UNIKASSEL VERSITÄT

SOFTWARE ENGINEERING
UNIVERSITÄT KASSEL

Programmiermethodik

Übung 4

Wintersemester 18 / 19

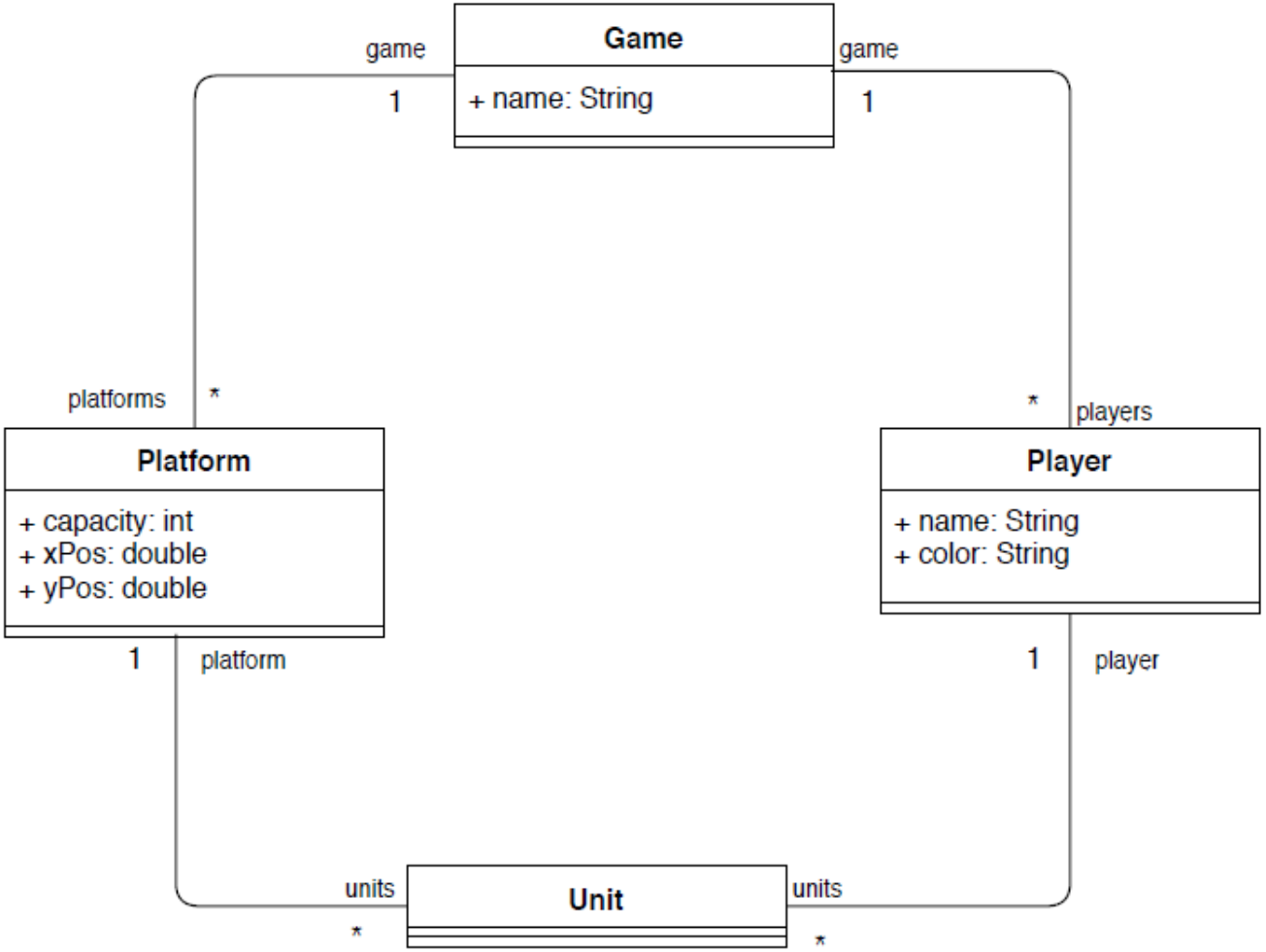
Fachgebiet Software Engineering

Lennert Raesch - Christoph Eickhoff

Agenda

- **Besprechung HA 3**
- **Methodenentwurf**
  - Textueller Entwurf
  - Demo: Implementierung in Java
  - Zetteltest zur Verifikation
- **Vorstellung neue Hausaufgabe**
- **Betreutes Arbeiten: Umgebung einrichten und zusätzliche Infos**
  - Git
  - Eclipse / IntelliJ
  - (UML Lab, Edobs / Visual Debugger)
  - SDMLib
  - Fulib

Hausaufgabe Aufagbe 1



Aktuelle Hausaufgabe Aufgabe 1

```
public class Game {
    private LinkedHashSet<Player> players =
                new LinkedHashSet<Player>();
    public LinkedHashSet<Player> getPlayers() {
      return players;
    }
    public void addPlayer(Player value) {
      if (value  != null && players.add(value)) {
        value.setGame(this);
      }
    }
    public void removePlayer(Player value) {
      if (this.players.remove(value)) {
        value.setGame(null);
      }
    }
    public Game withPlayer (Player value) {
       addPlayer(value);
      return this;
   }
}
```

```
public class Player {
    private Game game;

    public Game getGame() {
      return game;
    }
    public void setGame(Game game) {
          if (this.game != game) {
              if (this.game != null) {
                 this.game.removePlayer(this);
              }
              this.game = game;
              if (this.game != null) {
                  this.game.addPlayer(this);
              }
          }
     }
     public Player withGame(Game game) {
         setGame(game);
        return this;
    }
}
```

Aktuelle Hausaufgabe Aufgabe 2

```
public class ModelTest extends TestCase {
    private Game game;
    @Test
    public void testModel() {
        this.game = new Game().withName("LiveRisk");
        Player alice = new Player().withName("Alice").withColor( "red");
        for (int i = 0; i < 10; i++) {game.addPlayer(new Player());}
        game.addPlayer(alice);
        assertEquals("Alice", alice.getName());
        assertEquals("red", alice.getColor());
        assertEquals("LiveRisk", game.getName());
        assertEquals(11, game.getPlayers().size());
        // referentielle integrität
        assertEquals(game, alice.getGame());

        game.removePlayer(alice);
        assertNull(alice.getGame());
        assertFalse(game.getPlayers().contains(alice));
    }
}
```

Neue Hausaufgabe

- **Deadline: 23.11.2016, 13:00 Uhr**
- **Aufgabe 0 – Vorbereitung (1P)**
- **Aufgabe 1 - Modellierung mit NetworkParser (25P)**
- **Aufgabe 2 - Modellierung mit SDMLib (25P)**
- **Aufgabe 3 - Tests (28P)**
- **Aufgabe 4 – Implementierung (20P)**
- **Aufgabe 5 – Fragebogen ausfüllen (1P) !!!!!!!**
- **Voraussetzungen:**
  - Eine IDE
    - Jetbrain IntelliJ (https://www.jetbrains.com/idea/#chooseYourEdition)
    - Eclipse (http://www.eclipse.org/downloads/)
  - Git (Client deiner Wahl z.B. SourceTree: https://www.sourcetreeapp.com/)
  - NetworkParser / fulib (https://github.com/fujaba/)

