UNIKASSEL VERSITÄT

SOFTWARE ENGINEERING
UNIVERSITÄT KASSEL

Programmiermethodik

Übung 4

Wintersemester 12 / 13

Fachgebiet Software Engineering

Tobias George

george@uni-kassel.de

Agenda

- **Fragen zur aktuellen Hausaufgabe?**
- **Eclipse Live Demo**
- **Methodenentwurf**
  - Textueller Entwurf
  - Praktischer Teil: Implementierung in Java
  - Zetteltest zur Verifikation

Aktuelle Hausaufgabe

```
public class Mancala
{
   …
 private Player currentPlayer;

 public void setCurrentPlayer(Player value)
   {
     if (this.getCurrentPlayer() != value)
     {
       Mancala oldValue = this.getCurrentPlayer();
       if (oldValue != null)
       {
         this.currentPlayer = null;
         oldValue.setCurrentGame(null);
       }
       this.currentPlayer = value;

       if (value != null)
       {
         value.setCurrentGame(this);
       }
     }
   }
…
}
```

```
public class Player
{
   …
   private Mancala currentGame;

   public void setCurrentGame(Mancala value)
   {
     if (this.getCurrentGame() != value)
     {
       Mancala oldValue = this.getCurrentGame();
       if (oldValue != null)
       {
         this.currentGame = null;
         oldValue.setCurrentPlayer(null);
       }
       this.currentGame = value;

       if (value != null)
       {
         value.setCurrentPlayer(this);
       }
     }
   }
…
}
```

Eclipse Live Demo

- **Projekt anlegen, exportieren importieren**
- **Debugging**
- **Reverse Engineering**
- **Refactoring**
- **Plugins**
- **Shortcuts (Cheat Sheet)**
  - http://eclipse-tools.sourceforge.net/EclipseEmacsKeybindings_3_1.pdf

