Cryptographics

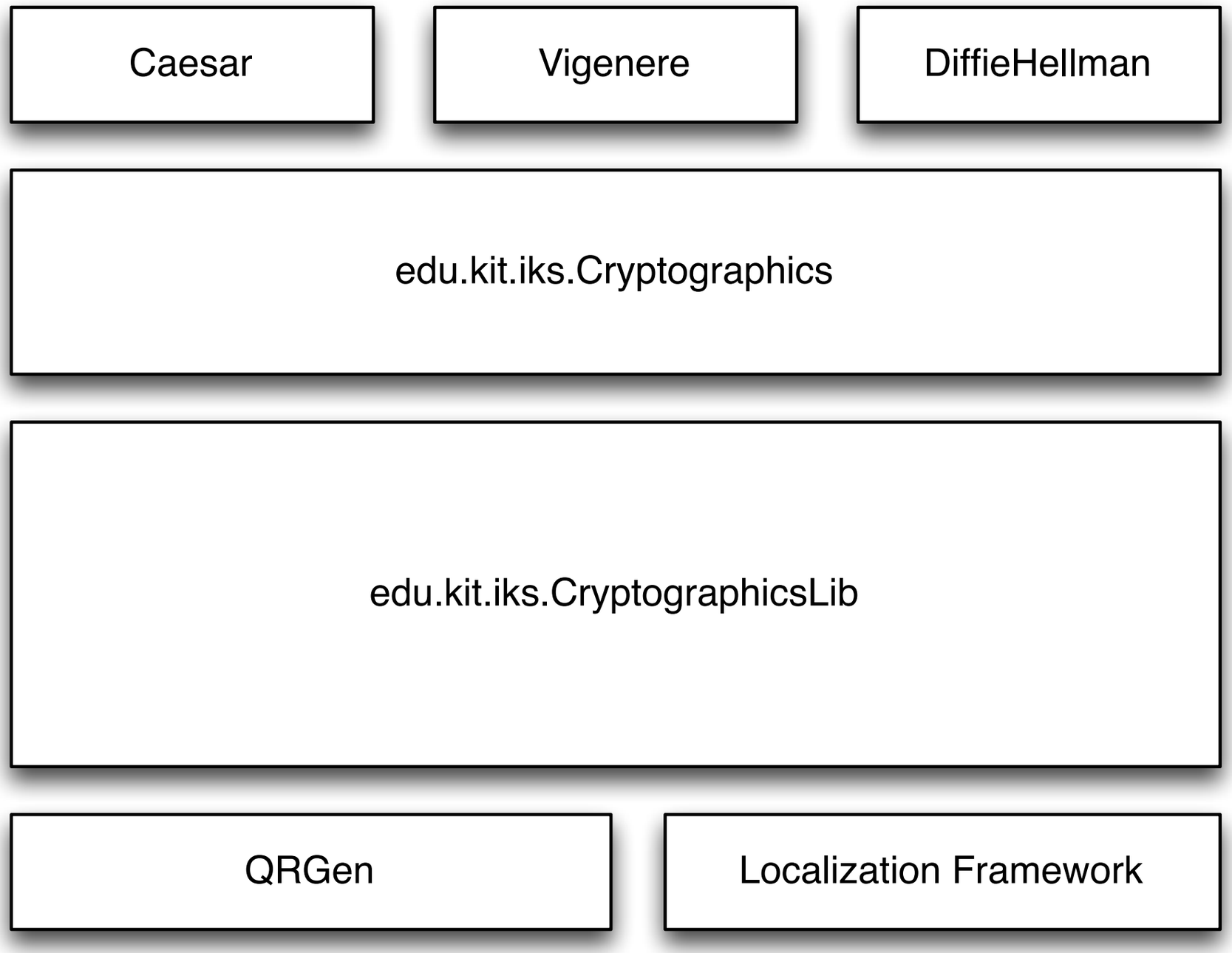
Entwurf

Inhalt

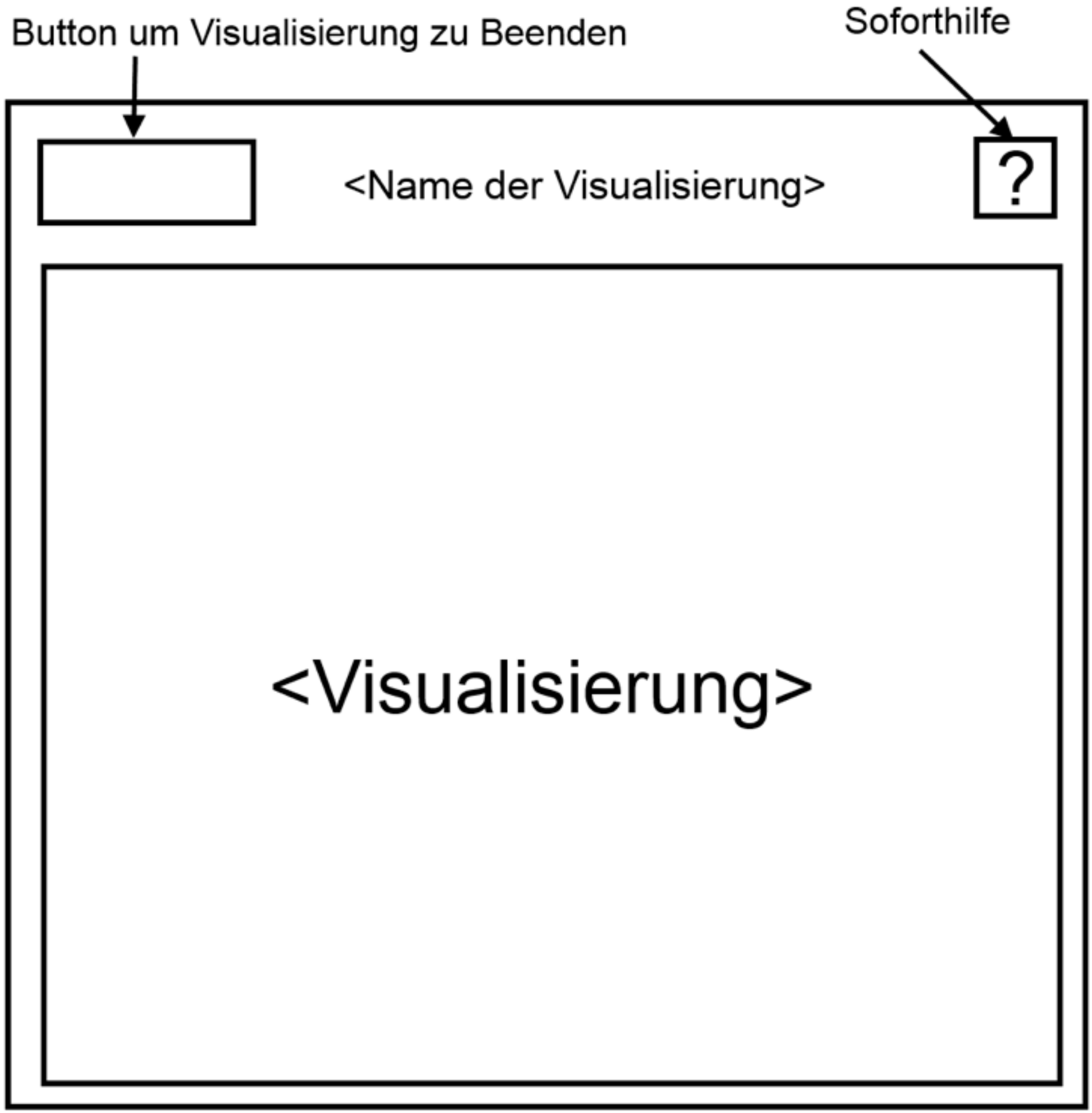
- Architektur
- Wichtige Klassen
- Klassendiagramme
- Sequenzdiagramme
- Arbeitsteilung