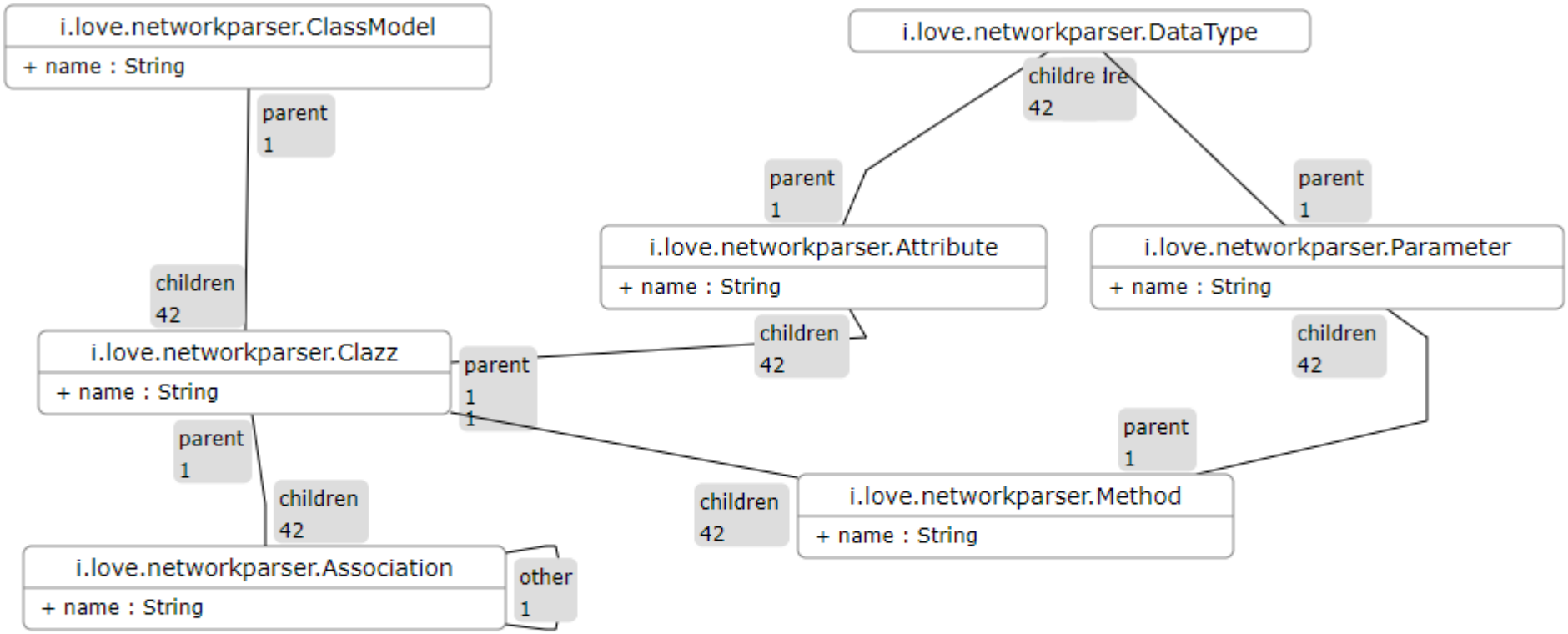
Einfache ClassBuilder (Nur zur Info IN HA 4 Ab Folie 8)

de.uniks.networkparser.ext.ClassModelBuilder

build(String... params) ClassModel
buildClass(String className) Clazz
createClass(String className) ClassModelBuilder
createAssociation(String role, int cardinality, Object otherclazz, String myRole,
int cardinality, Object... clazz) ClassModelBuilder
createAttribute(String name, DataType type, Clazz... clazz) ClassModelBuilder

```
ClassModelBuilder builder = new ClassModelBuilder("de.uniks.model");
Clazz student = builder.buildClass("Student");
builder.createAttribute("name", DataType.STRING)
          .createAttribute("matrikelno", DataType.INT);
builder.createClass("University").createAttribute("name", DataType.STRING);
builder.createAssociation("student",Association.MANY,student,"studs", Association.ONE);
builder.build("gen");
```

Model


i.love.networkparser.ClassModel
+ name : String
parent
1
children
42
i.love.networkparser.Clazz
+ name : String
parent
1
children
42
i.love.networkparser.Association
+ name : String
other
1
parent
1
1
i.love.networkparser.DataType
childre ire
42
parent
1
parent
1
i.love.networkparser.Attribute
+ name : String
children
42
i.love.networkparser.Parameter
+ name : String
children
42
parent
1
children
42
i.love.networkparser.Method
+ name : String

API 191 Methods 1/3

ClassModel

add(Object... p0) boolean
createClazz(String p0) Clazz
dumpHTML(String p0) boolean
fixClassModel() boolean
generate(String... p0) GraphModel
getAnnotation() Annotation
getAssociations(Condition... p0) AssociationSet
getAuthorName() String
getChildByName(String p0, Class p1) GraphMember
getClazz() Clazz
getClazzes(Condition... p0) ClazzSet
getDefaultPackage() String
getFeature(Feature p0, Clazz... p1) Feature
getGenerator(String... p0) ModelGenerator
getId() String
getModifier() Modifier
getName(boolean p0) String
getName() String
getNewList(boolean p0) BaseItem
getValue(String p0) Object
remove(GraphMember p0) boolean
resetGenerator()
setAuthorName(String p0) boolean
size() int
with(Annotation p0) GraphEntity
with(String p0) GraphModel
withFeature(Feature p0) ClassModel
withoutFeature(Feature p0) ClassModel

Clazz

createAttribute(String p0, DataType p1) Attribute
createBidirectional(Clazz p0, String p1, int p2, String p3, int p4) Association
createMethod(String p0, DataType p1, Parameter... p2) Method
createMethod(String p0, Parameter... p1) Method
createUniDirectional(Clazz p0, String p1, int p2) Association
enableEnumeration(Object... p0) Clazz
enableInterface() Clazz
getAnnotation() Annotation
getAssociations(Condition... p0) AssociationSet
getAttributes(Condition... p0) AttributeSet
getChildByName(String p0, Class p1) GraphMember
getClassModel() GraphModel
getClazz() Clazz
getId() String
getImplements() ClazzSet
getImports() SimpleSet
getInterfaces(boolean p0) ClazzSet
getKidClazzes(boolean p0) ClazzSet
getMethods(Condition... p0) MethodSet
getModifier() Modifier
getName(boolean p0) String
getName() String
getSuperClazzes(boolean p0) ClazzSet
getType() String
getValue(String p0) Object
getValues() SimpleSet
remove(GraphMember p0) boolean
setClassModel(GraphModel p0) boolean
with(Annotation p0) Clazz
with(String p0) GraphMember
with(Modifier... p0) Clazz
withAttribute(String p0, DataType p1) Clazz
withBidirectional(Clazz p0, String p1, int p2, String p3, int p4) Clazz
withExternal(boolean p0) GraphEntity
withKidClazzes(Clazz... p0) Clazz
withMethod(String p0, DataType p1, Parameter... p2) Clazz
withSuperClazz(Clazz... p0) Clazz
withUniDirectional(Clazz p0, String p1, int p2) Clazz