Methodenentwurf I

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

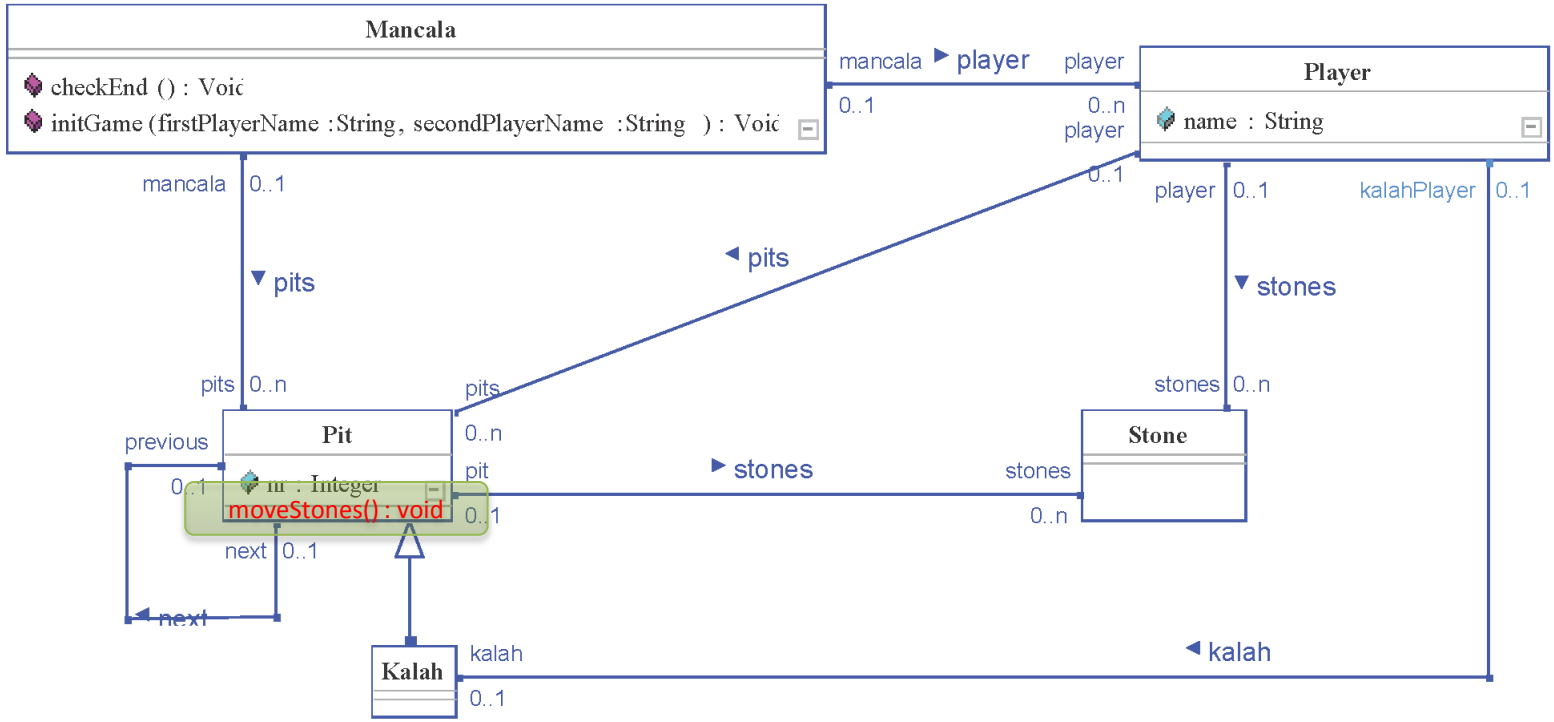
Methodenentwurf II

- **Beispiel: „Mancala“**

http://tinyurl.com/mancalaField

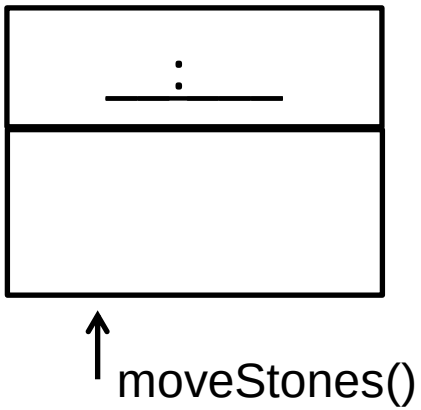
Methodenentwurf III

- **Beispiel: „Mancala“**

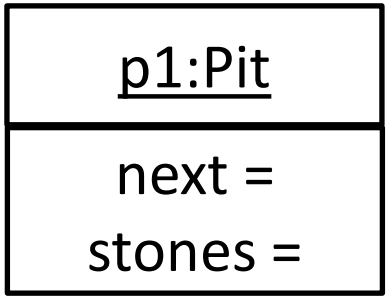


- **Textueller Methodenentwurf für: `moveStones():void` der Klasse `Pit`**

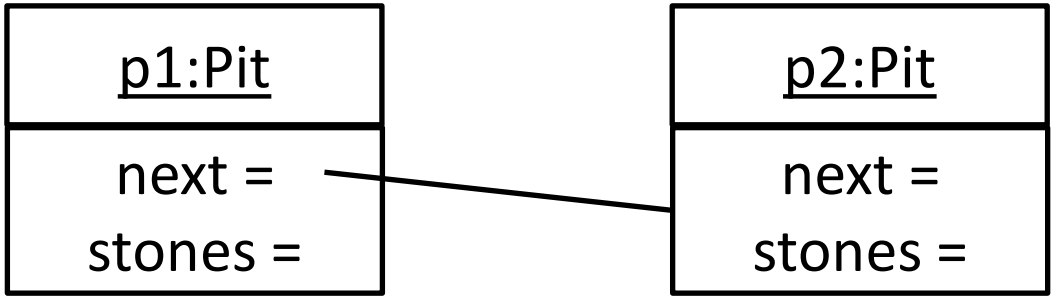