Architektur

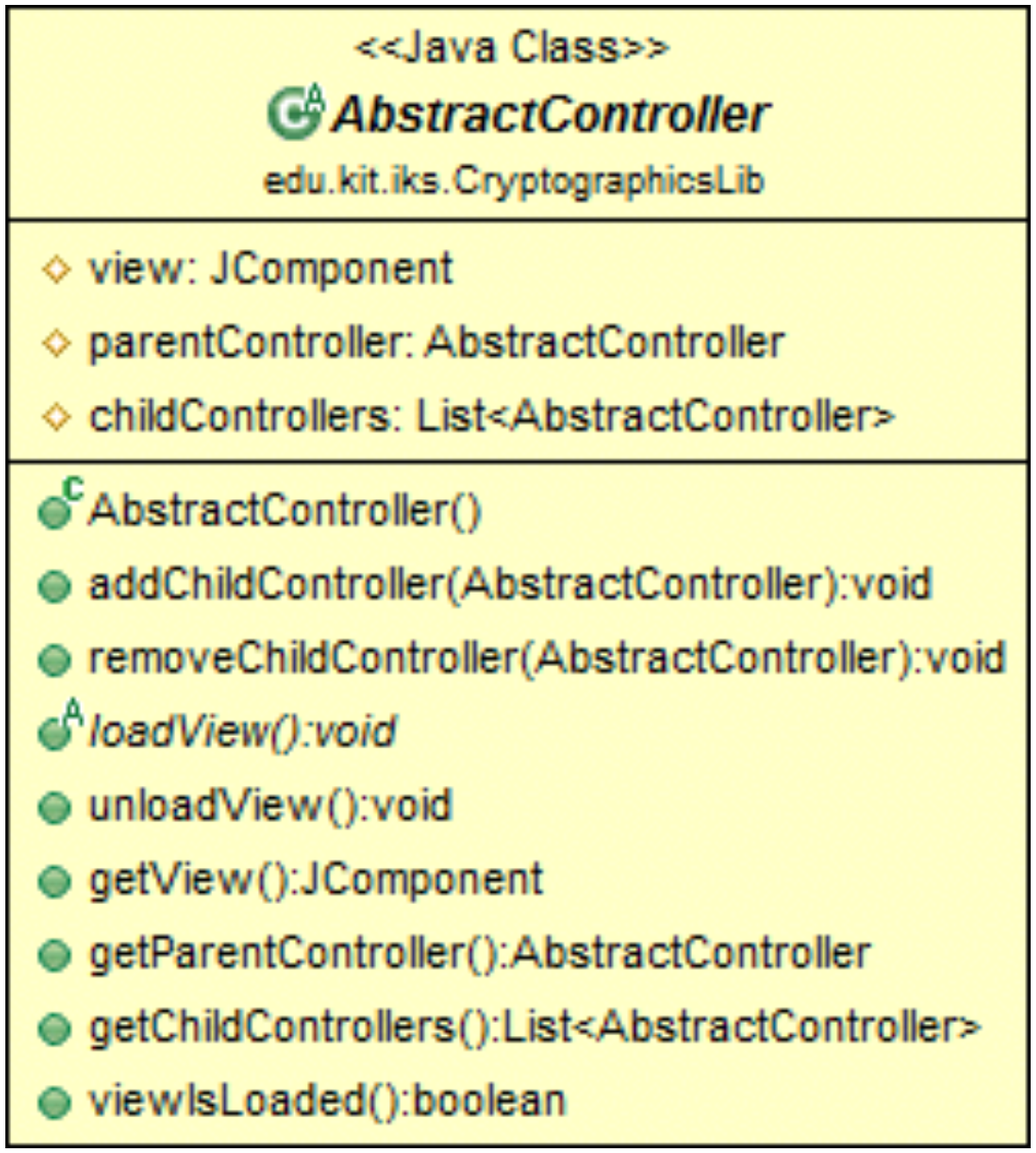
Schichtmodell



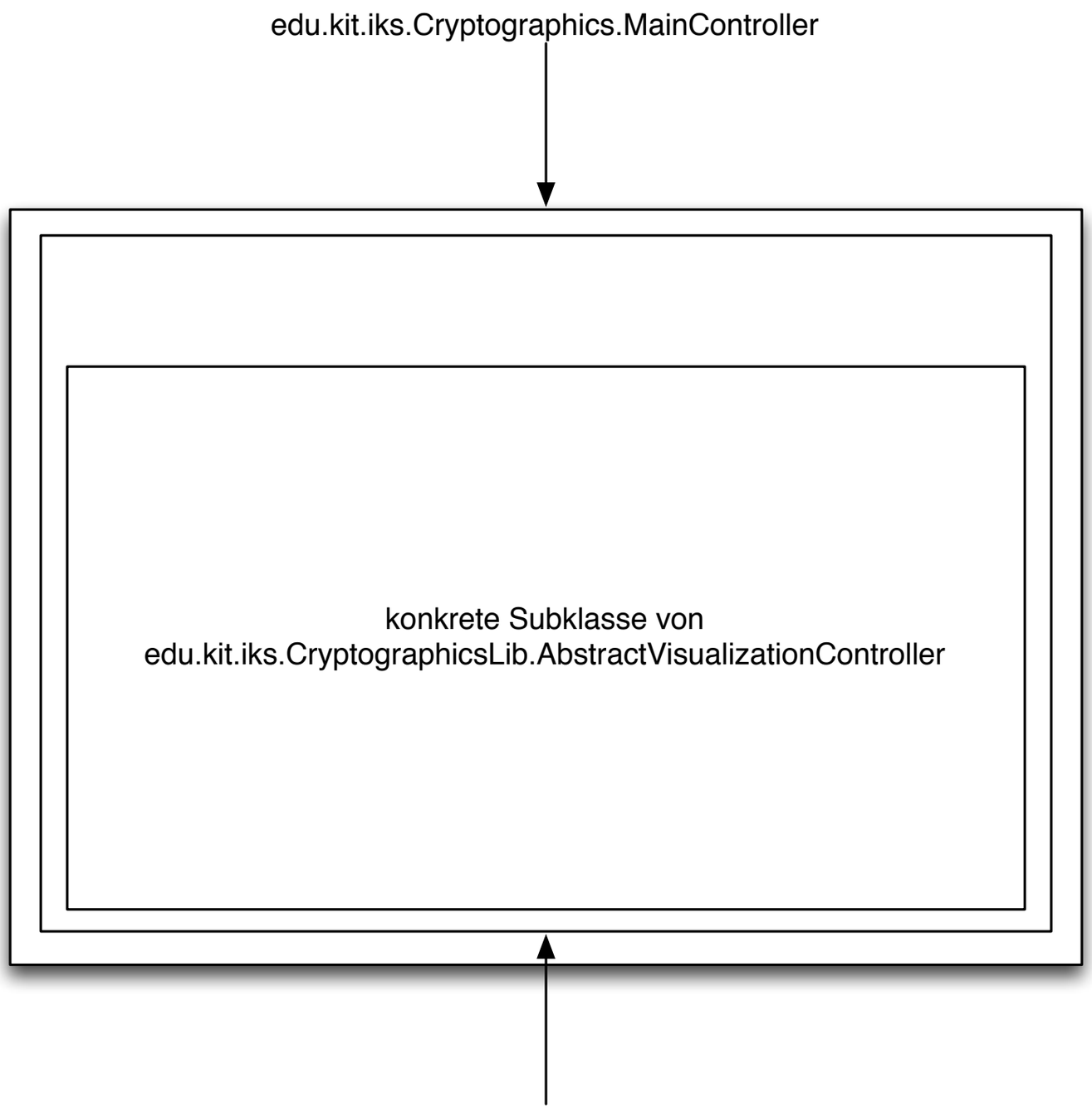
Wichtige Klassen


Button um Visualisierung zu Beenden
Soforthilfe
?
<Name der Visualisierung>
<Visualisierung>