API 191 Methods 2/3

Attribute

equals(Object p0) boolean
getAnnotation() Annotation
getClazz() Clazz
getModifier() Modifier
getName() String
getType() DataType
getValue(String p0, boolean p1) String
getValue(String p0) Object
remove(GraphMember p0) boolean
with(String p0) Attribute
with(Annotation p0) Attribute
with(DataType p0) Value
with(Modifier... p0) Attribute
with(String p0, DataType p1) Attribute
withValue(String p0) Value

DataType

create(Object p0) DataType
equals(Object p0) boolean
getClazz() Clazz
getName(boolean p0) String
hashCode() int
withExternal(boolean p0) DataType

Method

create(DataType p0) Parameter
getAnnotation() Annotation
getBody() String
getClazz() Clazz
getModifier() Modifier
getName(boolean p0, boolean p1) String
getName() String
getParameters(Condition... p0) ParameterSet
getReturnType() DataType
getThrows() SimpleSet
getValue(String p0) Object
isValidReturn() boolean
remove(GraphMember p0) boolean
with(Clazz p0) Method
with(DataType p0) Method
with(Throws... p0) Method
with(Parameter... p0) Method
with(String p0) Method
with(Modifier... p0) Method
with(Annotation p0) Method
withBody(String p0) Method
withParameter(String p0, DataType p1) Method
withParent(Clazz p0) Method

Parameter

getClazz() Clazz
getMethod() Method
getModifier() Modifier
getName() String
getType() DataType
getValue() String
getValue(String p0) Object
remove(GraphMember p0) boolean
with(DataType p0) Parameter
with(String p0) GraphMember
withParent(Method p0) Parameter
withValue(String p0) Valueu

API 191 Methods 3/3

Association

create(GraphEntity p0, GraphEntity p1) Association
getCardinality() int
getClazz() Clazz
getInfo() GraphLabel
getModifier() Modifier
getName() String
getOther() Association
getOtherClazz() Clazz
getType() AssociationTypes
getValue(String p0) Object
isSelfAssoc() boolean
remove(GraphMember p0) boolean
with(String p0) GraphMember
with(int p0) Association
with(Association p0) Association
with(Annotation p0) Association
with(AssociationTypes p0) Association
with(GraphEntity p0) Association
with(GraphLabel p0) Association

Annotation

create(String p0) Annotation
decode(BufferItem p0, char p1, Annotation p2) Annotation
decode(String p0) Annotation
getAnnotation(String p0) Annotation
getClazz() Clazz
getModifier() Modifier
getName() String
getParent() GraphMember
getValue() SimpleList
getValue(String p0) Object
hasNext() boolean
newInstance() Annotation
next() Annotation
remove(GraphMember p0) boolean
with(String p0) GraphMember

Throws

getClazz() Clazz
getModifier() Modifier
getName() String
getValue(String p0) Object
remove(GraphMember p0) boolean
with(String p0) GraphMember

Modifier

create(String p0) Modifier
create(Modifier... p0) Modifier
equals(Object p0) boolean
getClazz() Clazz
getModifier() Modifier
getName() String
getParent() GraphMember
getValue(String p0) Object
has(Modifier p0) boolean
hashCode() int
remove(GraphMember p0) boolean
with(String p0) GraphMember

NetworkParser DEMO

```
ClassModel model = new ClassModel("de.uniks.model");
Clazz studentClass = model.createClazz("Student");
studentClass.withAttribute("name", DataType.STRING)
       .createAttribute("matrikelno", DataType.INT);
Clazz uniClass = model.createClazz("University")
       .withAttribute("name", DataType.STRING);
uniClass.createBidirectional(studentClass, "students",
       Association.MANY, "studs", Association.ONE);
model.generate("gen");
```

fulib

org.fulib.Fulib

classModelBuilder(packagename: String , sourceFolder: String)
generator() : Generator

org.fulib.builder.ClassModelBuilder

buildClass(className: String) : ClassBuilder

org.fulib.builder.ClassBuilder

buildAttribute(name: String, type: String)
buildAttribute(name: String, type: String, initialValue : String)
buildAssociation(otherClass : ClassBuilder, myRoleName : String , myCardinality : int ,
otherRoleName: String , otherCardinality: int))

org.fulib.Generator

generate(model: ClassModel)

fulib Example

```
ClassModelBuilder mb = Fulib.classModelBuilder("de.uniks.model", "gen");
ClassBuilder studentClass = mb.buildClass("Student")
        .buildAttribute("name", mb.STRING)
        .buildAttribute("matrikelno", mb.INT);

ClassBuilder uniClass = mb.buildClass("University")
                .buildAttribute("name", mb.STRING);

uniClass.buildAssociation(studentClass, "student", mb.MANY, "studs", mb.ONE);

ClassModel classModel = mb.getClassModel();

Fulib.generator().generate(classModel);
```

Ende

Jetzt: Betreutes Arbeiten

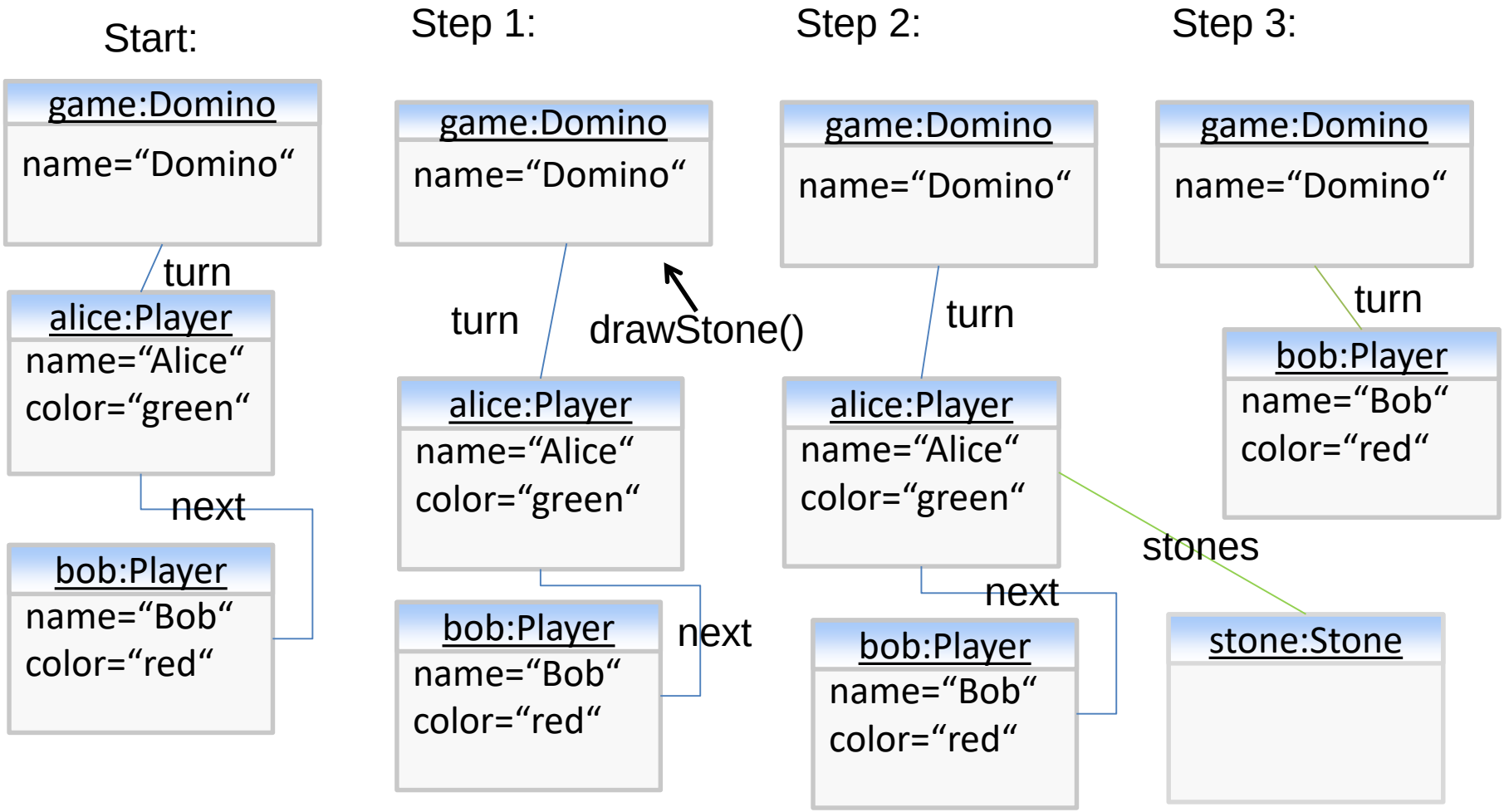
Ansonsten: Schönes WE!