Methodenentwurf IV - Zetteltest



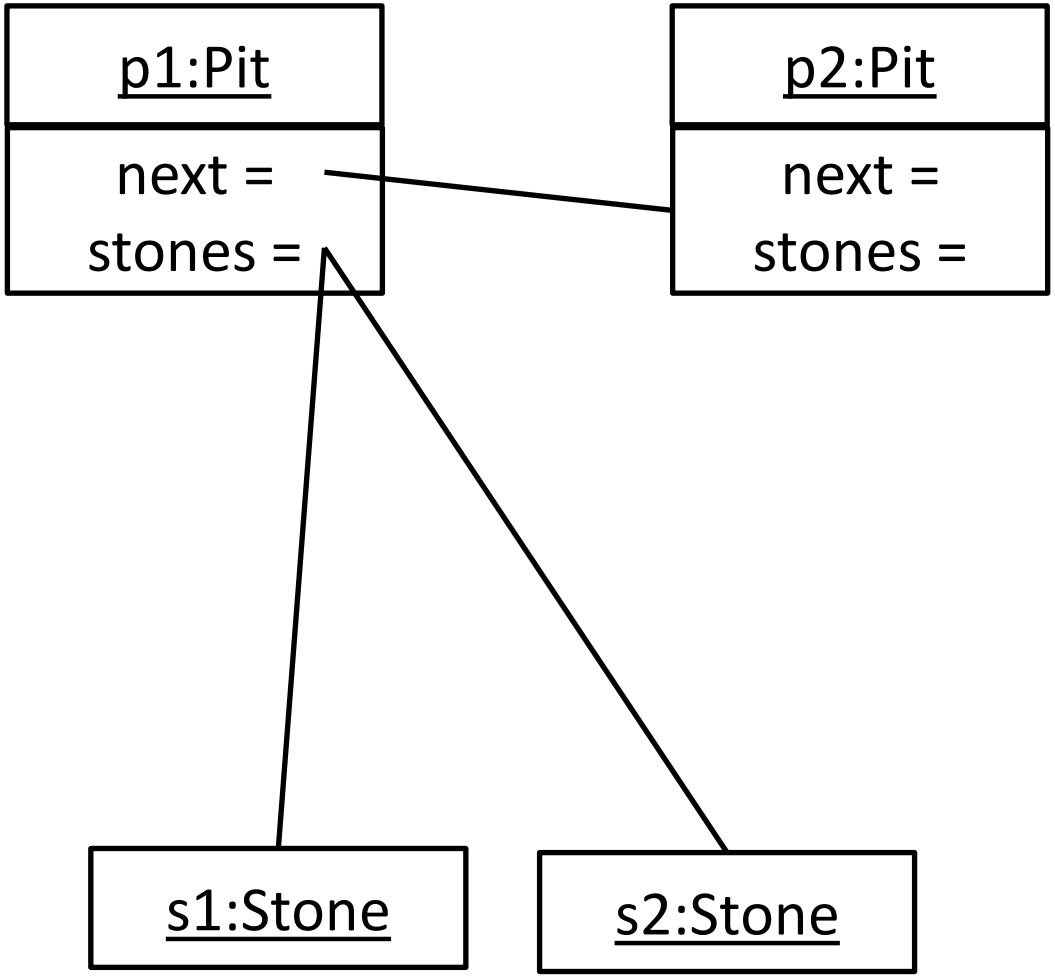
Methodenentwurf V - Zetteltest



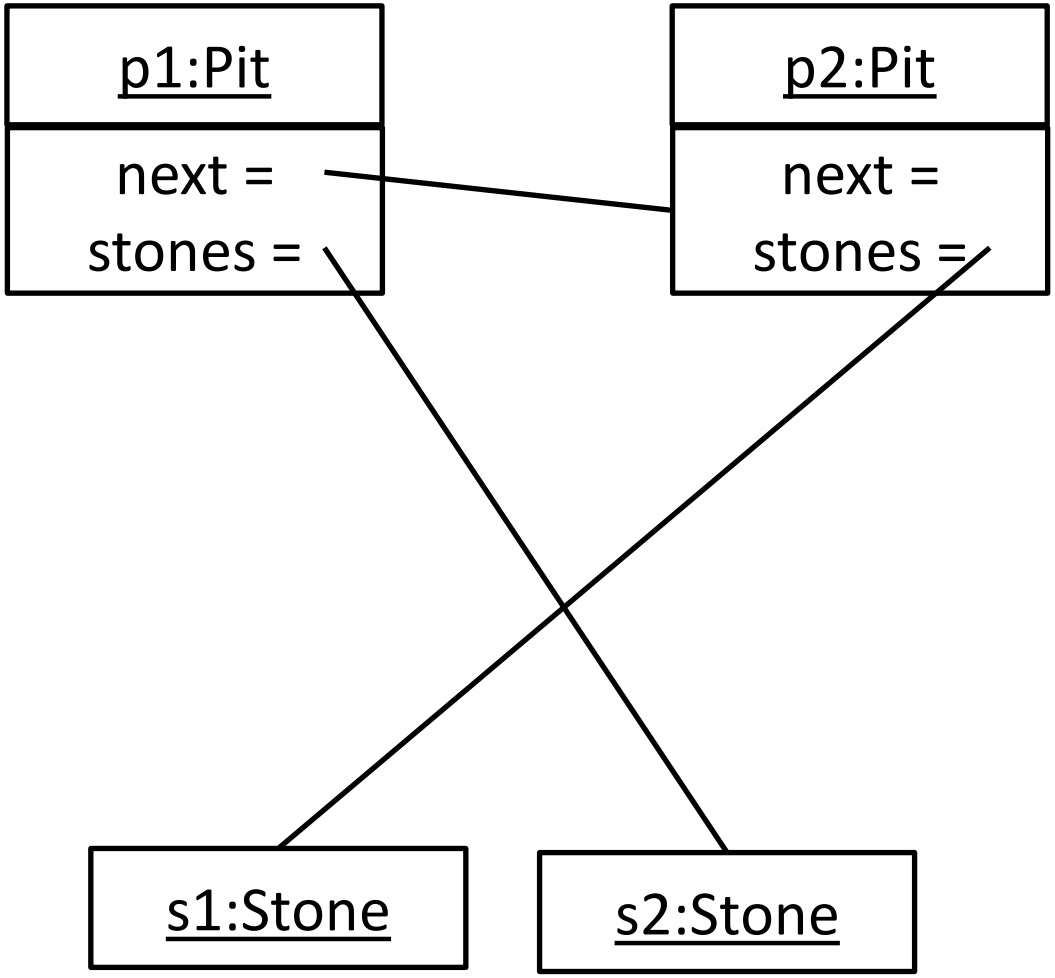
Methodenentwurf VI - Zetteltest



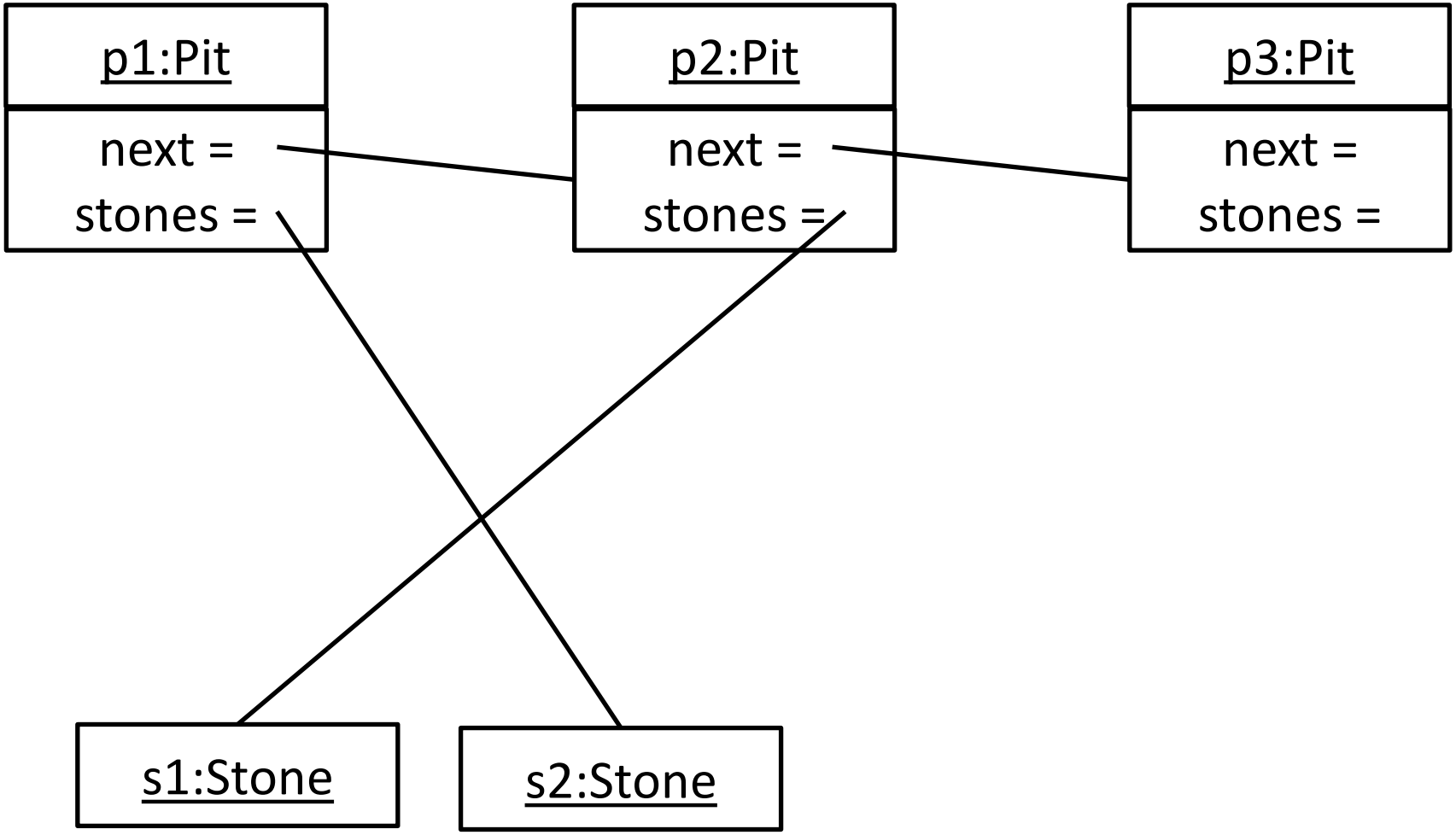
Methodenentwurf VII - Zetteltest



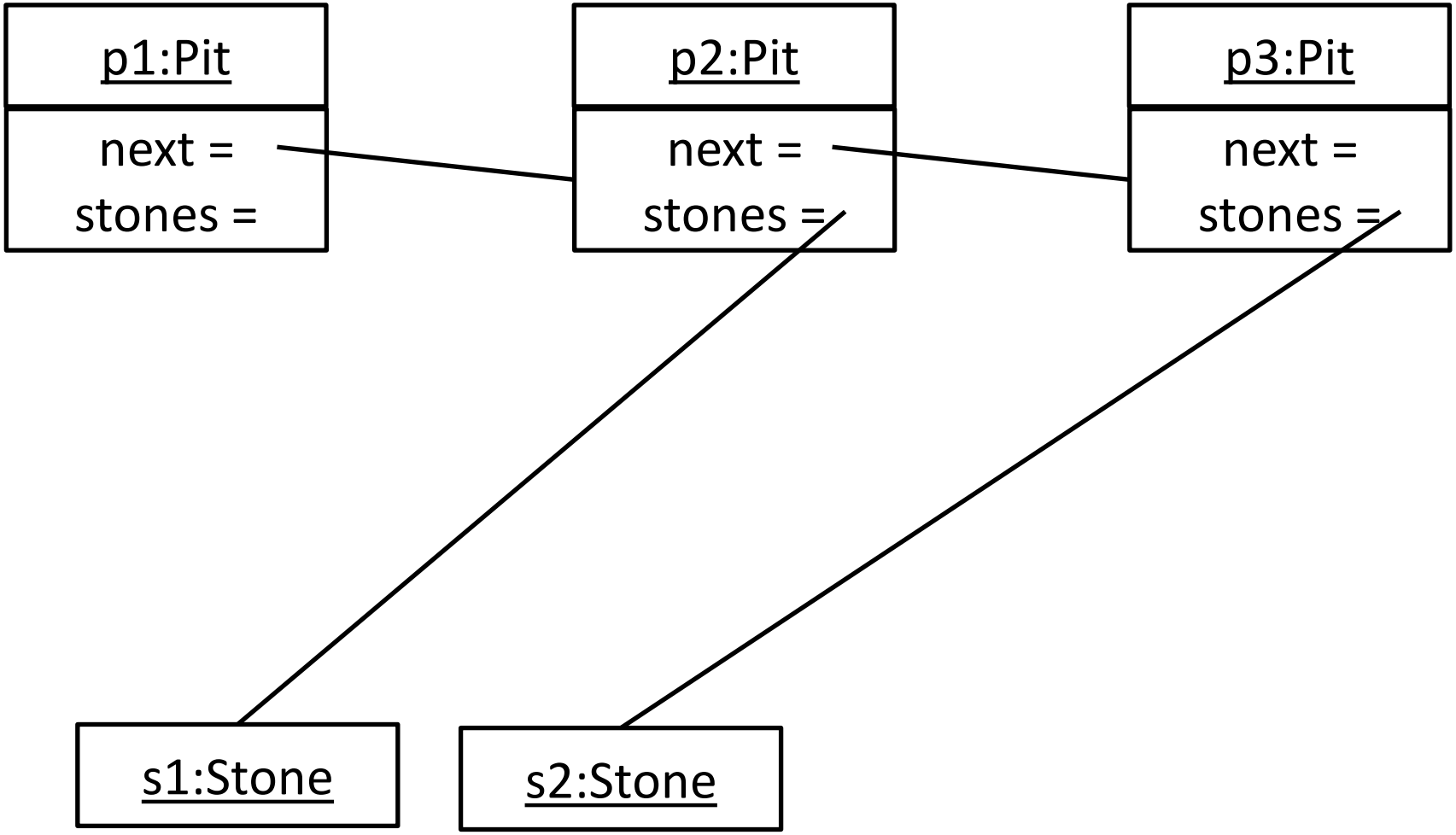
Methodenentwurf VIII - Zetteltest



Methodenentwurf IX - Zetteltest



Methodenentwurf X - Zetteltest



Methodenentwurf XI – Praktische Übung

- **Anforderung:**
  - Stones in einem Pit sollen auf die nachfolgenden Pits verteilt werden
- **Mögliche Lösung:**
  1. Setze Variable „`nextPit`“ auf das nächste Pit
  2. Überprüfe ob Stones in aktuellem Pit vorhanden sind
  3. Für jeden Stein:
     1. Werfe Stein in „`nextPit`“
     2. Setze Variable „`nextPit`“ auf das nächste Pit

Methodenentwurf XII – Praktische Übung

- Implementierung in Java:
- Eclipse Projekt vom Blog herunterladen
  - http://seblog.cs.uni-kassel.de/wp-content/uploads/2012/11/PMWS1213MancalaEclipseProject.zip
- In Eclipse importieren
- JUnit Test schreiben, der:
  - 3 Pits erzeugt und diese miteinander verbindet
  - 1 Player erzeugt
  - 2 Stones in das erste Pit legt
  - Methode `moveStones()` auf dem ersten Pit aufruft
  - Testet, ob in dem ersten Pit kein Stone und in jedem anderen Pit ein Stone liegt
- Methode `moveStones():void` der Klasse `Pit` nach textuellem Entwurf implementieren