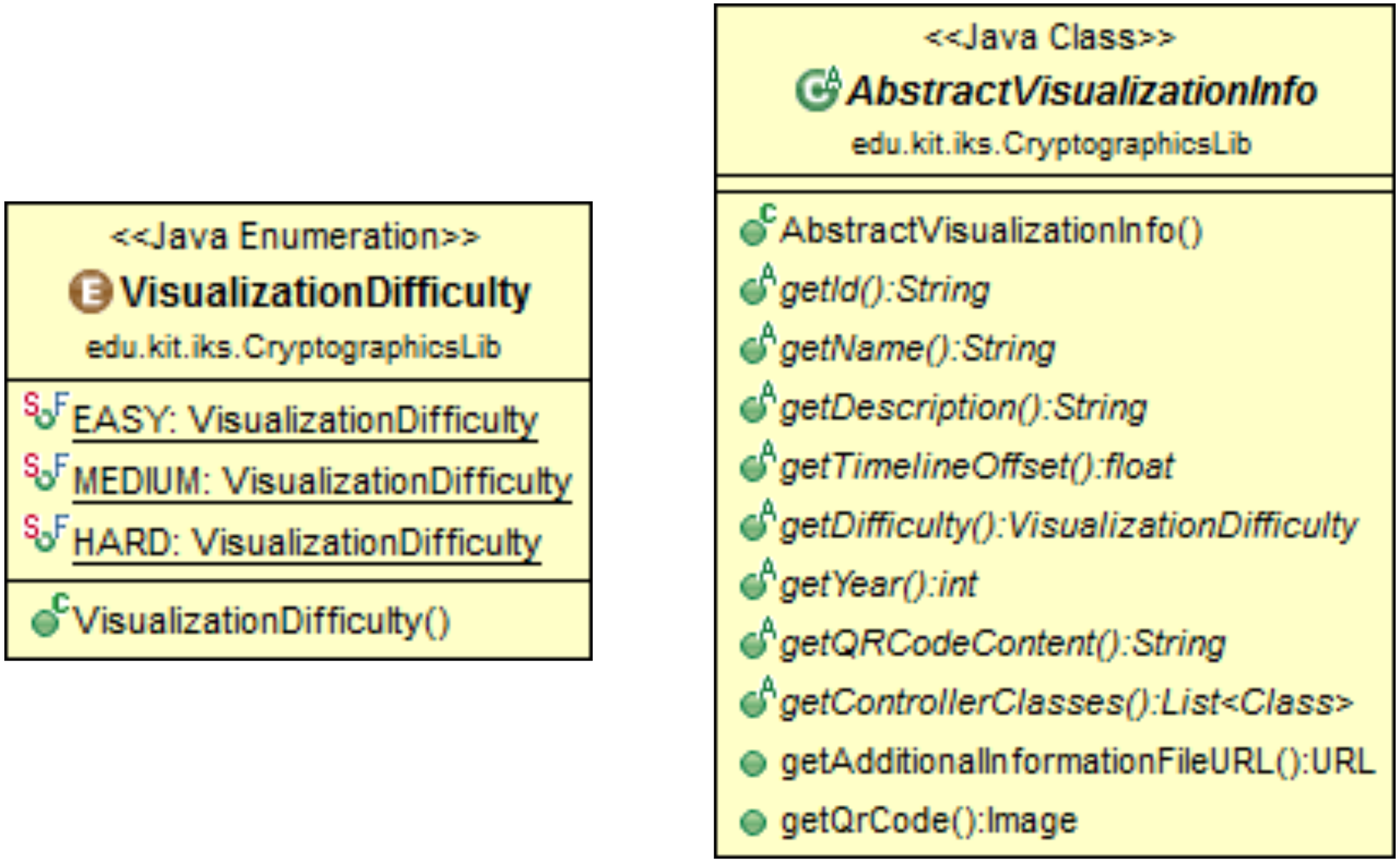
AbstractController



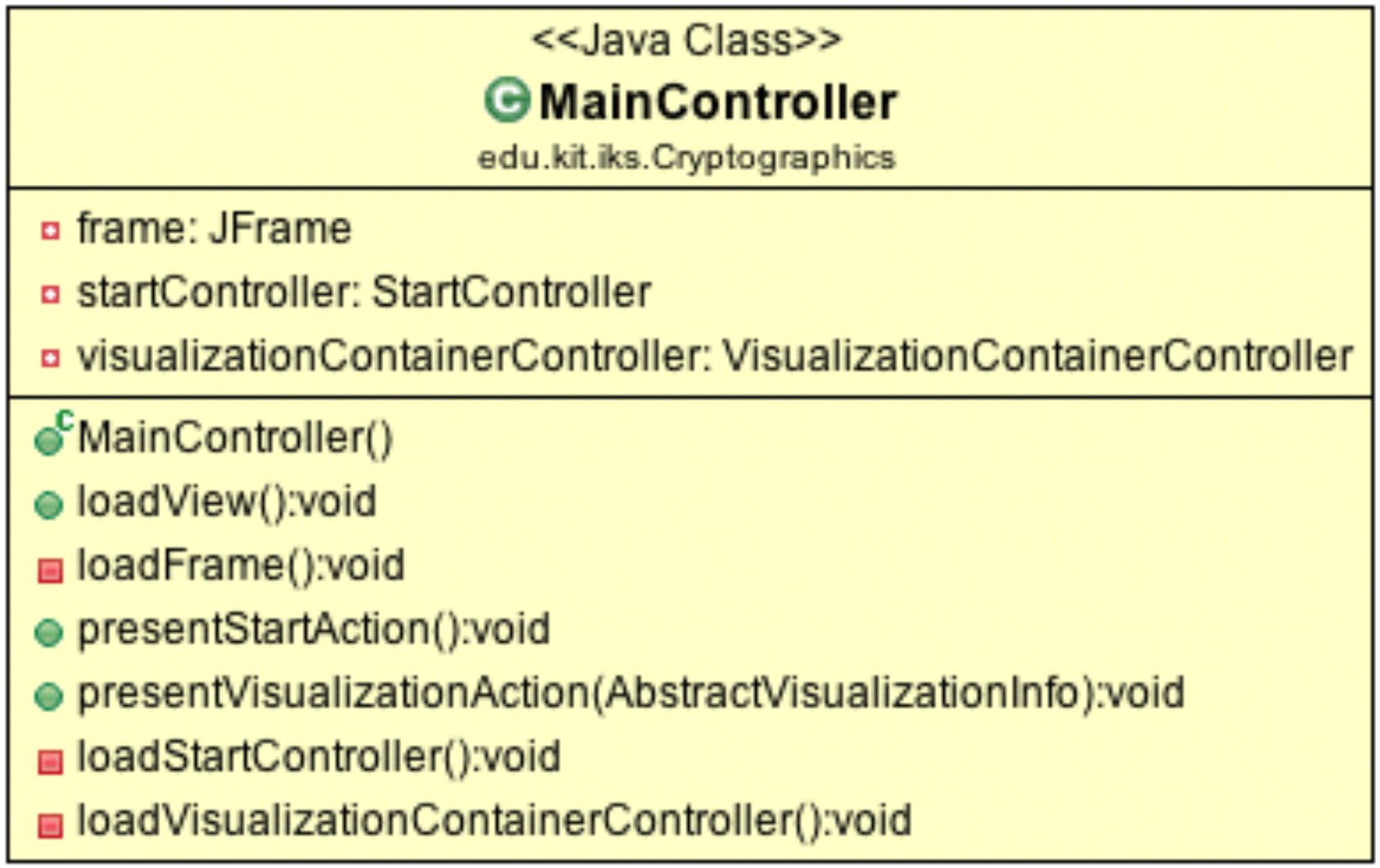
Controller-Hierarchie



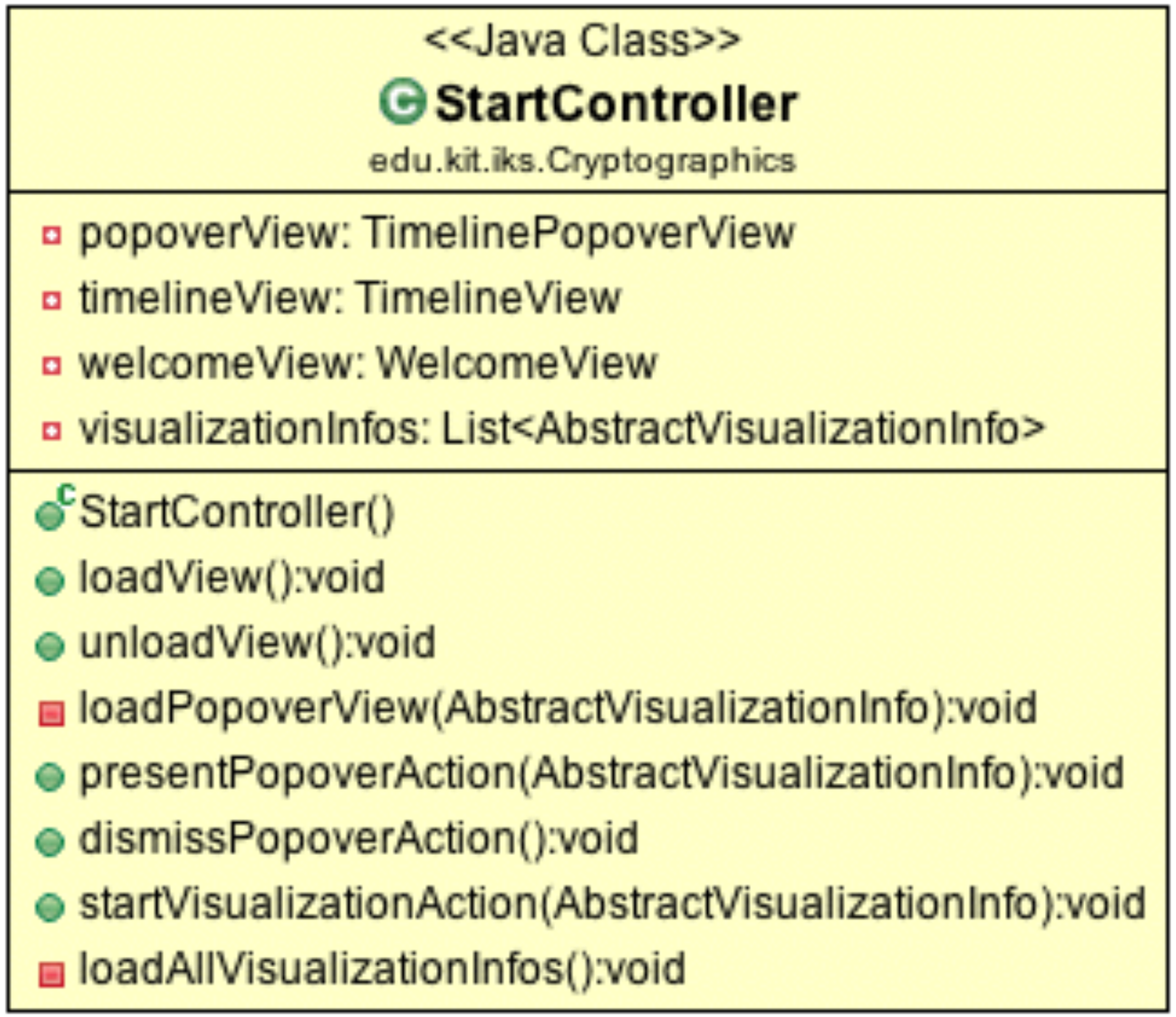
AbstractVisualizationInfo



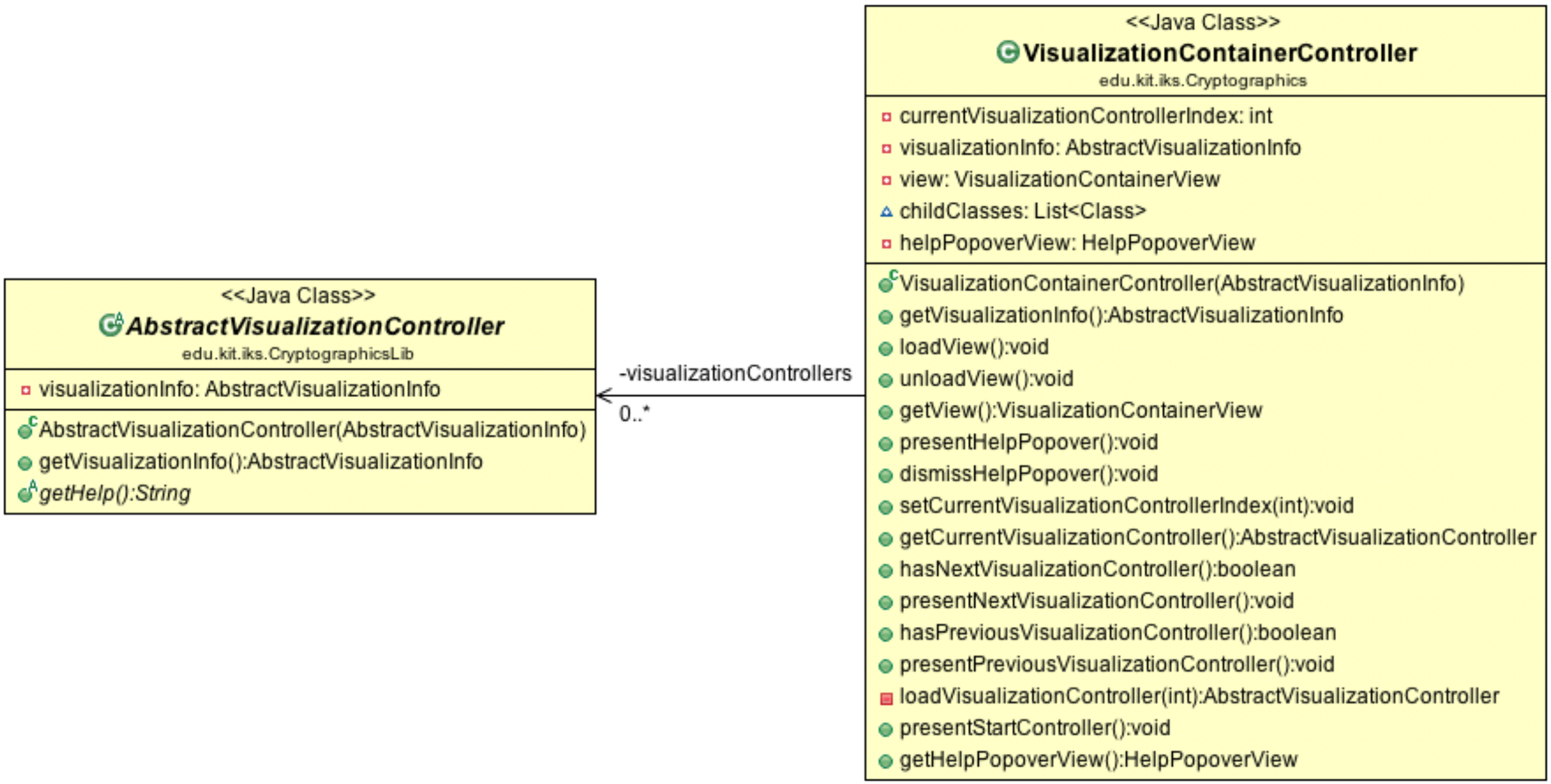
MainController



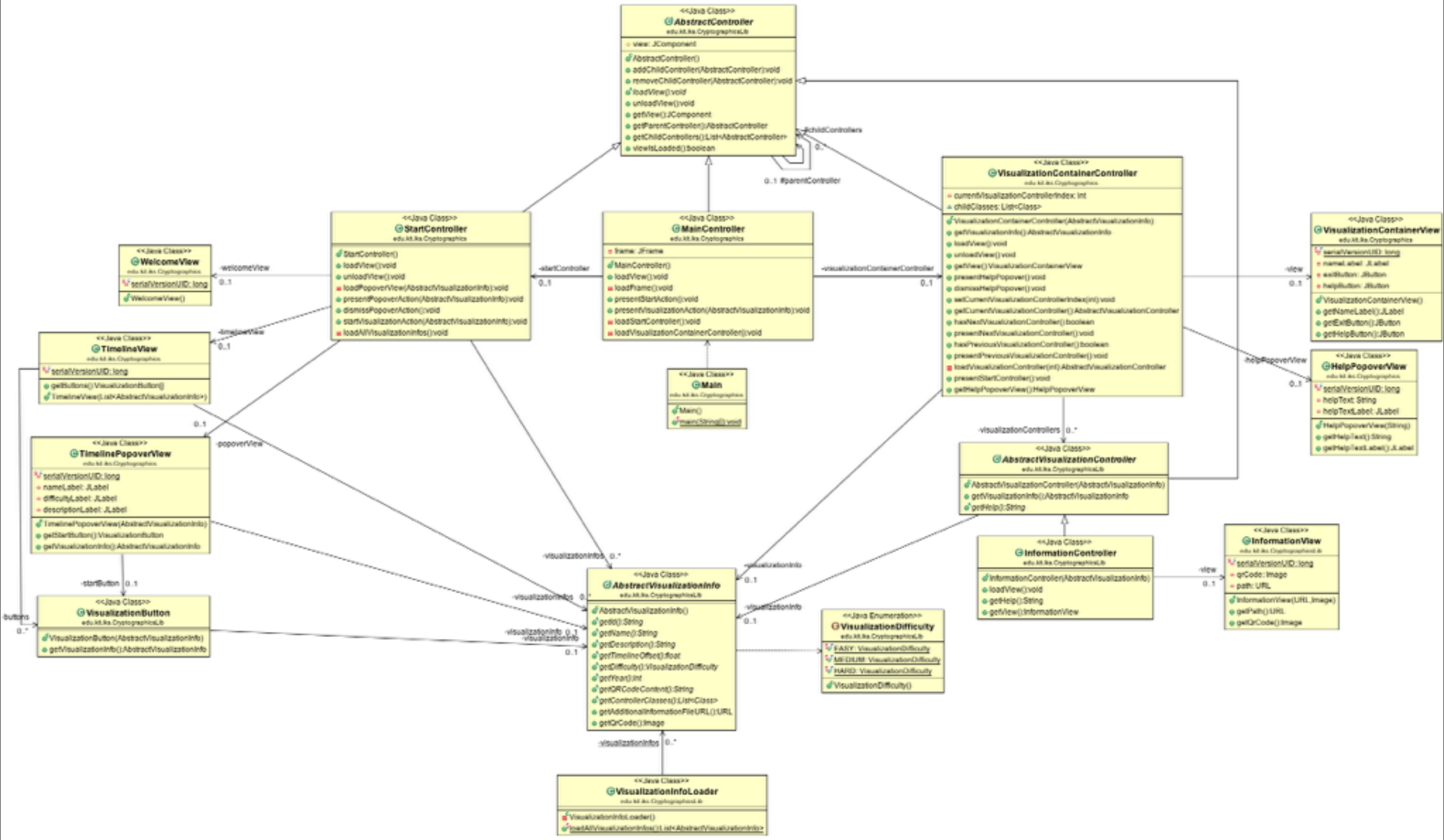
StartController

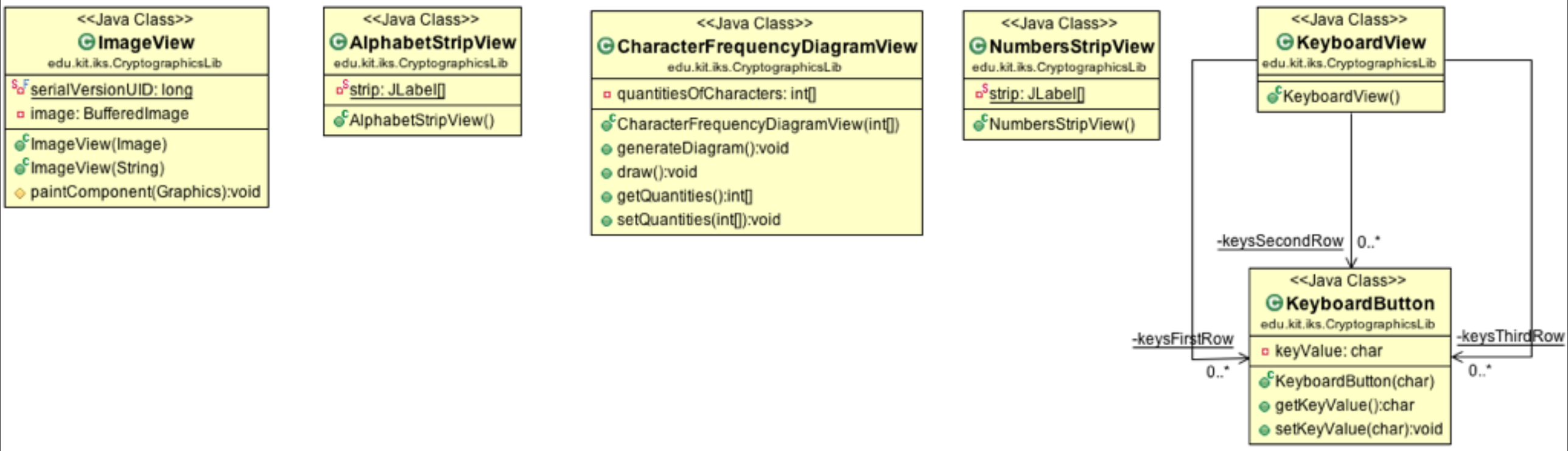


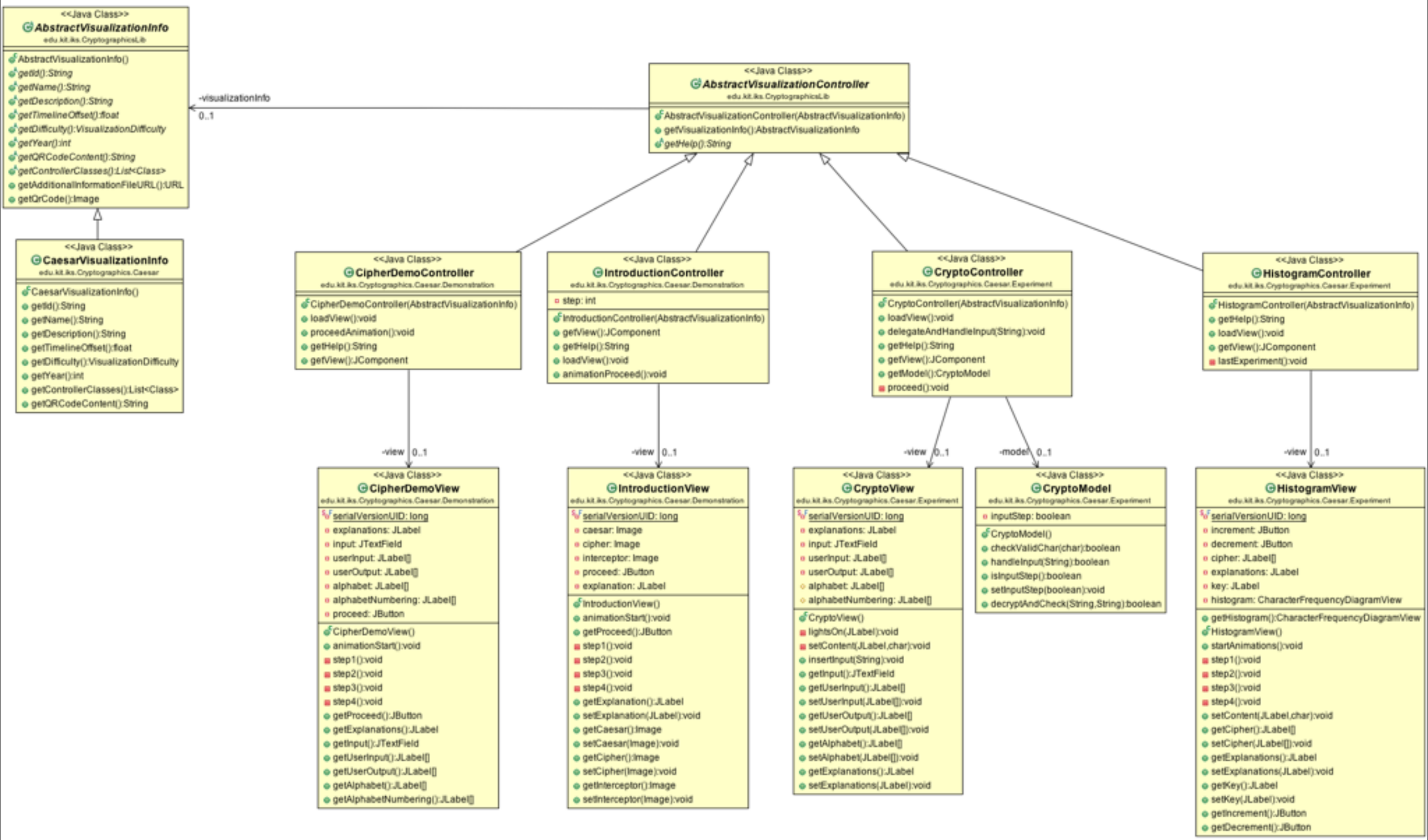
Visualization Controllers

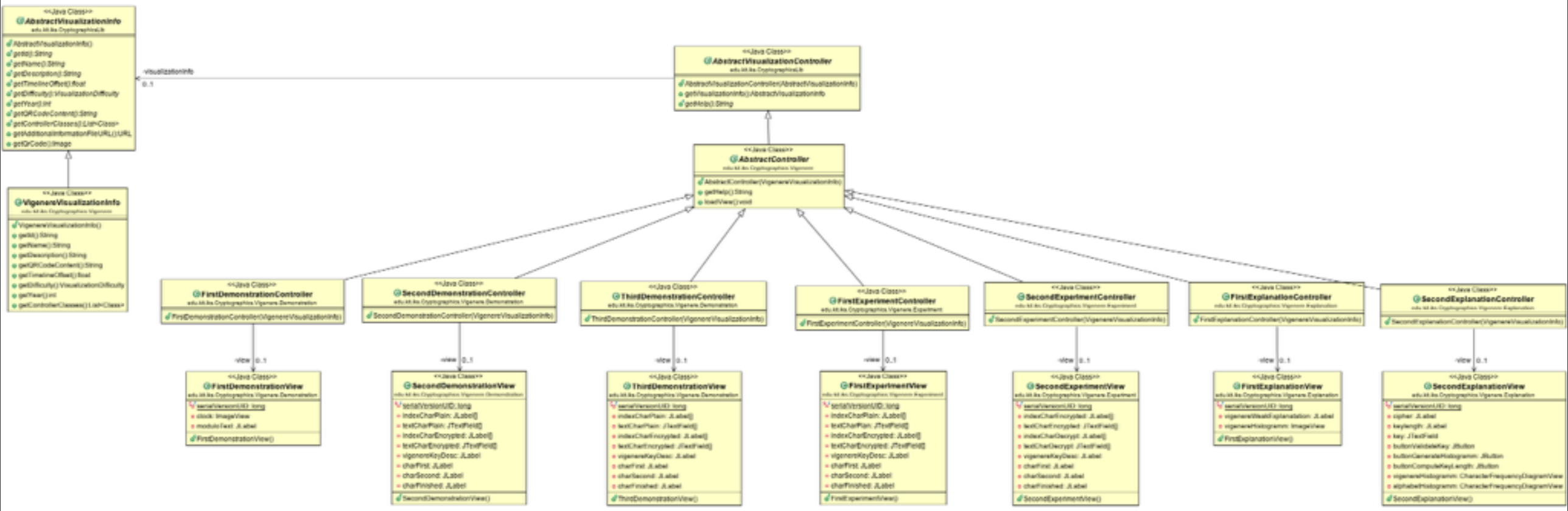


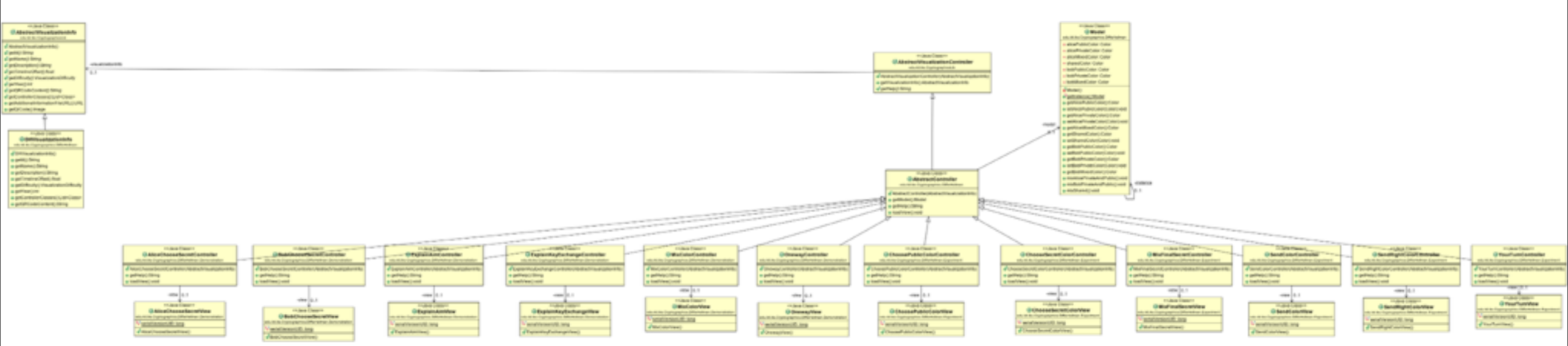
Klassendiagramme


<<Java Class>>
AbstractController
edu.kit.ika.CryptographicsLib
- view: JComponent
+ AbstractController()
+ addChildController(AbstractController):void
+ removeChildController(AbstractController):void
+ loadView():void
+ unloadView():void
+ getView():JComponent
+ getParentController():AbstractController
+ getChildControllers():List<AbstractController>
+ viewIsLoaded():boolean
-childControllers
0..*
0..1 #parentController
<<Java Class>>
StartController
edu.kit.ika.Cryptographics
+ StartController()
+ loadView():void
+ unloadView():void
- loadPopoverView(AbstractVisualizationInfo):void
+ presentPopoverAction(AbstractVisualizationInfo):void
+ dismissPopoverAction():void
+ startVisualizationAction(AbstractVisualizationInfo):void
- loadAllVisualizationInfos():void
<<Java Class>>
MainController
edu.kit.ika.Cryptographics
- frame: JFrame
+ MainController()
+ loadView():void
- loadFrame():void
+ presentStartAction():void
+ presentVisualizationAction(AbstractVisualizationInfo):void
- loadStartController():void
- loadVisualizationContainerController():void
<<Java Class>>
Main
edu.kit.ika.Cryptographics
+ Main()
+ main(String[]):void
<<Java Class>>
VisualizationContainerController
edu.kit.ika.Cryptographics
- currentVisualizationControllerIndex: int
