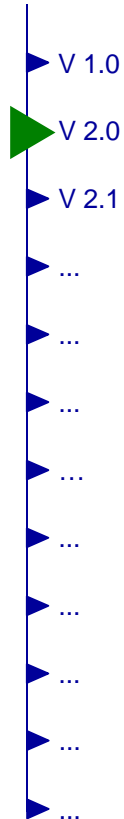




# Die nächsten beiden Folien...

---

Quasar-Beispiel:  
NIM



- ... illustrieren, wie das Hauptprogramm bis jetzt aufgebaut sein sollte
- als Sequenzdiagramm
  - als Java-Code

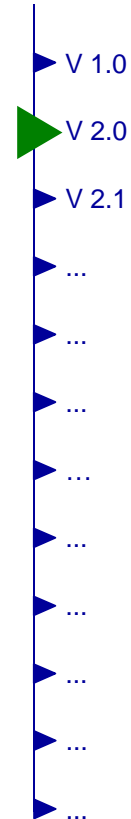




# Start-Programm (eine mögliche Variante)

```
public class Game implements UseCases, StateRetrieval {
```

Quasar-Beispiel:  
NIM

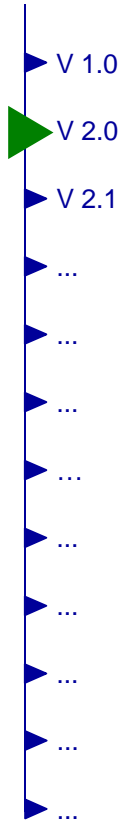


```
class GuiVsGuiSingleProcess {  
  
    public static void main( String[] args ) {  
  
        Game logic = new Game();  
  
        Player player1 = new Player();  
        player1.setUseCases(logic);  
        player1.setStateRetrieval(logic);  
  
        Player player2 = new Player();  
        player2.setUseCases(logic);  
        player2.setStateRetrieval(logic);  
  
        player1.start();  
        player2.start();  
        // die Spieler registrieren sich selbst beim Spiel  
    }  
}
```

```
public class Player implements ApplicationEvents {
```



Quasar-Beispiel:  
NIM



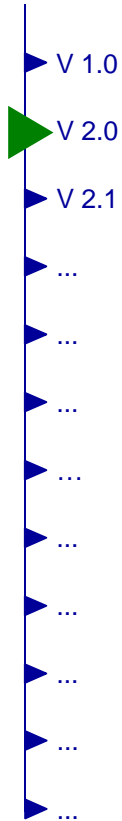
Stellen Sie sicher, dass Ihr  
Programm etwa dieser  
Struktur entspricht!

Erst dann...



## ... erweitern Sie Ihr Spiel: Kompositionsmanager, Spielregeln

Quasar-Beispiel:  
NIM



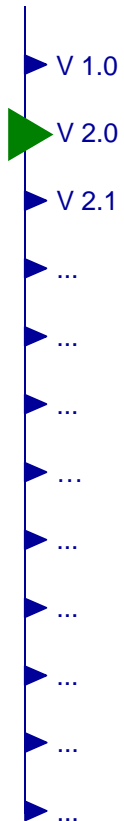
- Jedes Package darf beliebig viele *nicht-öffentliche* Klassen beinhalten
- Jedes Package enthält *genau eine öffentliche* Klasse: den **Kompositionsmanager** (die Klasse darf auch so heißen!)
- Der Kompositionsmanager
  - instanziiert alle nötigen nicht-öffentlichen Klassen und
  - verbindet sie mit anderen Komponenten
- Das Hauptprogramm
  - instanziiert alle Kompositionsmanager und
  - verbindet sie miteinander
- Über Komponentengrenzen hinweg sind nur die Kompositionsmanager sichtbar/zugreifbar, von den anderen Klassen ist nur bekannt, dass sie bestimmte Interfaces implementieren



## ... erweitern Sie Ihr Spiel: Fehlerfassade

---

Quasar-Beispiel:  
NIM



- Wenn Ihre Logik-Komponente Fehler bemerken oder verursachen kann:

Ergänzen Sie das Package um eine Fehlerfassade für das betroffene Interface / die betroffenen Interfaces