Methodenentwurf I

- **Nach Implementierung des Klassendiagramms erfolgt der Methodenentwurf**
- **Zunächst textuell:**
  - Schritte aufschreiben, die die Methode erledigen soll
  - Bedingungen: „Wenn noch nicht alle Spielsteine verteilt, dann….“
  - Sprünge: „Gehe zu Schritt X“

- **Implementierung: drawStone()**
  - Zieh einen Stein vom Stapel
  - Füge den Stein dem aktuellen Spieler hinzu
  - Der nächste Spieler ist am Zug

Methodenentwurf II



Methodenentwurf III

- **Mögliche Implementierung**

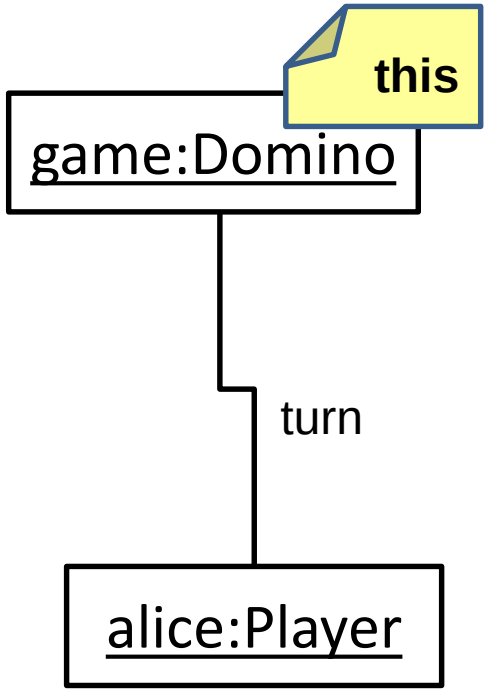
```
1  public void drawStone()
2  {
3        Player player = this.getCurrentPlayer();
4        Stone stone = this.getStones().get(0);
5        player.withStone(stone);
6        player = player.getNext();
7        this.setCurrentPlayer(player);
8  }
```

- **Verifikation durch Zetteltest**
  - Hilft Methoden zu verstehen
  - Fehler aufzudecken
  - Ist „manuelles Debuggen“

Methodenentwurf IV - Zetteltest

- **Mögliche Implementierung**

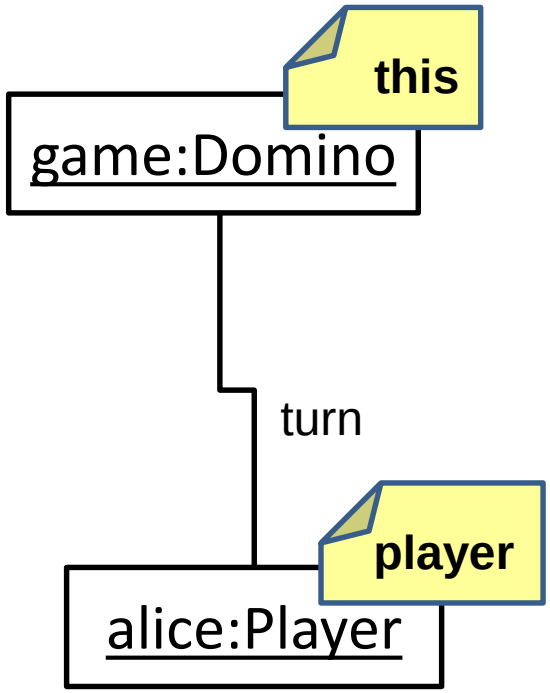
```
1  public void drawStone()
2  {
3      Player player = this.getCurrentPlayer();
4      Stone stone = this.getStones().get(0);
5      player.withStone(stone);
6      player = player.getNext();
7      this.setCurrentPlayer(player);
8  }
```



Methodenentwurf V - Zetteltest

- **Mögliche Implementierung**

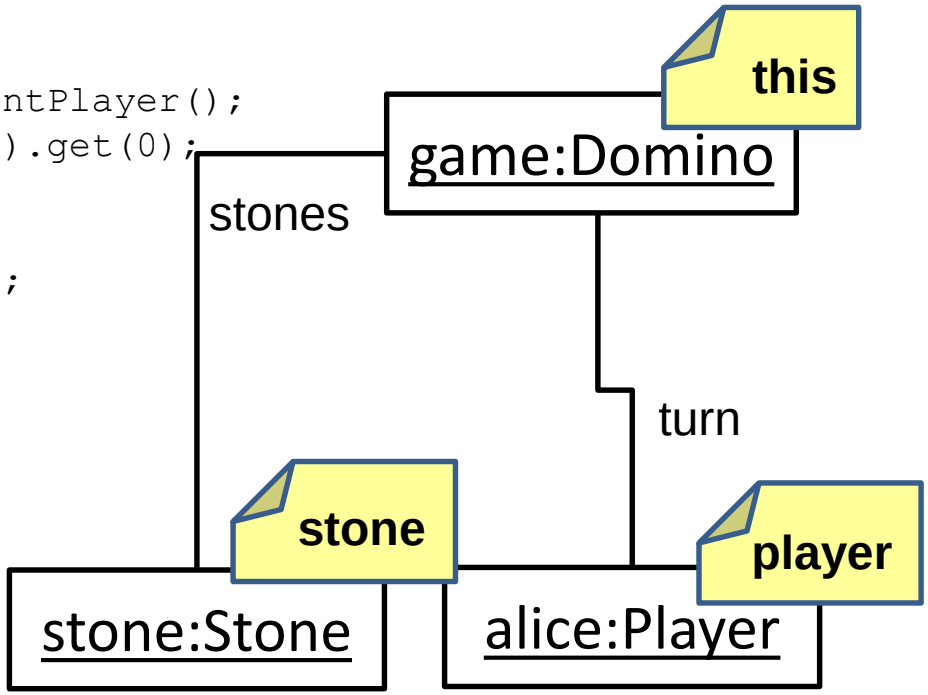
```
1  public void drawStone()
2  {
3        Player player = this.getCurrentPlayer();
4        Stone stone = this.getStones().get(0);
5        player.withStone(stone);
6        player = player.getNext();
7        this.setCurrentPlayer(player);
8  }
```



