

Tutorium SEI - Übungen - Von Syntax bis Vererbung (und Interfaces)

Aufgabe 040 - Das Meetingprotokoll

Ein Protokoll enthält das Datum und den Zeitraum, an dem ein Meeting stattgefunden hat. Zudem wird der Raum sowie die Namen der Teilnehmer vermerkt. Schließlich werden die wesentlichen Diskussionspunkte, Beschlüsse sowie Aktionspunkte im Protokoll festgehalten.

Folgende Methoden sollen wir implementieren:

- Der Konstruktor, der ein Protokoll initialisiert.
- Die Methode mit der Signatur `public void add(Item item)` fügt dem Protokoll ein *Item* hinzu. Diskussionspunkte, Beschlüsse sowie Aktionspunkte werden mithilfe von zugehörigen Unterklassen repräsentiert, welche von *Item* erben. Jeder Diskussionspunkt, Beschluss und Aktionspunkt besteht aus einem Text, der den Inhalt des entsprechenden Punktes beschreibt.
- Die Methode mit der Signatur `public String toFullProtocolString()` gibt das vollständige Protokoll als String zurück. Bereitsstehende Klassen sollen ebenfalls über geeignete `toString()`-Methoden verfügen.

Aufgabenstellung

1. Skizzieren Sie die zu implementierenden Klassen und oder Interfaces mit Klassennamen, Attributen und Methoden. (Es bietet sich eine UML-ähnliche Darstellungsweise mit Vererbungshierarchie an.) Dabei sollen folgende Informationen erkennbar sein:
 - Name der Klasse
 - Name und Datentyp aller Attribute der Klasse (bei Konstanten ist der Wert mit anzugeben)
 - Name, Parameter und Rückgabewerttyp aller Methoden der Klasse
2. Schreiben Sie eine Klasse mit dem Namen *MeetingMinutes*, die Besprechungsprotokolle repräsentiert. Bitte darauf achten, dass verwendete Arrays bedarfsgerecht wachsen und keinen Speicherplatz verschwenden.
3. Schreiben Sie ein Testprogramm, welches für den angegebenen Testfall die angegebene Ausgabe produziert.

Testfall

```
// Folgender Beispielcode sei als Testcode gegeben:
String[] participants = {"Luigi Lustig", "Michaela Schneider", "Ulf
Wiel"};
MeetingMinutes meeting = new MeetingMinutes("10.10.2017", "10-12 Uhr",
"R123", participants);

meeting.add(new DiscussionItem("Veröffentlichung Buch"));
meeting.add(new DecisionItem("Dem Antrag wurde einstimmig
zugestimmt."));
```

```
meeting.add(new ActionItem("Bis zum nächsten Meeting muss Kapitel 9  
(Buch A) fertig gestellt sein.));  
meeting.add(new ActionItem("Bis zum nächsten Meeting muss Kapitel 3  
(Buch B) fertig gestellt sein.));  
System.out.println(meeting);
```

Dabei sollte folgende Ausgabe herauskommen (in etwa):

```
Meeting: 10.10.2017 (10–12 Uhr), R123  
Participants: Luigi Lustig, Michaela Schneider, Ulf Wiel  
-----  
Discussion: Veröffentlichung Buch  
Decision: Dem Antrag wurde einstimmig zugestimmt.  
Action: Bis zum nächsten Meeting muss Kapitel 9 (Buch A) fertig  
gestellt sein.  
Action: Bis zum nächsten Meeting muss Kapitel 3 (Buch B) fertig  
gestellt sein.
```

Hilfen

- Überlegen Sie genau, ob und an welcher Stelle die Verwendung eines Interfaces sinnvoll sein könnte.
- Beim Hinzufügen des Item-Array um ein Element vergrößern. Wie könnte man das sinnvoll umsetzen?
- Bei der toString()-Methode kann man einen Ausgabestring definieren, in welchen alle Inhalte bis zur Rückgabe reingeschrieben werden. Beim Durchgehen der Teilnehmerliste kann man bei allen Teilnehmern außer dem letzten jeweils ein Komma setzen.
- Zur Erinnerung: Zeilenumbrüche in Strings kann man mit der Zeichenkombination `\n` realisieren.