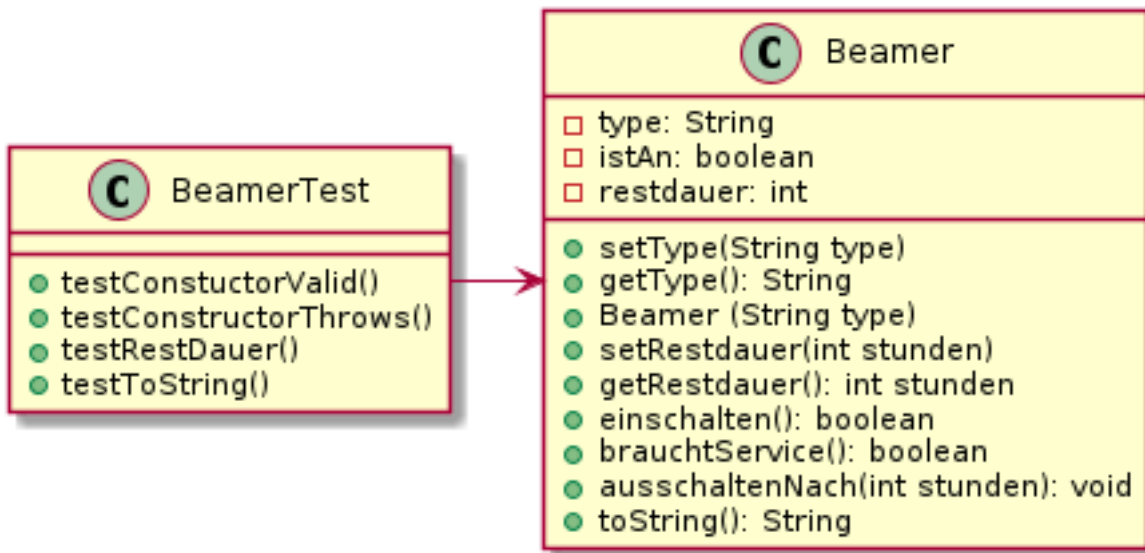


Praktische Leistungsfeststellung 3BKIF am 16. Oktober 2023

BEAMER

Zu implementieren sind die zwei Klassen `Beamer` und `BeamerTest` laut folgendem UML-Diagramm:



Implementierung Klasse Beamer

Konstruktor (4P)

Die Klasse `Beamer` modelliert einen Beamer. Der Konstruktor akzeptiert folgende 3 Typen: ["Standard", "LED", "Laser"]. Die Lebensdauer der Lichtquelle ist bei diesen 3 unterschiedlich. Standard: 2000 Betriebsstunden, LED: 60000 Betriebsstunden, Laser: 20000 Betriebsstunden. Bei ungültigen Parametern werfen Sie eine `IllegalArgumentException`, damit das Objekt nicht instanziiert wird. Ebenso setzen sie im Konstruktor entsprechend die verbleibenden Reststunden.

Methode `einschalten()` (3P)

... schaltet den Beamer ein. Der Rückgabewert (`boolean`) teilt dem Aufrufer mit, ob dies möglich ist, denn die Lebensdauer könnte schon verbraucht sein. Jeder Einschaltvorgang vermindert die Lebensdauer um 15 Stunden. Bei einer Restlaufzeit von weniger als 15 Stunden kann der Beamer somit **nicht** mehr eingeschaltet werden.

Methode `ausschaltenNach(int stunden)` (2P)

... schaltet den Beamer nach soundsovielen Betriebsstunden aus. In dieser Methode muss sich die `restdauer` um die entsprechende Stundenzahl verringern. Falls der Beamer eh nicht läuft, soll die Methode nichts tun.

Methode `brauchtService()` (3P)

... ermittelt, ob der Beamer eingeschalten werden kann.

Methode toString() (4P)

... liefert einen String, welcher den Beamer repräsentiert nach folgenden Beispielen:

- Standard-Beamer: Restlaufzeit: 1500 Stunden (eingeschaltet)
- LED-Beamer: Restlaufzeit: 40200 Stunden (ausgeschaltet)
- Laser-Beamer: Restlaufzeit: 5 Stunden (eingeschaltet) BENÖTIGT SERVICE

Aufgabe: Implementierung der Testklasse

Methode testConstructorValid() (4P)

Hier sollen Beamer-Objekte erzeugt werden und mittels der Methoden getType() sowie getRestdauer() geprüft werden, ob die entsprechenden Werte in den Objekten korrekt gesetzt wurden.

Methode testConstructorThrows() (4P)

Hier sollen ungültige Werte (null sowie ein unerlaubter Typ) an den Konstruktor übergeben werden, und der Test soll "asserten", dass eine IllegalArgumentException geworfen wurde.

Methode testRestDauer() (4P)

Legen Sie hier einen Beamer an und speichern ihn in einer Variablen.

- prüfen Sie, ob die Restdauer stimmt.
- Schalten Sie ein.
- prüfen Sie, ob die Restdauer stimmt.
- Schalten Sie nach ein paar Stunden aus.
- prüfen Sie, ob die Restdauer stimmt.
- Schalten Sie ein.
- prüfen Sie, ob die Restdauer stimmt.
- Schalten Sie nach ein paar Stunden aus.
- prüfen Sie, ob die Restdauer stimmt.

Methode testToString() (4P)

- Legen Sie einen Beamer an und bringen diesen in einen von Ihnen erwarteten Zustand.
- Überprüfen Sie, ob die toString() Methode den erwarteten String retourniert!

Essentielle Anmerkungen

Abgabe: Geben Sie das nicht gezippte Bluej Projekt auf Laufwerk Z: ab, indem Sie es dorthin kopieren. Danach können Sie den Labor-Rechner herunterfahren.

Compilierfähiger Code: Beachten Sie, dass nicht compilierbarer Code in der Abgabe zu einem negativen Ergebnis führt.

Richtige Einrückungen: Fehlerhaft bzw. nicht formatierter Code bewirkt Punkteabzug.

Punkteschlüssel

0-16: nicht genügend, 17-20: genügend, 21-24: befriedigend, 25-28: gut, 29-32: sehr gut

Gutes Gelingen!