

Software-Technik I (WS 2004/05)

7. Übungsblatt

1. Aufgabe

Spezifizieren Sie die Klassen `UseCaseAction` und `UseCaseActionExtension` aus dem dritten Übungsblatt durch Angabe der Klasseninvarianten sowie der Vor- und Nachbedingungen aller Methoden und Konstruktoren.

Hinweise:

- Beide Klassen haben die gleichen Methoden, aber diese haben teilweise unterschiedliche Vor- und Nachbedingungen. (Einige dieser Unterschiede weisen auf Entwurfsprobleme hin, mehr dazu in späteren Übungen.)
- Es soll nur die Schnittstelle der Klassen spezifiziert werden. Die Prädikate sollen deshalb *keine* Attribute enthalten.
- Sie dürfen beiden Klassen eine Methode `getName` hinzufügen, falls das die Spezifikation vereinfacht.

2. Aufgabe

Beweisen Sie mit Hilfe der in der ersten Aufgabe erstellten Spezifikation, dass das Programm

```
UseCaseAction action = new UseCaseAction(null, "Action");
SpecItem extension1 = action.addChild("Extension1");
System.out.println(action.getChildren()[0].toString());
action.removeChild(extension1);
System.out.println(action.toString());
SpecItem extension2 = new UseCaseActionExtension(action, "Extension2");
action.addChild(extension2);
System.out.println(action.getChildren()[0].getParent().toString());
```

folgendes ausgibt::

```
Extension1
Action (7 FP)
Action (10 FP)
```

Zeigen Sie dazu für jeden Konstruktor- und Methodenaufruf in der Reihenfolge ihrer Ausführung,

- das die Vorbedingung erfüllt ist,
- welche Werte sich für den Rückgabewert der Methode (falls vorhanden) und die Rückgabewerte der lesenden Zugriffsoperationen (`get*` und `toString`) des Objekts aus der Nachbedingung (einschließlich Klasseninvariante) schließen lassen.

Erlaubt die in der ersten Aufgabe erstellte Spezifikation einen solchen Beweis nicht, so ist sie unvollständig oder inkonsistent und sollte überarbeitet werden.

Letzter Abgabetermin: 25.01.2005