Methodenentwurf XIII

- **Mögliche Implementierung**

```
1  public void moveStones ()
2  {
3        Pit nextPit = this.getNext();
4        for(Iterator<Stone> iter = getIteratorOfStones(); iter.hasNext();)
5        {
6                Stone stone = iter.next();
7                nextPit.addToStones(stone);
8                nextPit = nextPit.getNext();
9        }
10 }
```

- **Verifikation durch Zetteltest**

Methodenentwurf XIV - Zetteltest

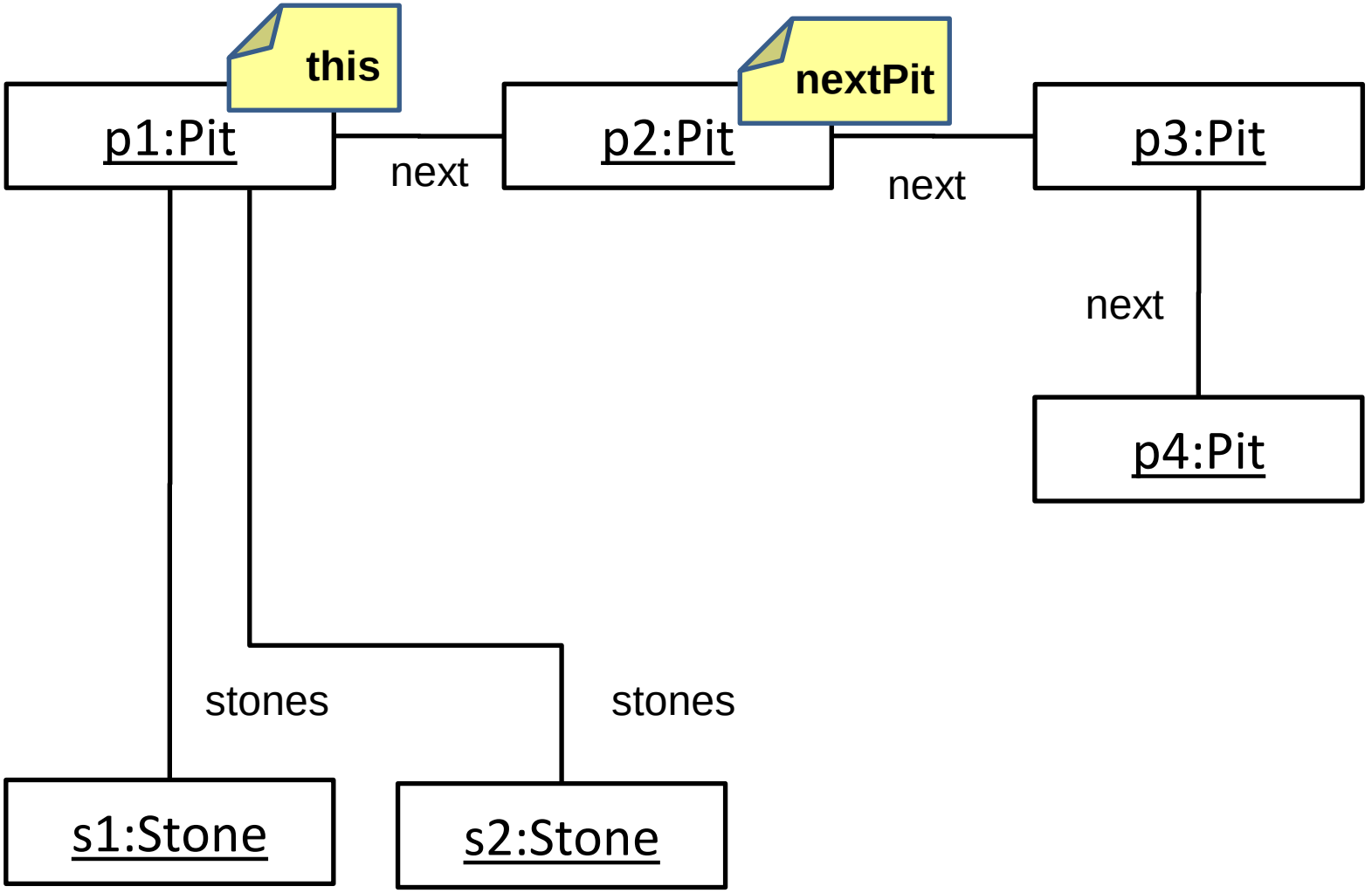
- **Zetteltest**
  - Hilft Methoden zu verstehen
  - Fehler aufzudecken
  - Ist „manuelles Debuggen“
- **Zetteltest Methode `moveStones():void` der Klasse `Pit`**