Methodenentwurf VI - Zetteltest

- **Mögliche Implementierung**

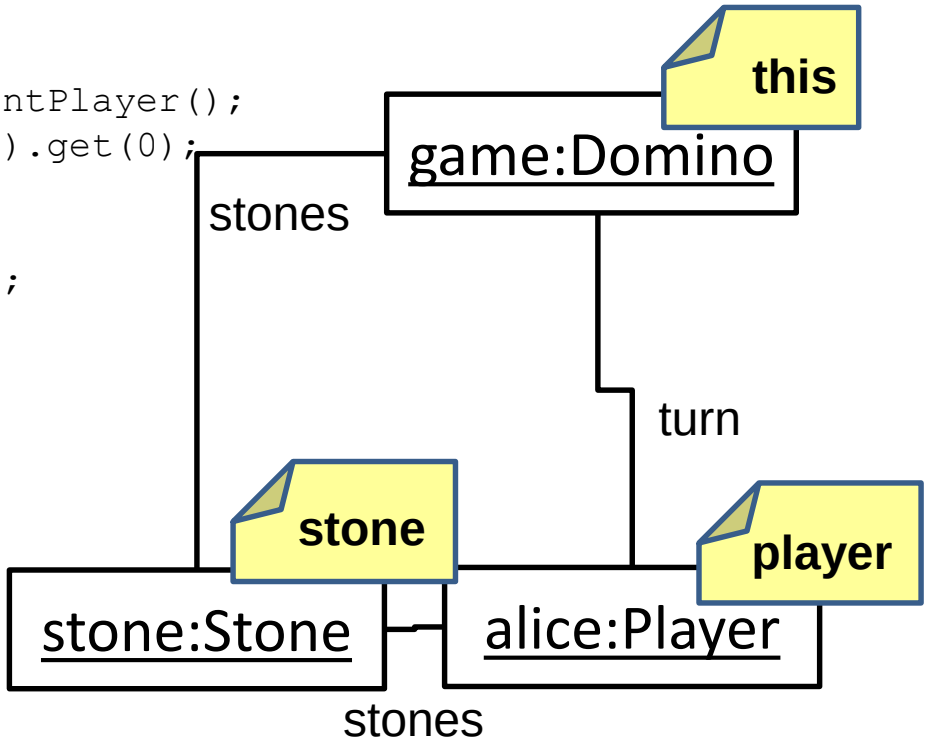
```
1  public void drawStone()
2  {
3        Player player = this.getCurrentPlayer();
4        Stone stone = this.getStones().get(0);
5        player.withStone(stone);
6        player = player.getNext();
7        this.setCurrentPlayer(player);
8  }
```



Methodenentwurf VII - Zetteltest

- **Mögliche Implementierung**

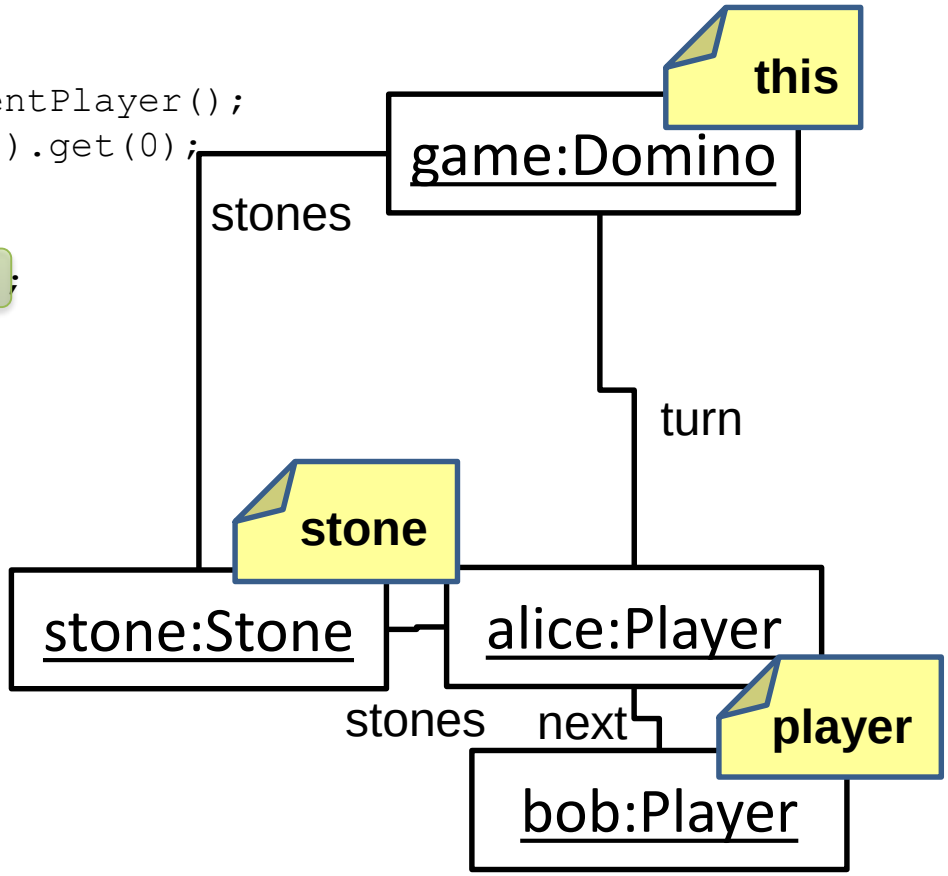
```
1  public void drawStone()
2  {
3        Player player = this.getCurrentPlayer();
4        Stone stone = this.getStones().get(0);
5        player.withStone(stone);
6        player = player.getNext();
7        this.setCurrentPlayer(player);
8  }
```



Methodenentwurf VIII - Zetteltest

- **Mögliche Implementierung**

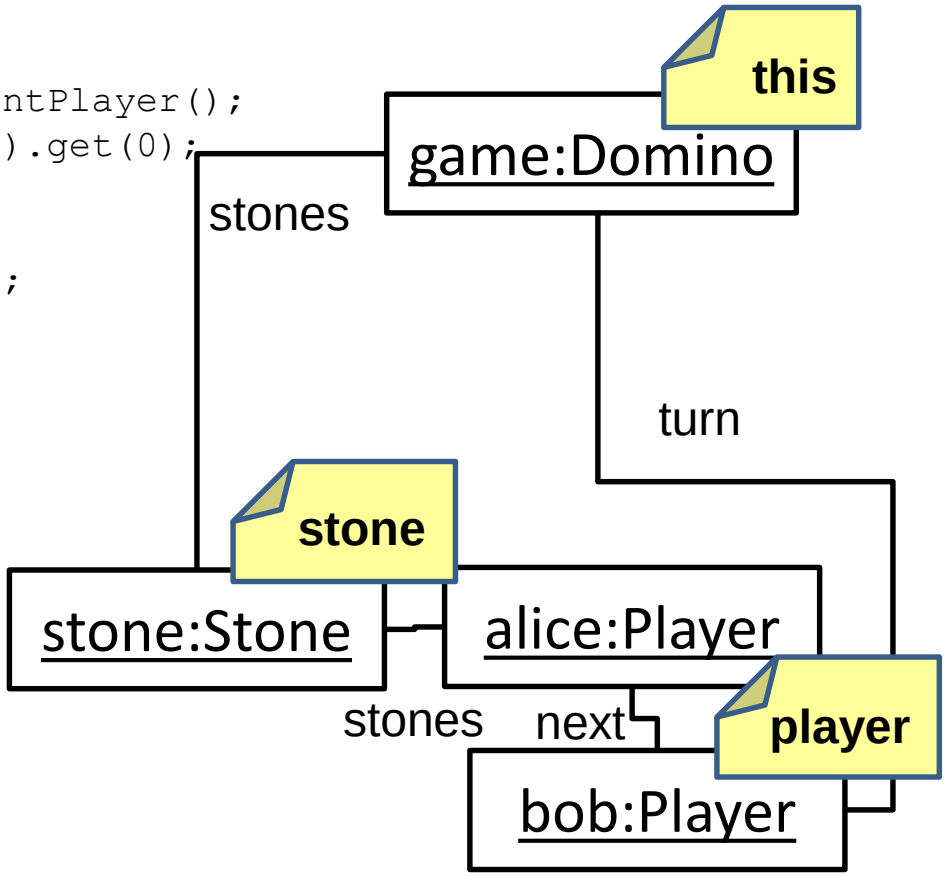
```
1  public void drawStone()
2  {
3        Player player = this.getCurrentPlayer();
4        Stone stone = this.getStones().get(0);
5        player.withStone(stone);
6        player = player.getNext();
7        this.setCurrentPlayer(player);
8  }
```



