Vorstellung einer Lösung für die Aufgabe 4-2 von dem Studenten Julius P.

<u>Wichtige Entwurfsentscheidungen bzw. Fragestellungen:</u>

- *F1*: Für die Wiederverwendung der Klasse Container hat er eine Strategie abgeleitet und diese im Kommentar der Klasse dokumentiert. Wie bewerten Sie diese? Wie bewerten Sie zudem die Implementierung des Singleton-Patterns?

- *F2*: Bis auf die Klasse UserStory und ContainerException wurden *alle* Funktionalitäten in die Klasse Container implementiert. Die wichtigsten Zuständigkeiten (*responsibilities*) sind somit in einer Klasse, d.h. Container. Alle Klassen befinden sich in *einem* Package. Vorteile aus seiner Sicht: dies erhöht die Verständlichkeit, die Übersichtlichkeit und die Anpassbarkeit!

- *F3*: Das Abspeichern über die Methode store klappt noch nicht so recht, da wird eine Exception geworfen. Was könnte hier schiefgegangen sein? Warum wird das Programm dann direkt beendet? Auch die Ausgabe des Prompts ist etwas merkwürdig, die Begrüßung wird ständig wiederholt. Die Funktionsweise des Scanners hat Hr. P. noch nicht ganz verstanden. Hr. P. bittet sie um Rat!

- *F4*: Bei der Ausgabe der UserStory-Objekte (Methode Start-Ausgabe) weiß der Student nicht so recht weiter, da er nicht versteht, wie bzw. nach welchen Kriterien er sortieren soll. Kann man hier ggf. etwas wiederverwenden aus dem SDK? Zudem: Sind die Attribute von der Klasse UserStory vollständig?

- *F5*: Bei der Ausgabe soll auch mit dem Pattern „ FilterMap Reduce“ gearbeitet werden; er ist aber unsicher und hat keine brauchbare Lösung. Für die neue User-Story US1 hat er keine Idee, wie eine Filterung nach Projekten implementiert werden soll. Für die Studie der Musterlösung aus der Übung 2-2 (Klasse Container) hatte er wegen Karneval „noch keine Gelegenheit“. Bitte hierzu auch das Kapitel 9 (siehe LEA) beachten!

- *F6*: Wie könnte eine Zerlegung in mehrere Packages aussehen? Welches Muster könnte hier dienen?

- *F7*: Reduktion des Fehlerhandlings: die meisten Exceptions werden zur main-Methode delegiert (Anwendung Pattern Chain of Responsibility, siehe Methode store). Zudem ist das Fehlerhandling für die load-Operation sehr benutzerfreundlich, da viele Details dazu ausgegeben werden. (*Bearbeitung spätestens in der Vorlesung / Kapitel 6!*)