Methodenentwurf XV - Zetteltest

- **Mögliche Implementierung**

```
1  public void moveStones ()
2  {
3        Pit nextPit = this.getNext();
4        for(Iterator<Stone> iter = getIteratorOfStones(); iter.hasNext();)
5        {
6                Stone stone = iter.next();
7                nextPit.addToStones(stone);
8                nextPit = nextPit.getNext();
9        }
10 }
```

Methodenentwurf XVI - Zetteltest



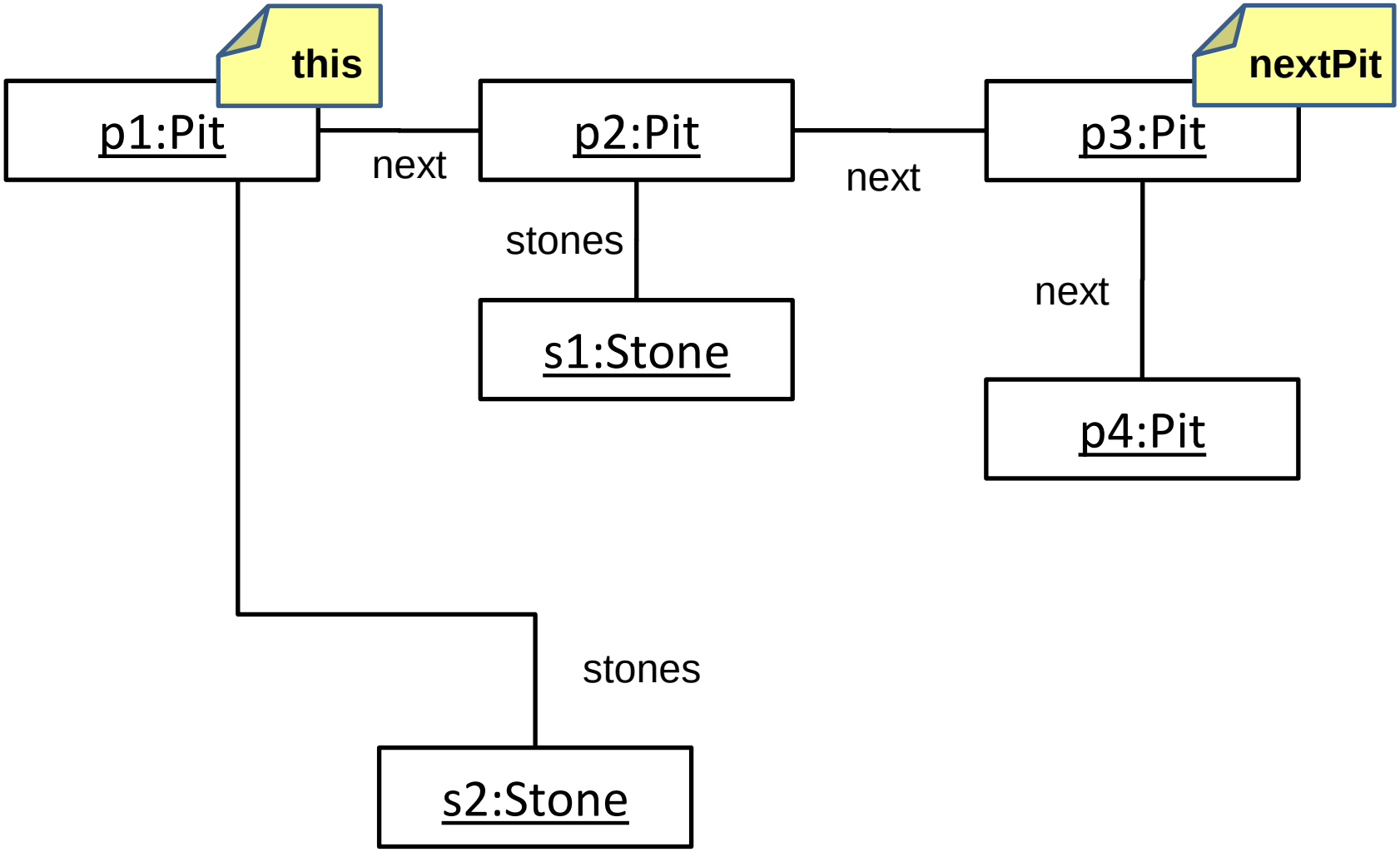
Methodenentwurf XVII - Zetteltest

- **Mögliche Implementierung**

```
1  public void moveStones ()
2  {
3        Pit nextPit = this.getNext();
4        for(Iterator<Stone> iter = getIteratorOfStones(); iter.hasNext();)
5        {
6              Stone stone = iter.next();
7              nextPit.addToStones(stone);
8              nextPit = nextPit.getNext();
9        }
10 }
```