childClasses: List<Class>
+ VisualizationContainerController(AbstractVisualizationInfo)
+ getVisualizationInfo():AbstractVisualizationInfo
+ loadView():void
+ unloadView():void
+ getView():VisualizationContainerView
+ presentHelpPopover():void
+ dismissHelpPopover():void
+ setCurrentVisualizationControllerIndex(int):void
+ getCurrentVisualizationController():AbstractVisualizationController
+ hasNextVisualizationController():boolean
+ presentNextVisualizationController():void
+ hasPreviousVisualizationController():boolean
+ presentPreviousVisualizationController():void
- loadVisualizationController(int):AbstractVisualizationController
+ presentStartController():void
+ getHelpPopoverView():HelpPopoverView
<<Java Class>>
VisualizationContainerView
edu.kit.ika.Cryptographics
- serialVersionUID: long
- nameLabel: JLabel
- exitButton: JButton
- helpButton: JButton
+ VisualizationContainerView()
+ getNameLabel():JLabel
+ getExitButton():JButton
+ getHelpButton():JButton
<<Java Class>>
HelpPopoverView
edu.kit.ika.Cryptographics
- serialVersionUID:long
- helpText: String
- helpTextLabel: JLabel
+ HelpPopoverView(String)
+ getHelpText():String
+ getHelpTextLabel():JLabel
<<Java Class>>
WelcomeView
edu.kit.ika.Cryptographics
- serialVersionUID: long
+ WelcomeView()
-welcomeView
0..1
-startController
0..1
-visualizationContainerController
0..1
-view
0..1
-helpPopoverView
0..1
<<Java Class>>
TimelineView
edu.kit.ika.Cryptographics
- serialVersionUID: long
+ getButtons():VisualizationButton[]
+ TimelineView(List<AbstractVisualizationInfo>)
-timelineView
0..1
0..1
-popoverView
<<Java Class>>
TimelinePopoverView
edu.kit.ika.Cryptographics
- serialVersionUID:long
- nameLabel: JLabel
- difficultyLabel: JLabel
- descriptionLabel: JLabel
+ TimelinePopoverView(AbstractVisualizationInfo)
