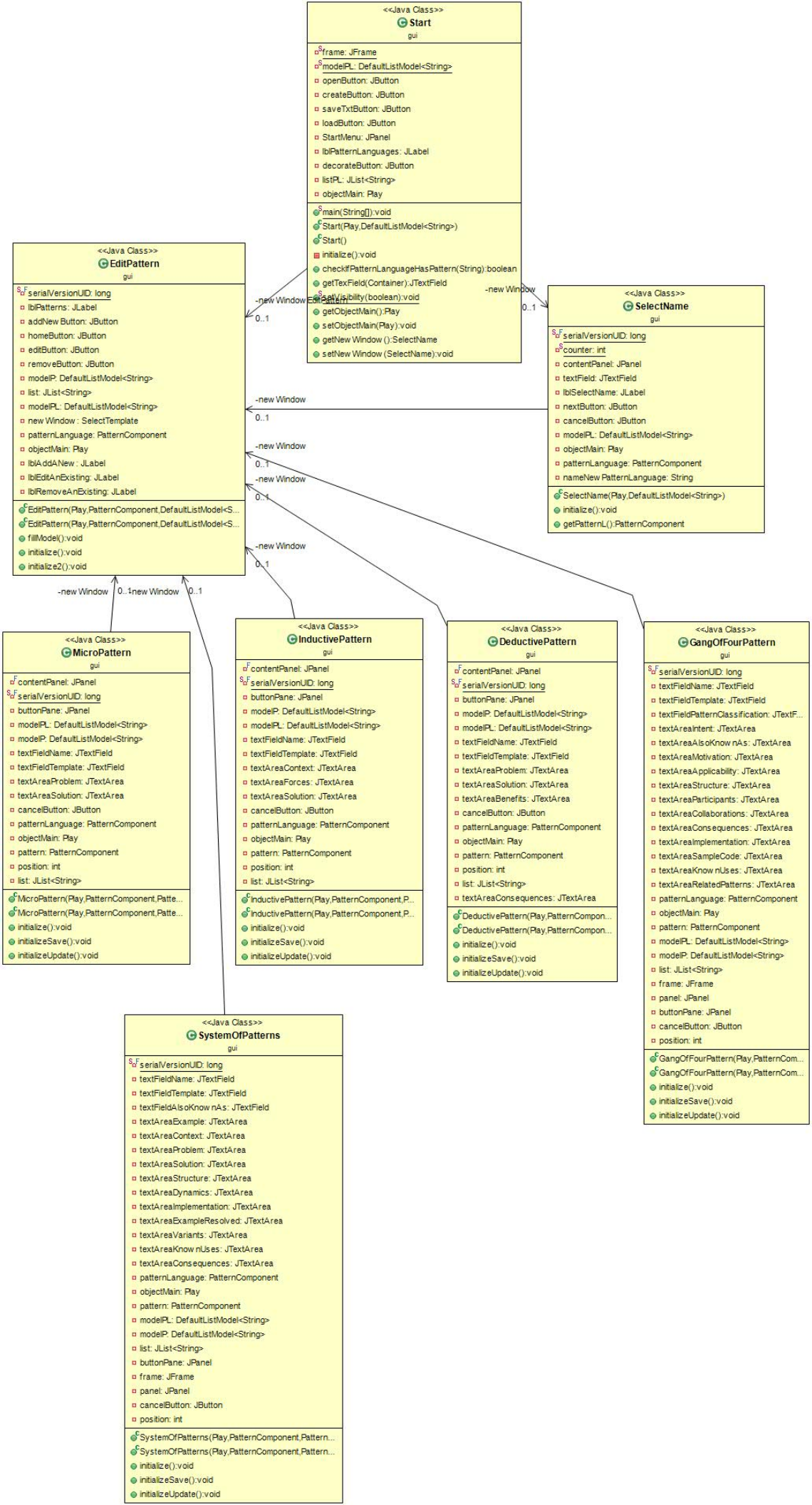
Test ID	*<TestUserStory8>*
Class	*<Play ->CompositePattern.class>*
Test Class	*<PatternComponent.class>*
Test Method	*<Play.saveatternLanguage()>*
Description	*<The user save a pattern language in a file with tex format>*

Tests for User Story **<9>**

Test ID	*<TestUserStory9>*
Class	*<Play ->CompositePattern.class>*
Test Class	*<PatternComponent.class>*
Test Method	*<Play.saveatternLanguage()>*
Description	*<The user load a pattern language from a disk>*