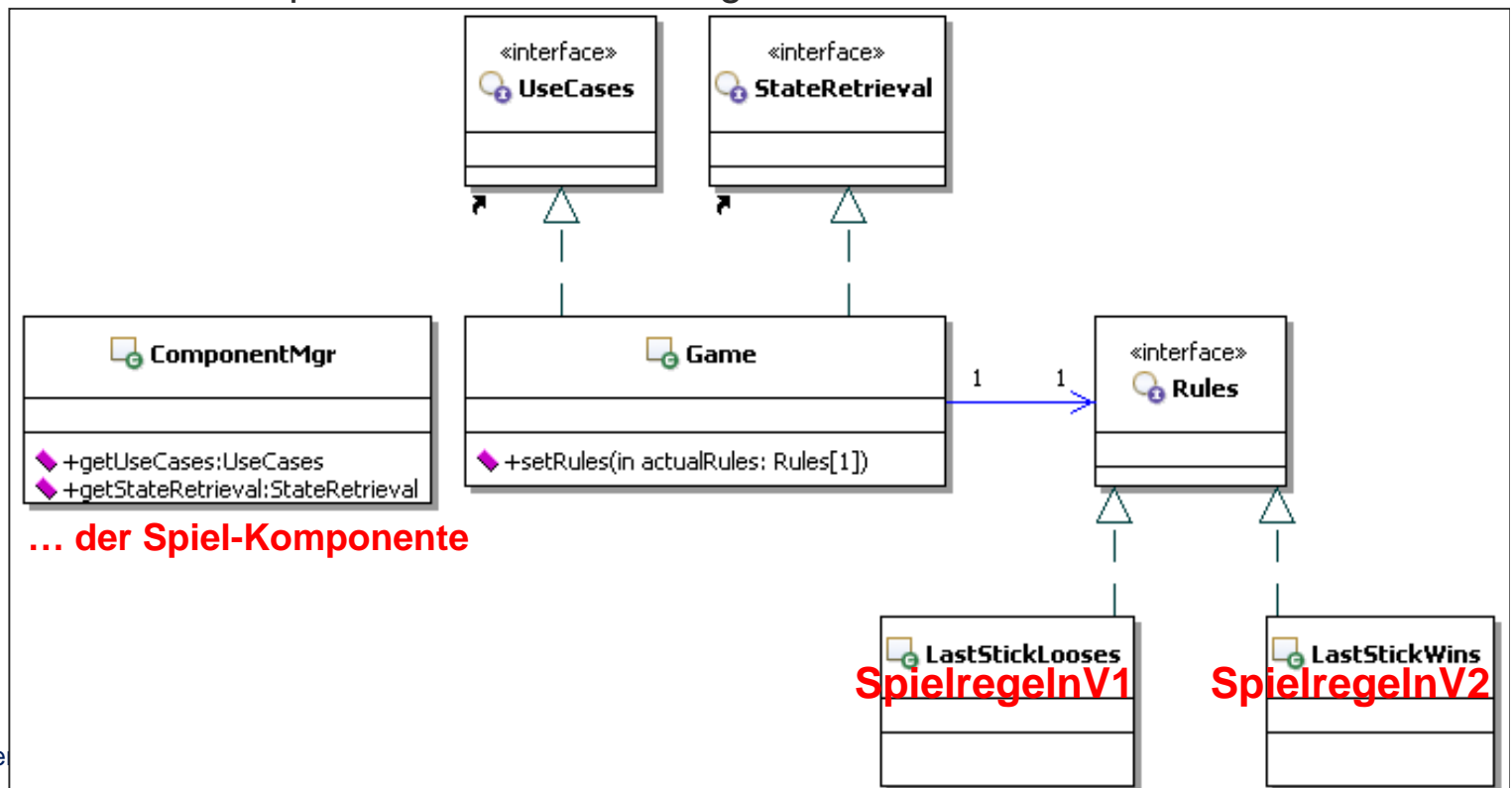
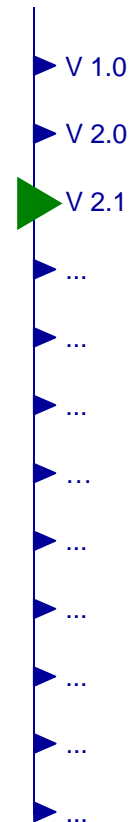


# Spielregel-Änderung

Die Spielregeln werden wie folgt geändert:

- Beim Programmstart wird bestimmt, ob der Spieler mit dem letzten Hölzchen verliert oder gewinnt
- Die Einstellung dafür liest der Kompositionsmanager der Spiel-Komponente aus einer Konfigurationsdatei (und baut seine Komponente dementsprechend); Entsprechendes gilt für den Kompositionsmanager der KI-Komponente
- Alle anderen Komponenten bleiben unangetastet!

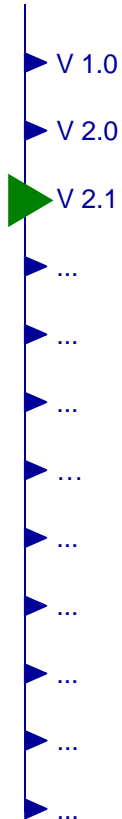
Quasar-Beispiel:  
NIM





# Erfolgskontrolle

Quasar-Beispiel:  
NIM



Wenn Sie alles richtig gemacht haben, können Sie zum Beispiel

- durch Ändern einer einzigen Zeile im Hauptprogramm eine der GUIs durch eine KI-Komponente ersetzen oder
- durch Ändern der Konfigurationsdatei für den Spiel-Kompositionsmanager die Spielregeln ändern, ohne dass diese Änderung nach außen dringt