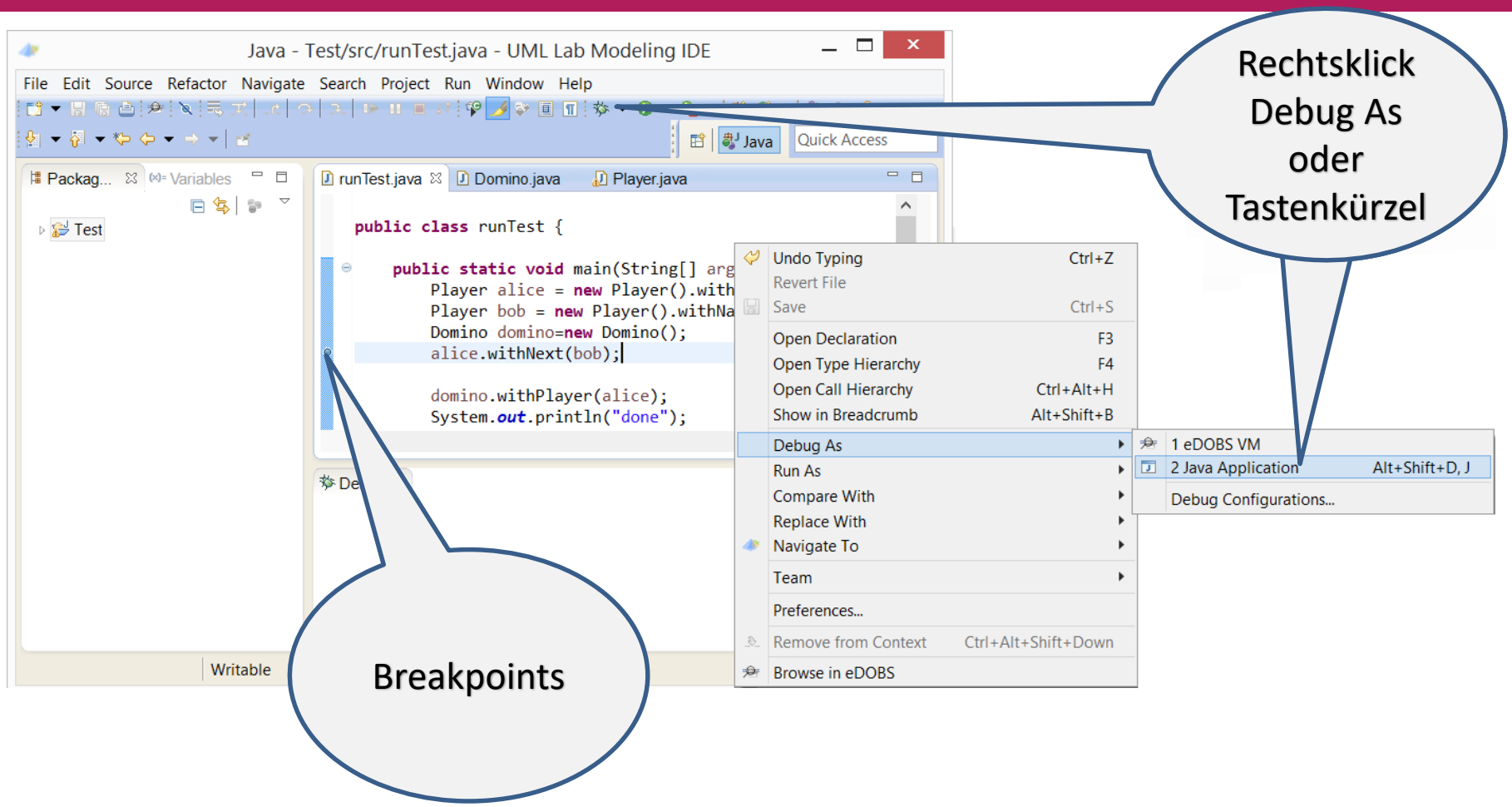
Methodenentwurf IX - Zetteltest

- **Mögliche Implementierung**

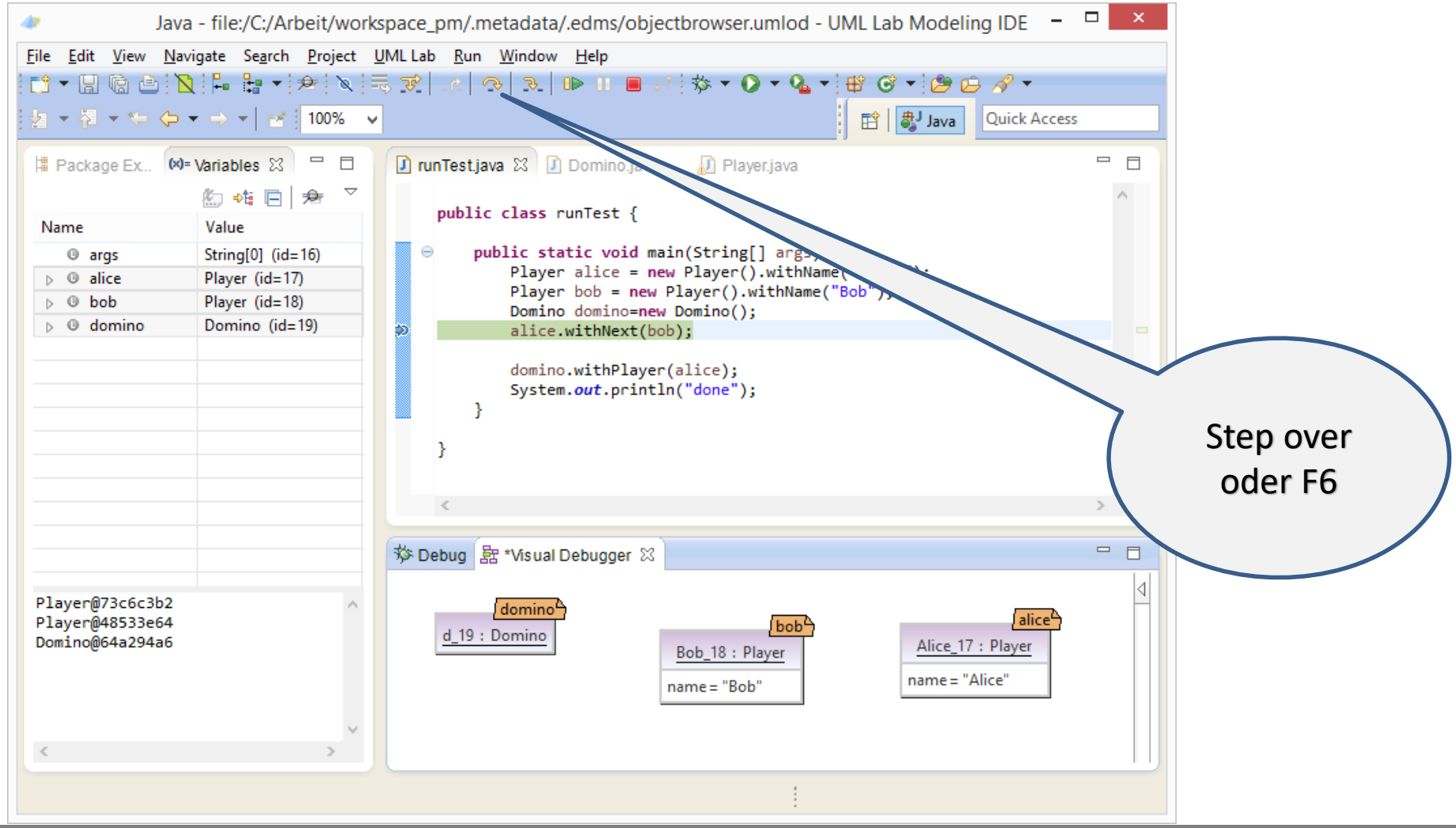
```
1  public void drawStone()
2  {
3        Player player = this.getCurrentPlayer();
4        Stone stone = this.getStones().get(0);
5        player.withStone(stone);
6        player = player.getNext();
7        this.setCurrentPlayer(player);
8  }
```



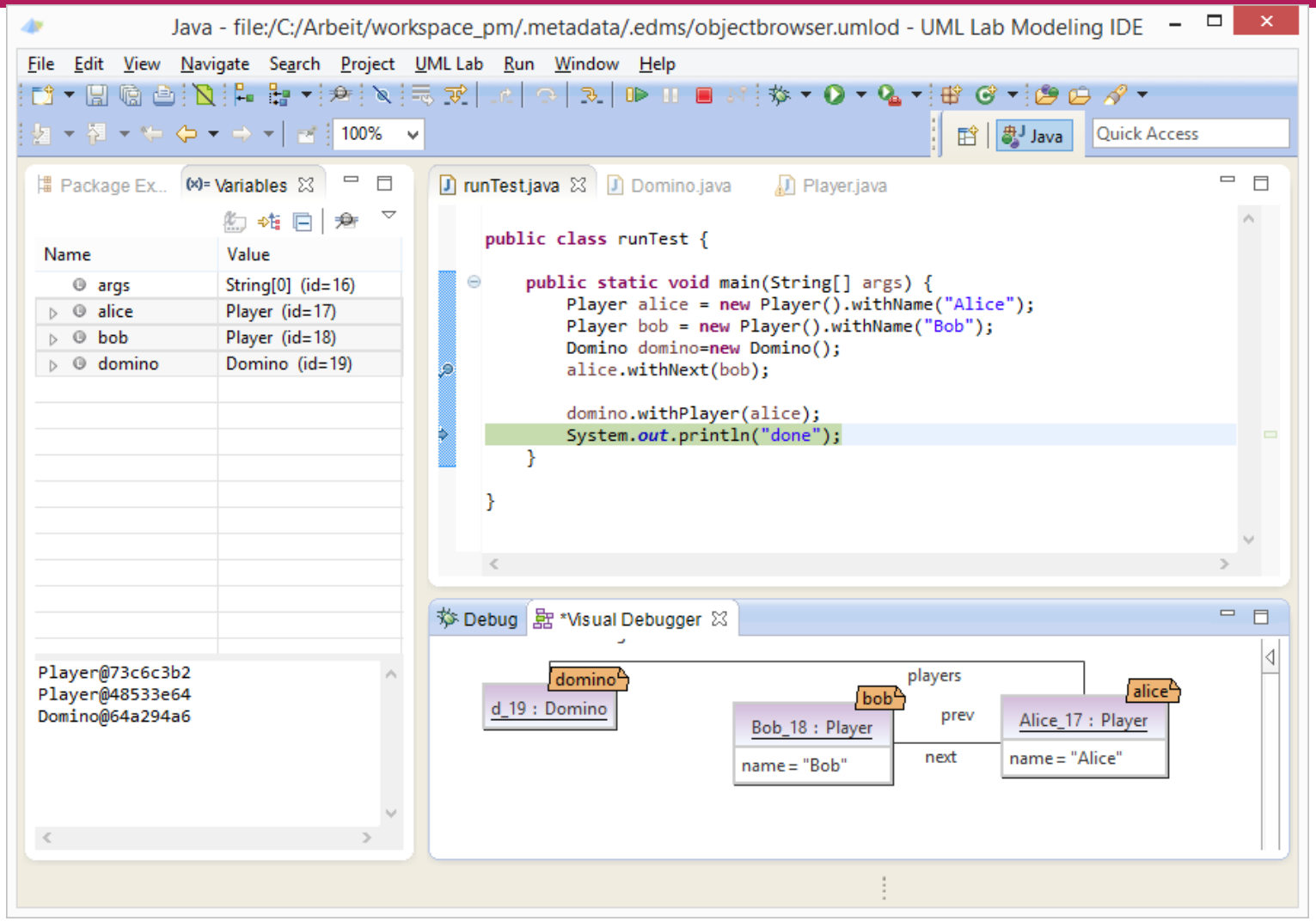
Debuggen


Java - Test/src/runTest.java - UML Lab Modeling IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Java
Quick Access
Packag...
Variables
Test
runTest.java
Domino.java
Player.java
public class runTest {
public static void main(String[] arg
Player alice = new Player().with
Player bob = new Player().withNa
Domino domino=new Domino();
alice.withNext(bob);
domino.withPlayer(alice);
System.out.println("done");
Writable
Undo Typing Ctrl+Z
Revert File
Save Ctrl+S
Open Declaration F3
Open Type Hierarchy F4
Open Call Hierarchy Ctrl+Alt+H
Show in Breadcrumb Alt+Shift+B
Debug As
Run As
Compare With
Replace With
Navigate To
Team
Preferences...
Remove from Context Ctrl+Alt+Shift+Down
Browse in eDOBS
1 eDOBS VM
2 Java Application Alt+Shift+D, J
Debug Configurations...
Rechtsklick
Debug As
oder
Tastenkürzel
Breakpoints