Design


<<Java Class>>
Play
compositePattern
counter: int
Play()
getTemplates():TemplateFactory
getPatternLanguages():HashMap<String,PatternComponent>
createPatternLanguage(String):PatternComposite
createPattern(String):Pattern
removePattern(PatternComposite,Pattern):void
savePatternLanguage(PatternComponent):void
loadContentsOfPatternLanguage(PatternComponent):boolean
decoratePatternLanguage(PatternComponent):void
saveEndTag(OutputStream):void
showHashMap():void
notifyDefaultTemplate():void
notifyRemovedPattern():void
notify(String):void
getTempDecorator():DecoratorAbstractFactory
setTempDecorator(DecoratorAbstractFactory):void
<<Java Class>>
PatternPart
compositePattern
contents: String
PatternPart(String,String)
getContents():String
setContents(String):void
saveContents(OutputStream):void
toString():String
<<Java Class>>
PatternComponent
compositePattern
outputWriter: PrintWriter
name: String
PatternComponent(String)
getName():String
setName(String):void
getOutputWriter():PrintWriter
setOutputWriter(PrintWriter):void
saveName(OutputStream):void
getComponentsList():ArrayList<PatternComponent>
getContents():String
setContents(String):void
setContents(String,int):void
saveContents(OutputStream):void
add(PatternComponent):void
set(int,PatternComponent):void
remove(String):void
getInputStream():FileInputStream
getOutputStream():FileOutputStream
getSizeArrayList():int
getChild(int):PatternComponent
createPathToWrite(String):void
createPathToRead(String):void
loadPatternLanguage(InputStream):boolean
-decoratorFactory 0..1
<<Java Interface>>
DecoratorAbstractFactory
compositePattern
createPatternLanguageDecorator(PatternComponent):PatternComponent
createPatternDecorator(PatternComponent):PatternComponent
createPatternPartDecorator(PatternComponent):PatternComponent
-tempDecorator 0..1
-templates 0..1
<<Java Class>>
TemplateFactory
compositePattern
getTemplates():HashMap<String,Pattern>
TemplateFactory()
createTemplate(String):Pattern
-templatesMap 0..*
<<Java Class>>
Pattern
compositePattern
Pattern(String)
clone():Pattern
setContents(int,String):void
decorateComponents(DecoratorAbstractFactory):void
-componentsList 0..*
<<Java Class>>
PatternComposite
compositePattern
outputStream: FileOutputStream
inputStream: FileInputStream
inputReader: Scanner
path: String
PatternComposite(String)
createPathToWrite(String):void
createPathToRead(String):void
add(PatternComponent):void
set(int,PatternComponent):void
remove(Pattern):void
getChild(int):PatternComponent
setContents(String,int):void
saveContents(OutputStream):void
decorateComponents(DecoratorAbstractFactory):void
toString():String
loadPatternLanguage(InputStream):boolean
getSizeArrayList():int
getComponentsList():ArrayList<PatternComponent>
setComponentsList(ArrayList<PatternComponent>):void
getOutputStream():FileOutputStream
setOutputStream(FileOutputStream):void
getInputStream():FileInputStream
setInputStream(FileInputStream):void
getInputReader():Scanner
setInputReader(Scanner):void
getPath():String
setPath(String):void
<<Java Class>>
LatexDecoratorFactory
compositePattern
counterPatterns: int
LatexDecoratorFactory()
createPatternLanguageDecorator(PatternComponent):P...
createPatternDecorator(PatternComponent):PatternCo...
createPatternPartDecorator(PatternComponent):Pattern...
<<Java Class>>
Decorator
compositePattern
beginTag: String
endTag: String
contents: String
Decorator(String,String,String)
Decorator(String,String,String,String)
saveName(OutputStream):void
saveContents(OutputStream):void
loadPatternLanguage(InputStream):boolean
copyArraylist(PatternComponent):void
getContents():String
getEndTag():String
setEndTag(String):void
getBeginTag():String
setBeginTag(String):void


