

Aufgabe: Bandit und „OhmLogger“

Logger mit eigenem Formatter

1. Der Standard-Formatter des Standard-Loggers liefert leider eine 2-zeilige Ausgabe. Schreiben Sie deshalb eine eigene Logger-Klasse (z.B. OhmLogger), die einen eigenen Formatter (z.B. „OhmFormatter“) definiert. Dieser Formatter soll eine 1-zeilige, kompakte Textausgabe der Form
| Zeitstempel | Level | Klasse (= Quelle der Meldung) | Meldung|
realisieren. Die Klasse soll als Singleton realisiert werden. Über den Aufruf
`private static Logger lg = OhmLogger.getLogger();`
soll genau ein Loggerobjekt mit dem eigenen Formatter zurückgeliefert werden (Singleton).
2. Ermöglichen Sie die Einstellung des Logger-Levels und des Log-Verzeichnisses bzw. der Log-Datei mit Hilfe einer „Properties“-Datei.
3. Verwenden Sie Ihren „OhmLogger“ in allen späteren Programmen!

Spiel: Einarmiger Bandit

Realisieren Sie eine GUI-Anwendung „einarmiger Bandit“ á la „Las Vegas“ in MVC-Struktur. Dieser sollte die Struktur Start/Stopp auf Knopfdruck haben.



Dazu benötigt man unbedingt das Entwurfsmuster Subscriber / Publisher. Man implementiere eine neue Applikation („Model“), die aus drei nebenläufigen Zählern (siehe Würfel!

Wiederverwendbarkeit) besteht.

VORGABE: Zu Übungszwecken unbedingt mit 3 eigenständigen Zähler-Threads arbeiten und nicht die vermeintlich einfache Lösung „3 Integer-Werte im Model“ verwenden! Auch der „OhmLogger“ sollte - wie in Zukunft immer - Anwendung finden.

Hinweis: Die nebenläufige Zählerklasse des Würfels soll wiederverwendet werden. Es können somit 3 nebenläufige Zählerobjekte erzeugt werden, welche das Bandit Model aggregiert.

Später kann man die Zahlen durch Indizes eines Arrays mit Bildern ersetzen und hat dann einen klassischen „einarmigen Banditen“.

Als freiwillige Weiterarbeit kann man auch die 3 Zähler so implementieren, dass diese nacheinander langsam „auslaufen“ und anschließend stehen bleiben. Wie die Geldausgabe bei 3 gleichen Bildern geschieht überlasse ich Ihrer Kreativität ;-)

Achtung: **nicht** 3 Models verwenden - der Controller hat die Referenz auf **ein** Modellobjekt welches 3 Zähler besitzt (Aggregation)!

TIPP: Das Model ist Publisher (für den Adapter) und gleichzeitig Subscriber für die drei Zähler - muss also 3 Subscriptions verwalten! Der Adapter ist Subscriber vom Model.