+ getStartButton():VisualizationButton
+ getVisualizationInfo():AbstractVisualizationInfo
-startButton
0..1
<<Java Class>>
VisualizationButton
edu.kit.ika.CryptographicsLib
+ VisualizationButton(AbstractVisualizationInfo)
+ getVisualizationInfo():AbstractVisualizationInfo
-buttons
0..*
-visualizationInfo
0..1
-visualizationInfo
0..1
-visualizationInfos
0..*
<<Java Class>>
AbstractVisualizationInfo
edu.kit.ika.CryptographicsLib
+ AbstractVisualizationInfo()
+ getId():String
+ getName():String
+ getDescription():String
+ getTimelineOffset():float
+ getDifficulty():VisualizationDifficulty
+ getYear():int
+ getQRCodeContent():String
+ getControllerClasses():List<Class>
+ getAdditionalInformationFileURL():URL
+ getQrCode():Image
-visualizationInfo
0..1
-visualizationInfo
0..1
-visualizationInfos
0..*
<<Java Enumeration>>
VisualizationDifficulty
edu.kit.ika.CryptographicsLib
EASY: VisualizationDifficulty
MEDIUM: VisualizationDifficulty
HARD: VisualizationDifficulty
+ VisualizationDifficulty()
<<Java Class>>
VisualizationInfoLoader
edu.kit.ika.Cryptographics.ik
- VisualizationInfoLoader()
+ loadAllVisualizationInfos():List<AbstractVisualizationInfo>
-visualizationControllers
0..*
<<Java Class>>
AbstractVisualizationController
edu.kit.ika.CryptographicsLib
+ AbstractVisualizationController(AbstractVisualizationInfo)
+ getVisualizationInfo():AbstractVisualizationInfo
+ getHelp():String
<<Java Class>>
InformationController
edu.kit.ika.CryptographicsLib
+ InformationController(AbstractVisualizationInfo)
+ loadView():void
+ getHelp():String
+ getView():InformationView
-view
0..1
<<Java Class>>
InformationView
edu.kit.ika.Cryptographics.ik
- serialVersionUID: long
- qrCode: Image
- path: URL
+ InformationView(URL,Image)
+ getPath():URL
+ getQrCode():Image