<<Java Class>>
Start
gui
frame: JFrame
modelPL: DefaultListModel<String>
openButton: JButton
createButton: JButton
saveTxtButton: JButton
loadButton: JButton
StartMenu: JPanel
lblPatternLanguages: JLabel
decorateButton: JButton
listPL: JList<String>
objectMain: Play
main(String[]):void
Start(Play,DefaultListModel<String>)
Start()
initialize():void
checkIfPatternLanguageHasPattern(String):boolean
getTexField(Container):JTextField
setVisibility(boolean):void
getObjectMain():Play
setObjectMain(Play):void
getNew Window ():SelectName
setNew Window (SelectName):void
-new Window
0..1
<<Java Class>>
EditPattern
gui
serialVersionUID: long
lblPatterns: JLabel
addNew Button: JButton
homeButton: JButton
editButton: JButton
removeButton: JButton
modelP: DefaultListModel<String>
list: JList<String>
modelPL: DefaultListModel<String>
new Window : SelectTemplate
patternLanguage: PatternComponent
objectMain: Play
lblAddANew : JLabel
lblEditAnExisting: JLabel
lblRemoveAnExisting: JLabel
EditPattern(Play,PatternComponent,DefaultListModel<S...
EditPattern(Play,PatternComponent,DefaultListModel<S...
fillModel():void
initialize():void
initialize2():void
<<Java Class>>
SelectName
gui
serialVersionUID: long
counter: int
contentPanel: JPanel
textField: JTextField
lblSelectName: JLabel
nextButton: JButton
cancelButton: JButton
modelPL: DefaultListModel<String>
objectMain: Play
patternLanguage: PatternComponent
nameNew PatternLanguage: String
SelectName(Play,DefaultListModel<String>)
initialize():void
getPatternL():PatternComponent
<<Java Class>>
MicroPattern
gui
contentPanel: JPanel
serialVersionUID: long
buttonPane: JPanel
modelPL: DefaultListModel<String>
modelP: DefaultListModel<String>
textFieldName: JTextField
textFieldTemplate: JTextField
textAreaProblem: JTextArea
textAreaSolution: JTextArea
cancelButton: JButton
patternLanguage: PatternComponent
objectMain: Play
pattern: PatternComponent
position: int
list: JList<String>
MicroPattern(Play,PatternComponent,Patte...
MicroPattern(Play,PatternComponent,Patte...
initialize():void
initializeSave():void
initializeUpdate():void
<<Java Class>>
InductivePattern
gui
contentPanel: JPanel
serialVersionUID: long
buttonPane: JPanel
modelP: DefaultListModel<String>
modelPL: DefaultListModel<String>
textFieldName: JTextField
textFieldTemplate: JTextField
textAreaContext: JTextArea
textAreaForces: JTextArea
textAreaSolution: JTextArea
cancelButton: JButton
patternLanguage: PatternComponent
objectMain: Play
pattern: PatternComponent
position: int
list: JList<String>
InductivePattern(Play,PatternComponent,P...
InductivePattern(Play,PatternComponent,P...
initialize():void
initializeSave():void
initializeUpdate():void
<<Java Class>>
DeductivePattern
gui
contentPanel: JPanel
serialVersionUID: long
buttonPane: JPanel
modelP: DefaultListModel<String>
modelPL: DefaultListModel<String>
textFieldName: JTextField
textFieldTemplate: JTextField
textAreaProblem: JTextArea
textAreaSolution: JTextArea
textAreaBenefits: JTextArea
cancelButton: JButton
patternLanguage: PatternComponent
objectMain: Play
pattern: PatternComponent
position: int
list: JList<String>
textAreaConsequences: JTextArea
DeductivePattern(Play,PatternCompon...
DeductivePattern(Play,PatternCompon...
initialize():void
initializeSave():void
initializeUpdate():void
<<Java Class>>
GangOfFourPattern
gui
serialVersionUID: long
textFieldName: JTextField
textFieldTemplate: JTextField
textFieldPatternClassification: JTextF...
textAreaIntent: JTextArea
textAreaAlsoKnow nAs: JTextArea
textAreaMotivation: JTextArea
textAreaApplicability: JTextArea
textAreaStructure: JTextArea
textAreaParticipants: JTextArea
textAreaCollaborations: JTextArea
textAreaConsequences: JTextArea
textAreaImplementation: JTextArea
textAreaSampleCode: JTextArea
textAreaKnow nUses: JTextArea
textAreaRelatedPatterns: JTextArea
patternLanguage: PatternComponent
objectMain: Play
pattern: PatternComponent
modelPL: DefaultListModel<String>
modelP: DefaultListModel<String>
list: JList<String>
frame: JFrame
panel: JPanel
buttonPane: JPanel
cancelButton: JButton
position: int
GangOfFourPattern(Play,PatternCom...
GangOfFourPattern(Play,PatternCom...
initialize():void
initializeSave():void
initializeUpdate():void
<<Java Class>>
SystemOfPatterns
gui
serialVersionUID: long
textFieldName: JTextField
textFieldTemplate: JTextField
textFieldAlsoKnow nAs: JTextField
textAreaExample: JTextArea
textAreaContext: JTextArea
textAreaProblem: JTextArea
textAreaSolution: JTextArea
textAreaStructure: JTextArea
textAreaDynamics: JTextArea
textAreaImplementation: JTextArea
textAreaExampleResolved: JTextArea
textAreaVariants: JTextArea
textAreaKnow nUses: JTextArea
textAreaConsequences: JTextArea
patternLanguage: PatternComponent
objectMain: Play
pattern: PatternComponent
modelPL: DefaultListModel<String>
modelP: DefaultListModel<String>
list: JList<String>
buttonPane: JPanel
frame: JFrame
panel: JPanel
cancelButton: JButton
position: int
SystemOfPatterns(Play,PatternComponent,Pattern...
SystemOfPatterns(Play,PatternComponent,Pattern...
initialize():void
initializeSave():void
initializeUpdate():void
-new Window
0..1
-new Window
0..1
-new Window
0..1
-new Window
0..1
-new Window
0..1
-new Window
0..1
-new Window
0..1