- **Verifikation durch Zetteltest**

Methodenentwurf XVIII - Zetteltest



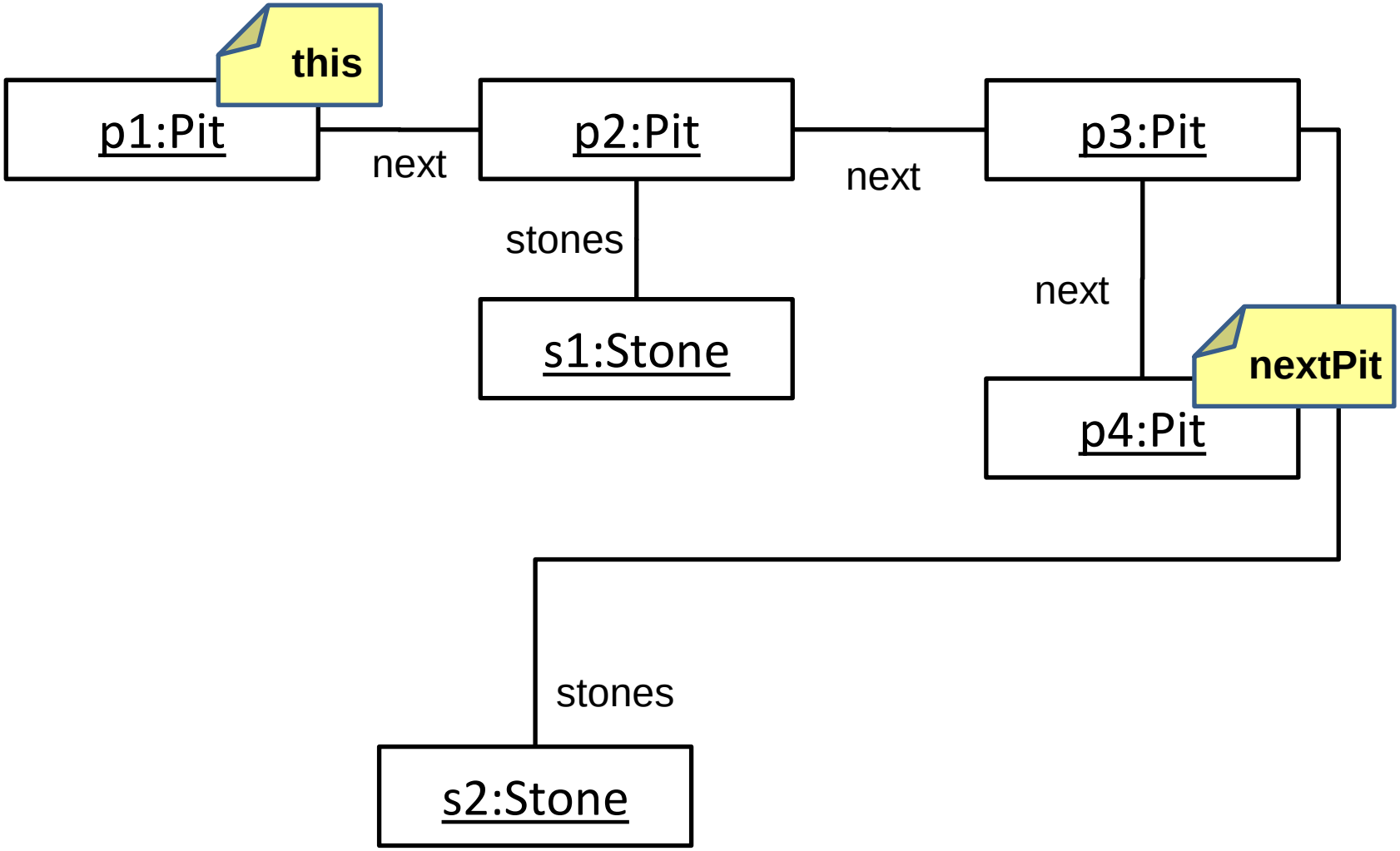
Methodenentwurf XIX - Zetteltest

- **Mögliche Implementierung**

```
1  public void moveStones ()
2  {
3          Pit nextPit = this.getNext();
4          for(Iterator<Stone> iter = getIteratorOfStones(); iter.hasNext();)
5          {
6                  Stone stone = iter.next();
7                  nextPit.addToStones(stone);
8                  nextPit = nextPit.getNext();
9          }
10 }
```

- **Verifikation durch Zetteltest**

Methodenentwurf XX - Zetteltest



Ende

Jetzt: Betreutes Arbeiten

Ansonsten: Schönes WE!