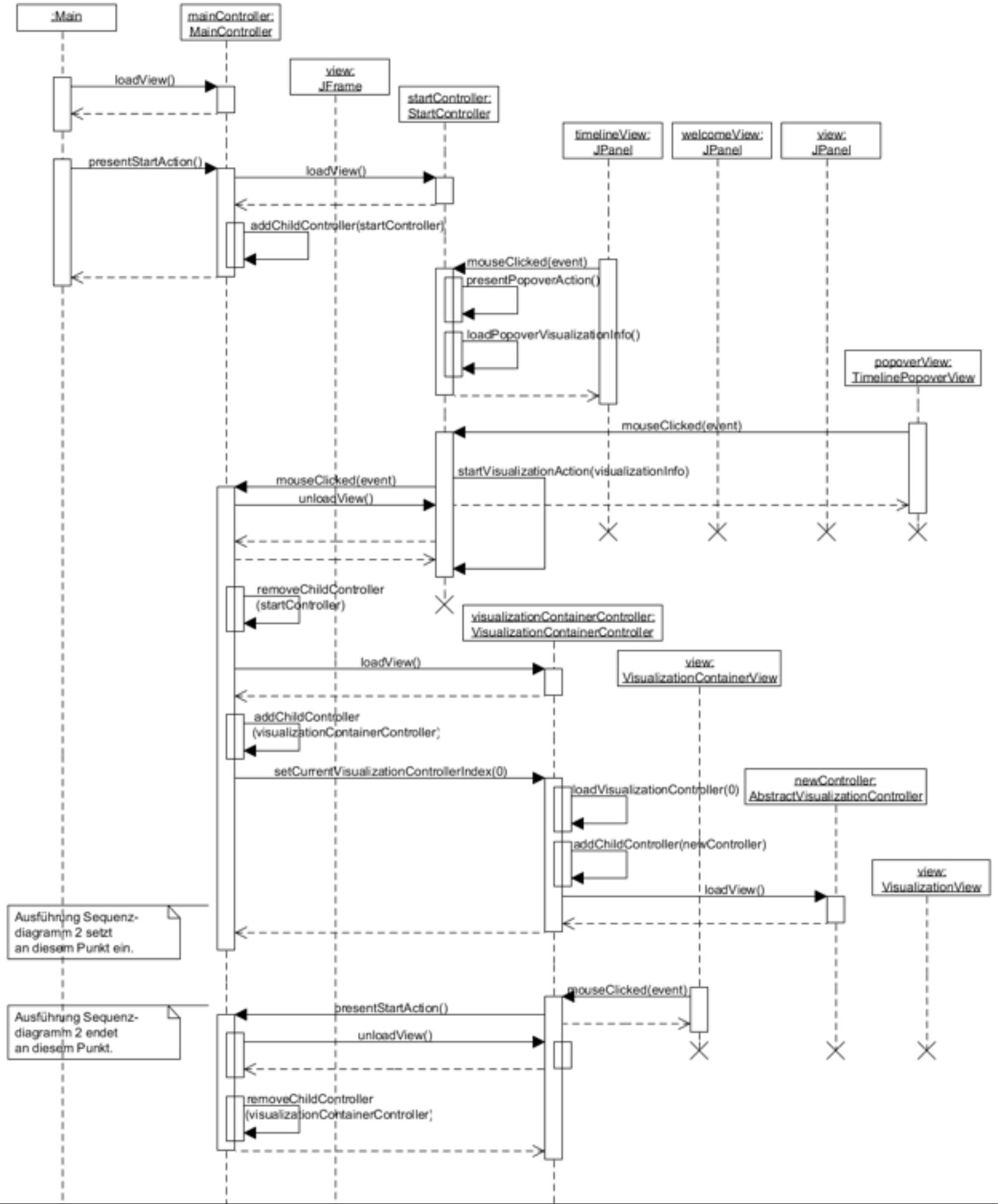

<<Java Class>>
ImageView
edu.kit.iks.CryptographicsLib
serialVersionUID: long
image: BufferedImage
ImageView(Image)
ImageView(String)
paintComponent(Graphics):void
<<Java Class>>
AlphabetStripView
edu.kit.iks.CryptographicsLib
strip: JLabel[]
AlphabetStripView()
<<Java Class>>
CharacterFrequencyDiagramView
edu.kit.iks.CryptographicsLib
quantitiesOfCharacters: int[]
CharacterFrequencyDiagramView(int[])
generateDiagram():void
draw():void
getQuantities():int[]
setQuantities(int[]):void
<<Java Class>>
NumbersStripView
edu.kit.iks.CryptographicsLib
strip: JLabel[]
NumbersStripView()
<<Java Class>>
KeyboardView
edu.kit.iks.CryptographicsLib
KeyboardView()
-keysSecondRow
0..*
<<Java Class>>
KeyboardButton
edu.kit.iks.CryptographicsLib
keyValue: char
KeyboardButton(char)
getKeyValue():char
setKeyValue(char):void
-keysFirstRow
0..*
-keysThirdRow
0..*