Class Name: PatternComponent	
Responsibilities: • Αναπαρασταση μεθόδων τις οποίες εχουν την δυνατότητα κλάσεις που κληρονομούν απο αυτές να τις υλοποιήσουν με το δικό τους τρόπο. • Κρατάει το όνομα των PatternLanguage, Pattern, Pattern-part(templates).	**Collaborations:** • PatternPart • PatternComposite • TemplateFactory • Pattern

Class Name: PatternComposite	
Responsibilities: • Αναπαράσταση των περιεχομενων τωνpatternlanguage και των pattern • Δημιουργία patternlangauge • Επεξεργασία αυτών • Αφαίρεση • Αποθηκευση τους • Διαμορφωση ως LateX αρχειο	**Collaborations:** • PatternComponent • Pattern • Pattern Part • Template Factory • Decorator • LatexDecoratorFactory • DecoratorAbstractFactory

Class Name: PatternPart	
Responsibilities: • Αναπαράσταση των περιεχομενων των pattern • Δημιουργία pattern • Επεξεργασία αυτών • Αφαίρεση • Αποθηκευση τους	**Collaborations:** • PatternComponent • Pattern • Template Factory

Class Name: TemplateFactory	
Responsibilities: • Αναπαρασταση των περιεχομενων της φορμας (templates) • Δημιουργια των περιεχομενων των templates • Επεξεργασία αυτών • Αποθήκευση αυτών	**Collaborations:** • PatternComponent • Pattern

Class Name: Pattern	
Responsibilities: • Δημιουργια Pattern • Διαμορφωση ως LateX αρχειο	**Collaborations:** • PatternComponent • PatternComposite • Pattern • LatexDecoratorFactory • DecoratorAbstractFactory

Class Name: DecoratorAbstractFactory	
Responsibilities: • Δηλωση μεθοδων	**Collaborations:** • PatternComposite • LatexDecoratorFactory • Decorator • Pattern

Class Name: LatexDecoratorFactory	
Responsibilities: • Δημιουργια μεθοδων • Διαμορφωση των Pattern Language σε μορφη Latex • Διαμορφωση των Pattern σε μορφη Latex • Διαμορφωση των Pattern Part σε μορφη Latex	**Collaborations:** • PatternComposite • LatexDecoratorFactory • Decorator • Pattern

Class Name: Decorator	
Responsibilities: • Δημιουργια μεθοδων • Αποθηκευση των Decorator αντικειμενων σε μορφη Latex	**Collaborations:** • PatternComposite • LatexDecoratorFactory • DecoratorAbstractFactory • Pattern

GUI:

Class Name: Start	
Responsibilities: • Δημιουργεί το πρώτο frame με το μενού για προσθήκη/επεξεργασία των Pattern Language • Δημιουργεί το παράθυρο επεξεργασίας των Pattern κάθε γλώσσας • Δημιουργεί το παράθυρο που δίνεται το όνομα της γλώσσας που θελουμε να φτιάξουμε	**Collaborations:** • SelectName • EditPattern

Class Name: SelectName	
Responsibilities: • Δίνει όνομα στην γλώσσα την οποία πρόκειται να δημιουργήσουμε	**Collaborations:** • EditPattern

Class Name: EditPattern	
Responsibilities: • Προσθήκη/Επεξεργασία/ Διαγραφή των Pattern μιας γλώσσας	**Collaborations:** • Με όλα τα frame των προεπιλεγμένων template • Start • SelectTemplate

Class Name: Templates	
Responsibilities: • Είδοδος δεδομένων για το pattern που επεξεργαζόμαστε	**Collaborations:** • EditPattern

Implementation

A printed copy of the source code (including the implementation of (1) classes and (2) tests) of this release is attached to the release report.