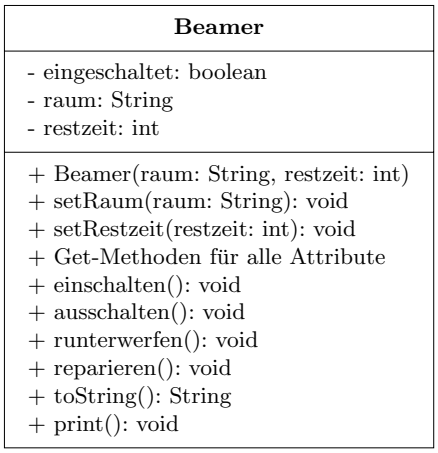
Übungsblatt zu Klassen, Testklassen und Berechnungen
Klasse: Beamer

SPENGERGASSE

Aufgabe 1 *Implementierung der Klasse* `Beamer`

Die Klasse `Beamer` modelliert einen Beamer, die Eigenschaften und Methoden sind durch folgendes UML-Diagramm gegeben:



Die `restzeit` gibt an, wie lange der Beamer noch verwendet werden kann (in Stunden). Restzeit gleich 0 bedeutet, dass der Beamer kaputt ist. Die Methoden `einschalten` und `ausschalten` ändern den Status des Beamers. Beim Einschalten wird die Restzeit um Eins reduziert. Die Methode `reparieren` erhöht die Restzeit des Beamers um 50 Stunden, und kann nur ausgeführt werden, wenn der Beamer kaputt ist, oder die Restlaufzeit weniger als 5 Stunden beträgt.

Bemerkung: Wenn ein Beamer mit einer Restzeit von einer Stunde eingeschaltet wird, so wird die Restzeit sofort auf 0 gesetzt. Der Beamer läuft aber weiter, und ist erst nach dem Ausschalten (oder Runterwerfen) funktionsunfähig, und damit kaputt.

Die `toString`, bzw. `print`-Methoden sollen folgende Formatierung verwenden:

```
Beamer in Raum C2.07 (ausgeschaltet), Restzeit: 1 h
Beamer in Raum C2.07 (eingeschaltet), Restzeit: 0 h, Ende der Lebensdauer erreicht
Beamer in Raum C2.07 (ausgeschaltet), Restzeit: 0 h, KAPUTT!
```

Bei diesen Ausgaben wurde ein Beamer mit Restlaufzeit von einer Stunde zunächst eingeschaltet, und dann `print` aufgerufen. Es folgt die Ausgabe mit "Ende der Lebensdauer", aber der Beamer ist noch eingeschaltet. Nach dem Ausschalten liefert `print` dann die Ausgabe, dass der Beamer "KAPUTT" ist.