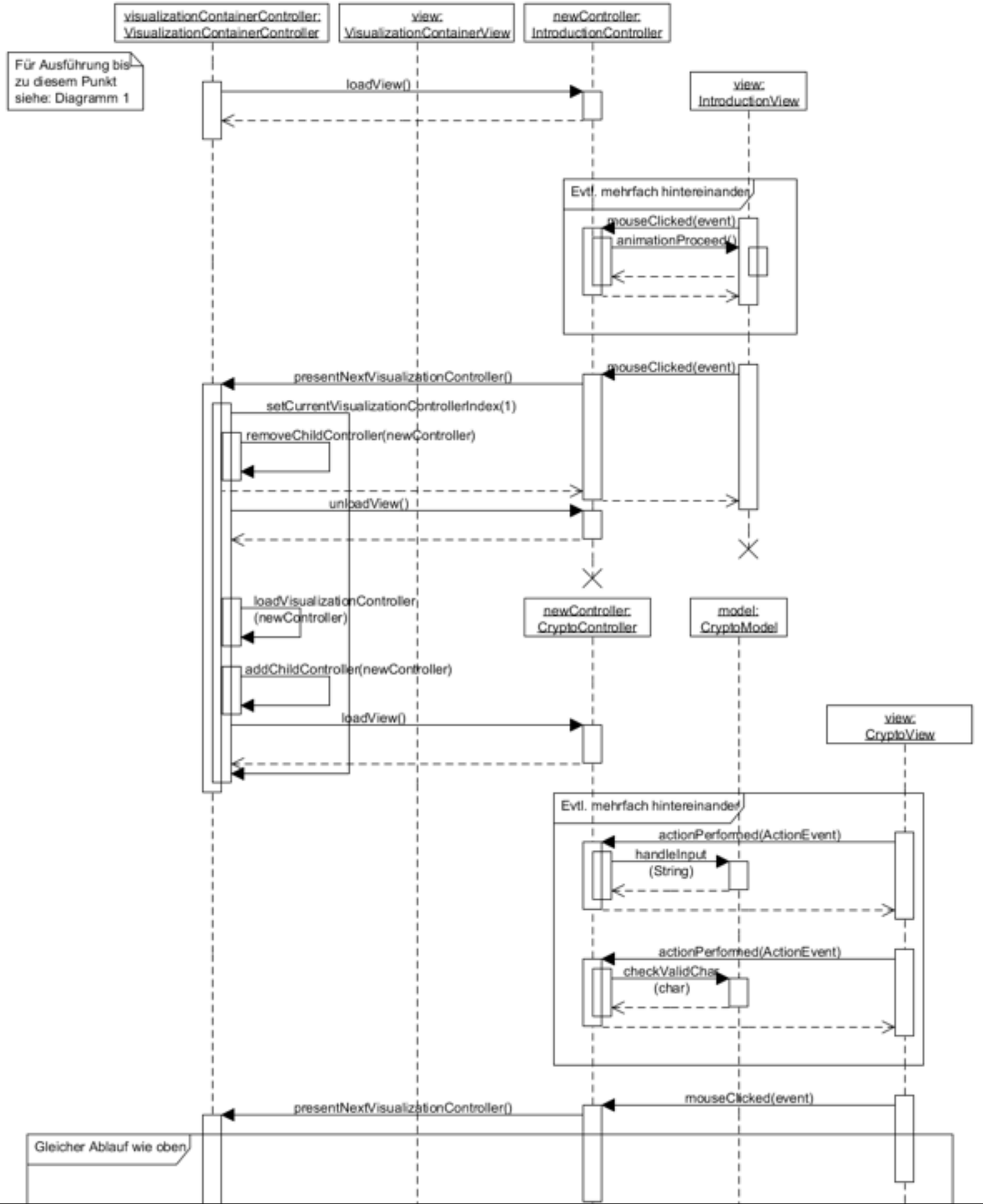

<<Java Class>>
AbstractVisualizationInfo
edu.kit.iks.CryptographicsLib
AbstractVisualizationInfo()
getId():String
getName():String
getDescription():String
getTimelineOffset():float
getDifficulty():VisualizationDifficulty
getYear():int
getQRCodeContent():String
getControllerClasses():List<Class>
getAdditionalInformationFileURL():URL
getQrCode():Image
-visualizationInfo
0..1
<<Java Class>>
AbstractVisualizationController
edu.kit.iks.CryptographicsLib
AbstractVisualizationController(AbstractVisualizationInfo)
getVisualizationInfo():AbstractVisualizationInfo
getHelp():String
<<Java Class>>
CaesarVisualizationInfo
edu.kit.iks.Cryptographics.Caesar
CaesarVisualizationInfo()
getId():String
getName():String
getDescription():String
getTimelineOffset():float
getDifficulty():VisualizationDifficulty
getYear():int
getControllerClasses():List<Class>
getQRCodeContent():String
<<Java Class>>
CipherDemoController
edu.kit.iks.Cryptographics.Caesar.Demonstration
CipherDemoController(AbstractVisualizationInfo)
loadView():void
proceedAnimation():void
getHelp():String
getView():JComponent
<<Java Class>>
IntroductionController
edu.kit.iks.Cryptographics.Caesar.Demonstration
step: int
IntroductionController(AbstractVisualizationInfo)
getView():JComponent
getHelp():String
loadView():void
animationProceed():void
<<Java Class>>
CryptoController
edu.kit.iks.Cryptographics.Caesar.Experiment
CryptoController(AbstractVisualizationInfo)
loadView():void
delegateAndHandleInput(String):void
getHelp():String
getView():JComponent
getModel():CryptoModel
proceed():void
<<Java Class>>
HistogramController
edu.kit.iks.Cryptographics.Caesar.Experiment
HistogramController(AbstractVisualizationInfo)
getHelp():String
loadView():void
getView():JComponent
lastExperiment():void
-view
0..1
-view
0..1
-view
0..1
-model
0..1
-view
0..1
<<Java Class>>
CipherDemoView
edu.kit.iks.Cryptographics.Caesar.Demonstration
serialVersionUID: long
explanations: JLabel
input: JTextField
userInput: JLabel[]
userOutput: JLabel[]
alphabet: JLabel[]
alphabetNumbering: JLabel[]
proceed: JButton
CipherDemoView()
animationStart():void
step1():void
step2():void
step3():void
step4():void
getProceed():JButton
getExplanations():JLabel
getInput():JTextField
getUserInput():JLabel[]
getUserOutput():JLabel[]
getAlphabet():JLabel[]
getAlphabetNumbering():JLabel[]
<<Java Class>>
IntroductionView
edu.kit.iks.Cryptographics.Caesar.Demonstration
serialVersionUID: long
caesar: Image
cipher: Image
interceptor: Image
proceed: JButton
explanation: JLabel
IntroductionView()
animationStart():void
getProceed():JButton
step1():void
step2():void
step3():void
step4():void
getExplanation():JLabel
setExplanation(JLabel):void
getCaesar():Image
setCaesar(Image):void
getCipher():Image
setCipher(Image):void
getInterceptor():Image
setInterceptor(Image):void
<<Java Class>>
CryptoView
edu.kit.iks.Cryptographics.Caesar.Experiment
serialVersionUID: long
explanations: JLabel
input: JTextField
userInput: JLabel[]
userOutput: JLabel[]
alphabet: JLabel[]
alphabetNumbering: JLabel[]
CryptoView()
lightsOn(JLabel):void
setContent(JLabel,char):void
insertInput(String):void
getInput():JTextField
getUserInput():JLabel[]
setUserInput(JLabel[]):void
getUserOutput():JLabel[]
setUserOutput(JLabel[]):void
getAlphabet():JLabel[]
setAlphabet(JLabel[]):void
getExplanations():JLabel
setExplanations(JLabel):void
<<Java Class>>
CryptoModel
edu.kit.iks.Cryptographics.Caesar.Experiment
inputStep: boolean
CryptoModel()
checkValidChar(char):boolean
handleInput(String):boolean
isInputStep():boolean
setInputStep(boolean):void
decryptAndCheck(String,String):boolean
<<Java Class>>
HistogramView
edu.kit.iks.Cryptographics.Caesar.Experiment
serialVersionUID: long
increment: JButton
decrement: JButton
cipher: JLabel[]
explanations: JLabel
key: JLabel
histogram: CharacterFrequencyDiagramView
getHistogram():CharacterFrequencyDiagramView
HistogramView()
startAnimations():void
step1():void
step2():void
step3():void
step4():void
setContent(JLabel,char):void
getCipher():JLabel[]
setCipher(JLabel[]):void
getExplanations():JLabel
setExplanations(JLabel):void
getKey():JLabel
setKey(JLabel):void
getIncrement():JButton
getDecrement():JButton


«Java Class»
AbstractVisualizationInfo
edu.kit.iks.CryptographicsLib
AbstractVisualizationInfo()
getId():String
getName():String
getDescription():String
getTimelineOffset():float
getDifficulty():VisualizationDifficulty
getYear():int
getQRCodeContent():String
getControllerClasses():List<Class>
getAdditionalInformationFile(URL):URL
getQrCode():Image
visualizationInfo
0..1
«Java Class»
AbstractVisualizationController
edu.kit.iks.CryptographicsLib
AbstractVisualizationController(AbstractVisualizationInfo)
getVisualizationInfo():AbstractVisualizationInfo
getHelp():String
«Java Class»
AbstractController
edu.kit.iks.Cryptographics.Vigenere
AbstractController(VigenereVisualizationInfo)
getHelp():String
loadView():void
«Java Class»
VigenereVisualizationInfo
edu.kit.iks.Cryptographics.Vigenere
VigenereVisualizationInfo()
getId():String
getName():String
getDescription():String
getQRCodeContent():String
getTimelineOffset():float
getDifficulty():VisualizationDifficulty
getYear():int
getControllerClasses():List<Class>
«Java Class»
FirstDemonstrationController
edu.kit.iks.Cryptographics.Vigenere.Demonstration
FirstDemonstrationController(VigenereVisualizationInfo)
«Java Class»
SecondDemonstrationController
edu.kit.iks.Cryptographics.Vigenere.Demonstration
SecondDemonstrationController(VigenereVisualizationInfo)
«Java Class»
ThirdDemonstrationController
edu.kit.iks.Cryptographics.Vigenere.Demonstration
ThirdDemonstrationController(VigenereVisualizationInfo)
«Java Class»
FirstExperimentController
edu.kit.iks.Cryptographics.Vigenere.Experiment
FirstExperimentController(VigenereVisualizationInfo)
«Java Class»
SecondExperimentController
edu.kit.iks.Cryptographics.Vigenere.Experiment
SecondExperimentController(VigenereVisualizationInfo)
«Java Class»
FirstExplanationController
edu.kit.iks.Cryptographics.Vigenere.Explanation
FirstExplanationController(VigenereVisualizationInfo)
«Java Class»
SecondExplanationController
edu.kit.iks.Cryptographics.Vigenere.Explanation
SecondExplanationController(VigenereVisualizationInfo)
view
0..1
«Java Class»
FirstDemonstrationView
edu.kit.iks.Cryptographics.Vigenere.Demonstration
serialVersionUID: long
clock: ImageView
moduloText: JLabel
FirstDemonstrationView()
«Java Class»
SecondDemonstrationView
edu.kit.iks.Cryptographics.Vigenere.Demonstration
serialVersionUID: long
indexCharPlain: JLabel[]
textCharPlain: JTextField[]
indexCharEncrypted: JLabel[]
textCharEncrypted: JTextField[]
vigenereKeyDesc: JLabel
charFirst: JLabel
charSecond: JLabel
charFinished: JLabel
SecondDemonstrationView()
«Java Class»
ThirdDemonstrationView
edu.kit.iks.Cryptographics.Vigenere.Demonstration
serialVersionUID: long
indexCharPlain: JLabel[]
textCharPlain: JTextField[]
indexCharEncrypted: JLabel[]
textCharEncrypted: JTextField[]
vigenereKeyDesc: JLabel
charFirst: JLabel
charSecond: JLabel
charFinished: JLabel
ThirdDemonstrationView()
«Java Class»
FirstExperimentView
edu.kit.iks.Cryptographics.Vigenere.Experiment
serialVersionUID: long
indexCharPlain: JLabel[]
textCharPlain: JTextField[]
indexCharEncrypted: JLabel[]
textCharEncrypted: JTextField[]
vigenereKeyDesc: JLabel
charFirst: JLabel
charSecond: JLabel
charFinished: JLabel
FirstExperimentView()
«Java Class»
SecondExperimentView
edu.kit.iks.Cryptographics.Vigenere.Experiment
serialVersionUID: long
indexCharEncrypted: JLabel[]
textCharEncrypted: JTextField[]
indexCharDecrypt: JLabel[]
textCharDecrypt: JTextField[]
vigenereKeyDesc: JLabel
charFirst: JLabel
charSecond: JLabel
charFinished: JLabel
SecondExperimentView()
«Java Class»
FirstExplanationView
edu.kit.iks.Cryptographics.Vigenere.Explanation
serialVersionUID: long
vigenereWealthExplanation: JLabel
vigenereHistogramm: ImageView
FirstExplanationView()
«Java Class»
SecondExplanationView
edu.kit.iks.Cryptographics.Vigenere.Explanation
serialVersionUID: long
cipher: JLabel
keylength: JLabel
key: JTextField
buttonValidateKey: JButton
buttonGenerateHistogramm: JButton
buttonComputeKeyLength: JButton
vigenereHistogramm: CharacterFrequencyDiagramView
alphabetHistogramm: CharacterFrequencyDiagramView
SecondExplanationView()