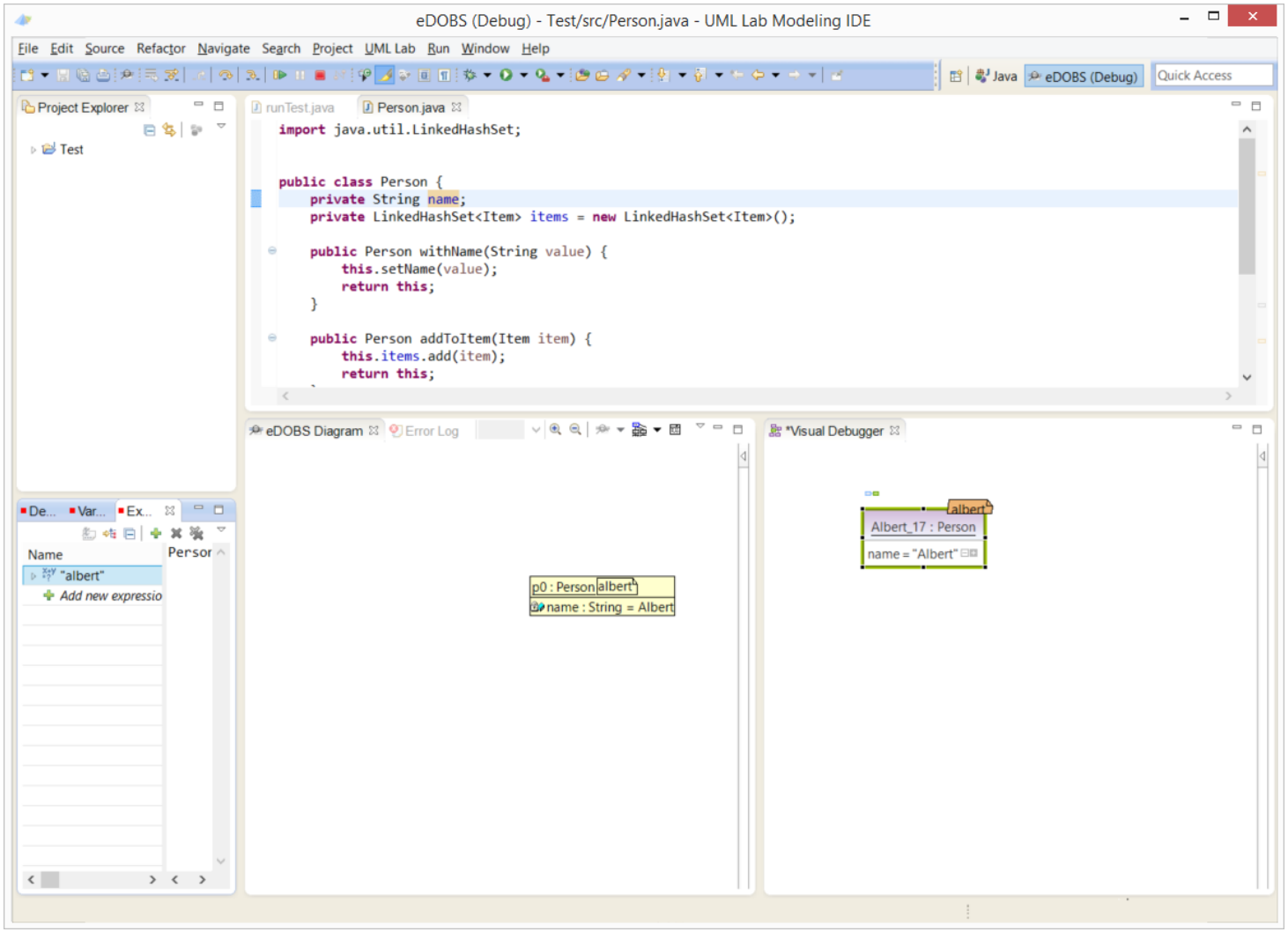
Debuggen/Variablen



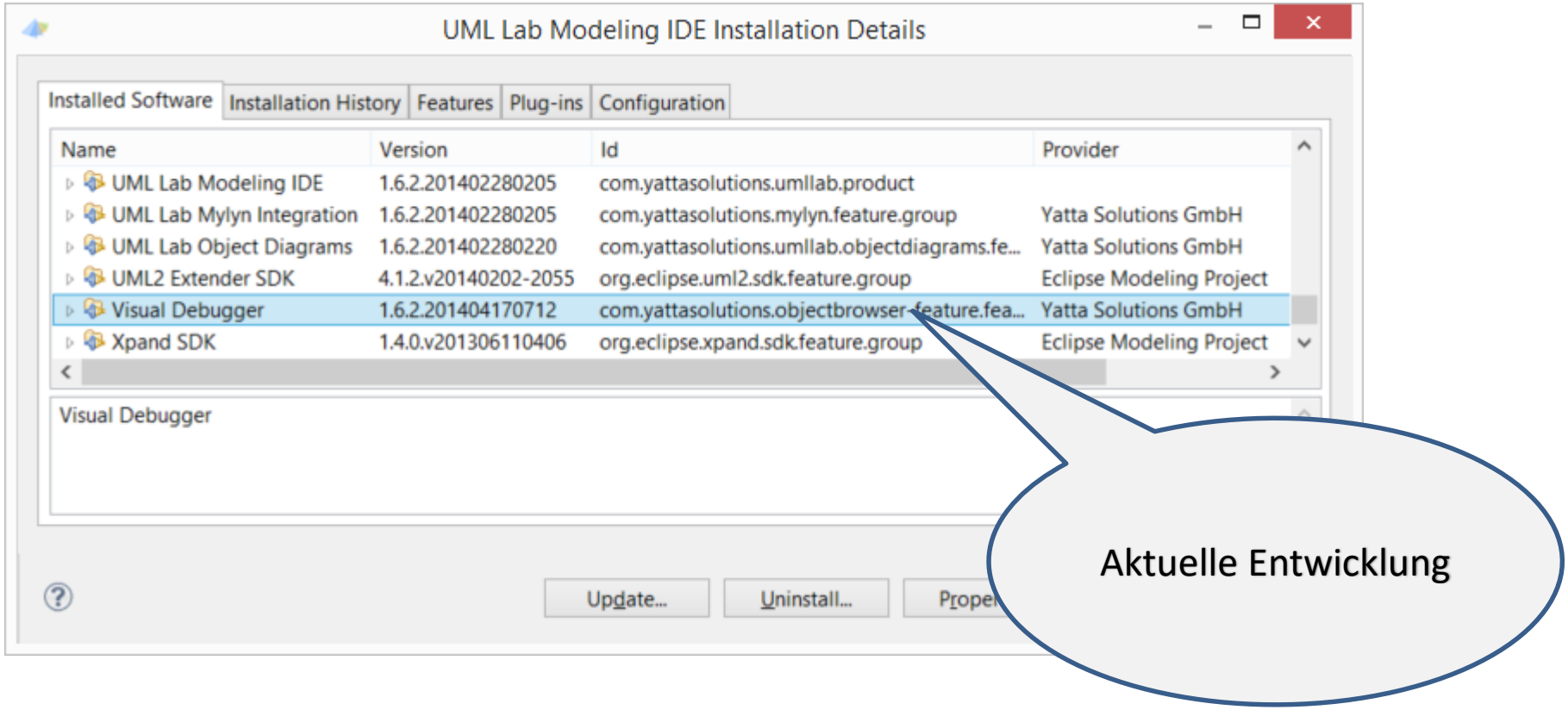
Debuggen/Variablen 2



EDobs / Visual Debugger



Visual Debugger / EDobs



EDobs www.se.eecs.uni-kassel.de/se/fileadmin/se/projects/eDOBS/update