«Java Class»
AbstractVisualizationInfo
info: All the Cryptographics Differentiation
+ AbstractVisualizationInfo()
+ getKey(): String
+ getName(): String
+ getDescription(): String
+ getCreationYear(): float
+ getDifficulty(): VisualizationDifficulty
+ getYear(): int
+ getQR(colorCode:int): String
+ getController(): Class<CryptoController>
+ getAdditionalInformation(): Map<String, String>
+ getQC(color): Image
«Java Class»
QRVisualizationInfo
info: All the Cryptographics Differentiation
+ QRVisualizationInfo()
+ getKey(): String
+ getName(): String
+ getDescription(): String
+ getCreationYear(): float
+ getDifficulty(): VisualizationDifficulty
+ getYear(): int
+ getController(): Class<QRController>
+ getQR(colorCode:int): String
«Java Class»
AbstractVisualizationController
info: All the Cryptographics Differentiation
+ AbstractVisualizationController(visualizationInfo:VisualizationInfo)
+ getVisualizationInfo(): AbstractVisualizationInfo
+ getKey(): String
«Java Class»
Model
info: All the Cryptographics Differentiation
- alicePublicColor : Color
- alicePrivateColor : Color
- aliceMixedColor : Color
- sharedColor : Color
- bobPublicColor : Color
- bobPrivateColor : Color
- bobMixedColor : Color
+ Model()
+ getInstance() : Model
+ getAlicePublicColor(): Color
+ setAlicePublicColor(color:Color): void
+ getAlicePrivateColor(): Color
+ setAlicePrivateColor(color:Color):void
+ getAliceMixedColor(): Color
+ setAliceMixedColor(color:Color): void
+ getBobPublicColor(): Color
+ setBobPublicColor(color:Color):void
+ getBobPrivateColor(): Color
+ setBobPrivateColor(color:Color):void
+ getBobMixedColor(): Color
+ setBobMixedColor(color:Color):void
+ resetAllPrivateAndPublic():void
+ resetBobPrivateAndPublic():void
+ resetAll():void
«Java Class»
AbstractController
info: All the Cryptographics Differentiation
+ AbstractController(model:Model, visualizationInfo:VisualizationInfo)
+ getModel(): Model
+ getKey(): String
+ loadViews(): void
«Java Class»
AliceChooseSecretController
info: All the Cryptographics Differentiation (Demonstration)
+ AliceChooseSecretController(model:Model, visualizationInfo:VisualizationInfo)
+ getKey(): String
+ loadViews(): void
«Java Class»
BobChooseSecretController
info: All the Cryptographics Differentiation (Demonstration)
+ BobChooseSecretController(model:Model, visualizationInfo:VisualizationInfo)
+ getKey(): String
+ loadViews(): void
«Java Class»
ExplainAlmController
info: All the Cryptographics Differentiation (Demonstration)
+ ExplainAlmController(model:Model, visualizationInfo:VisualizationInfo)
+ getKey(): String
+ loadViews(): void
«Java Class»
ExplainKeyExchangeController
info: All the Cryptographics Differentiation (Demonstration)
+ ExplainKeyExchangeController(model:Model, visualizationInfo:VisualizationInfo)
+ getKey(): String
+ loadViews(): void
«Java Class»
MixColorController
info: All the Cryptographics Differentiation (Demonstration)
+ MixColorController(model:Model, visualizationInfo:VisualizationInfo)
+ getKey(): String
+ loadViews(): void
«Java Class»
GiveawayController
info: All the Cryptographics Differentiation (Demonstration)
+ GiveawayController(model:Model, visualizationInfo:VisualizationInfo)
+ getKey(): String
+ loadViews(): void
«Java Class»
IChoosePublicColorController
info: All the Cryptographics Differentiation (Demonstration)
+ IChoosePublicColorController(model:Model, visualizationInfo:VisualizationInfo)
+ getKey(): String
+ loadViews(): void
«Java Class»
IChooseSecretColorController
info: All the Cryptographics Differentiation (Demonstration)
+ IChooseSecretColorController(model:Model, visualizationInfo:VisualizationInfo)
+ getKey(): String
+ loadViews(): void
«Java Class»
MixFinalSecretController
info: All the Cryptographics Differentiation (Demonstration)
+ MixFinalSecretController(model:Model, visualizationInfo:VisualizationInfo)
+ getKey(): String
+ loadViews(): void
«Java Class»
SendColorController
info: All the Cryptographics Differentiation (Demonstration)
+ SendColorController(model:Model, visualizationInfo:VisualizationInfo)
+ getKey(): String
+ loadViews(): void
«Java Class»
SendRightColorController
info: All the Cryptographics Differentiation (Demonstration)
+ SendRightColorController(model:Model, visualizationInfo:VisualizationInfo)
+ getKey(): String
+ loadViews(): void
«Java Class»
YourTurnController
info: All the Cryptographics Differentiation (Demonstration)
+ YourTurnController(model:Model, visualizationInfo:VisualizationInfo)
+ getKey(): String
+ loadViews(): void
«Java Class»
AliceChooseSecretView
info: All the Cryptographics Differentiation (Demonstration)
- serialVersionUID : long
+ AliceChooseSecretView()
«Java Class»
BobChooseSecretView
info: All the Cryptographics Differentiation (Demonstration)
- serialVersionUID : long
+ BobChooseSecretView()
«Java Class»
ExplainAlmView
info: All the Cryptographics Differentiation (Demonstration)
- serialVersionUID : long
+ ExplainAlmView()
«Java Class»
ExplainKeyExchangeView
info: All the Cryptographics Differentiation (Demonstration)
- serialVersionUID : long
+ ExplainKeyExchangeView()
«Java Class»
MixColorView
info: All the Cryptographics Differentiation (Demonstration)
- serialVersionUID : long
+ MixColorView()
«Java Class»
GiveawayView
info: All the Cryptographics Differentiation (Demonstration)
- serialVersionUID : long
+ GiveawayView()
«Java Class»
IChoosePublicColorView
info: All the Cryptographics Differentiation (Demonstration)
- serialVersionUID : long
+ IChoosePublicColorView()
«Java Class»
IChooseSecretColorView
info: All the Cryptographics Differentiation (Demonstration)
- serialVersionUID : long
+ IChooseSecretColorView()
«Java Class»
MixFinalSecretView
info: All the Cryptographics Differentiation (Demonstration)
- serialVersionUID : long
+ MixFinalSecretView()
«Java Class»
SendColorView
info: All the Cryptographics Differentiation (Demonstration)
- serialVersionUID : long
+ SendColorView()
«Java Class»
SendRightColorView
info: All the Cryptographics Differentiation (Demonstration)
- serialVersionUID : long
+ SendRightColorView()
«Java Class»
YourTurnView
info: All the Cryptographics Differentiation (Demonstration)
- serialVersionUID : long
+ YourTurnView()
«visualizationInfo»
1..1
«controller»
1..1
«model»
1..1
«view»
1..1

Sequenzdiagramme


:Main
mainController:
MainController
view:
JFrame
startController:
StartController
timelineView:
JPanel
welcomeView:
JPanel
view:
JPanel
popoverView:
TimelinePopoverView
loadView()
presentStartAction()
loadView()
addChildController(startController)
mouseClicked(event)
presentPopoverAction()
loadPopoverVisualizationInfo()
mouseClicked(event)
startVisualizationAction(visualizationInfo)
mouseClicked(event)
unloadView()
removeChildController
(startController)
visualizationContainerController:
VisualizationContainerController
loadView()
view:
VisualizationContainerView
addChildController
(visualizationContainerController)
setCurrentVisualizationControllerIndex(0)
loadVisualizationController(0)
newController:
AbstractVisualizationController
addChildController(newController)
view:
VisualizationView
loadView()
Ausführung Sequenz-
diagramm 2 setzt
an diesem Punkt ein.
mouseClicked(event)
presentStartAction()
Ausführung Sequenz-
diagramm 2 endet
an diesem Punkt.
unloadView()
removeChildController
(visualizationContainerController)


visualizationContainerController:
VisualizationContainerController
view:
VisualizationContainerView
newController:
IntroductionController
Für Ausführung bis
zu diesem Punkt
siehe: Diagramm 1
loadView()
view:
IntroductionView
Evtl. mehrfach hintereinander
mouseClicked(event)
animationProceed()
mouseClicked(event)
presentNextVisualizationController()
setCurrentVisualizationControllerIndex(1)
removeChildController(newController)
unloadView()
loadVisualizationController
(newController)
addChildController(newController)
loadView()
newController:
CryptoController
model:
CryptoModel
view:
CryptoView
Evtl. mehrfach hintereinander
actionPerformed(ActionEvent)
handleInput
(String)
actionPerformed(ActionEvent)
checkValidChar
(char)
presentNextVisualizationController()
mouseClicked(event)
Gleicher Ablauf wie oben

Arbeitsteilung

- prinzipiell haben alle überall mitgeholfen, es gab allerdings eine klare Aufteilung der Zuständigkeiten

- *Christian, Matthias J. und Matthias P.*: edu.kit.iks.Cryptographics und edu.kit.iks.CryptographicsLib

- *Wasilij*: edu.kit.iks.Cryptographics.Caesar

- *Aydin*: edu.kit.iks.Cryptographics.Vigenere

- *Julien*: edu.kit.iks.Cryptographics.DiffieHellman