Technology
Arts Sciences
TH Köln

Fakultät für Informations-, Medien- und Elektrotechnik
Institut für Nachrichtentechnik

Prof. Dr. Hans W. Nissen
Tel.: (0221) 8275 – 2489
Email: hans.nissen@th-koeln.de

Entwicklung komplexer Software-Systeme

Praktikumsblatt 1
Gruppe B
- Hausaufgaben -

Ziel: Implementierung von Entwurfsmustern

Abgabe der Lösungen: Bis zum 14.11., 08:00 Uhr morgens, im Master-Branch des Gitlab-Repositories P1EKS<IhreTeamnummer>. Abzugeben ist das vollständige IntelliJ-Projekt.
Hinweis: Ihr Gitlab-Repository zu diesem Praktikumsversuch enthält ein IntelliJ-Projekt, welches Sie als Grundlage für Ihre Implementierung verwenden sollen – Sie sollen also den Ordner P1EKS< IhreTeamnummer> als IntelliJ-Projekt öffnen.

Achtung: Seien Sie sorgfältig bei der Implementierung. Es ist sehr wichtig, dass Sie die Aufgabenstellung exakt und korrekt umsetzen. Verwenden Sie keine öffentlichen Attribute.

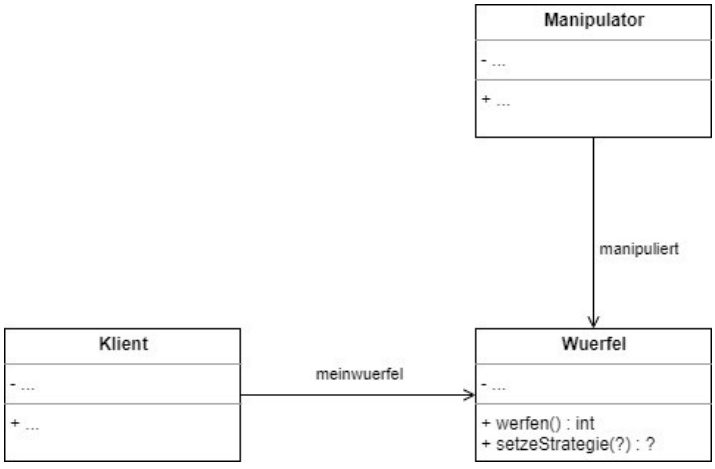
Aufgaben:

H1.1 Entwurfsmuster „Strategie“ implementieren

Implementierung bitte im Paket `aufgabe1`.

In einem Würfel-Spiel wird eine Klasse für einen sechsseitigen Würfel benötigt.

Ihr Auftraggeber fordert die Implementierung der Würfel-Klasse mit verschiedenen Würfelalgorithmen. Die Klasse soll `Wuerfel` heißen, die Operation zum Würfeln soll `werfen()` heißen. Die Klasse Klient ruft immer nur die Methode `werfen()` auf. Die Klasse `Manipulator` soll über die Methode `setzeStrategie()` die Möglichkeit erhalten, den Würfelalgorithmus während der Laufzeit beliebig zu ändern, ohne dass der Klient dieses bemerkt.



Es sollen die folgenden Algorithmen betrachtet werden:

- es wird eine Zufallszahl zwischen 1 und 6 gewürfelt

Technology
Arts Sciences
TH Köln

Fakultät für Informations-, Medien- und Elektrotechnik
Institut für Nachrichtentechnik

Prof. Dr. Hans W. Nissen
Tel.: (0221) 8275 – 2489
Email: hans.nissen@th-koeln.de

- der Würfel gibt beim Werfen immer einen fest voreingestellten Wert zurück (Bem.: den Wert können Sie sich selbst aussuchen)
- beim Werfen werden zyklisch Werte zurückgegeben (… 3 4 5 6 1 2 3 4 5 6 1 2 …)

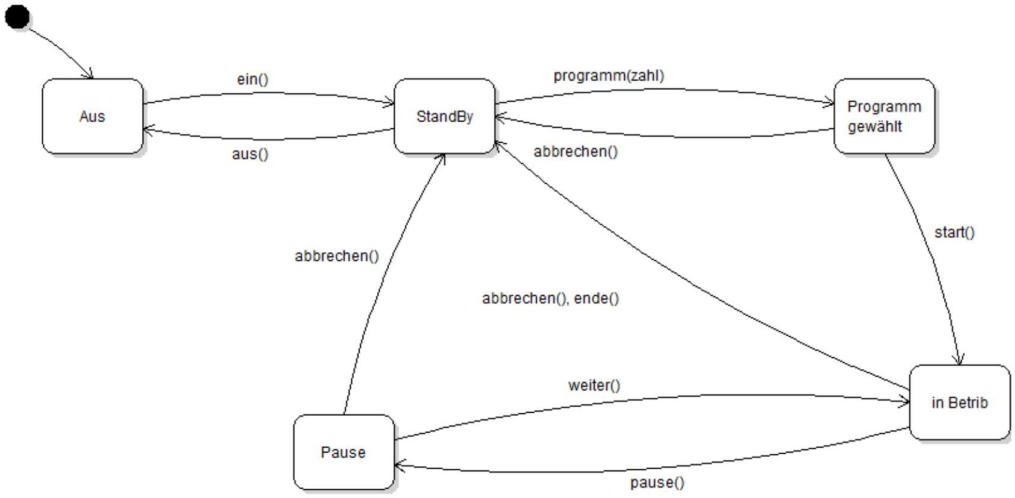
a) Implementieren die die Klassen `Wuerfel` und `Manipulator`. Verwenden Sie das aus der Übung bekannte Strategie-Muster.

b) Schreiben Sie hierfür ein kleines Testprogramm (= main-Methode) in einer neuen Klasse, in der ein `Klient` 10 Würfe mit jeder der drei Strategien ausführt und bei jedem Wurf der gewürfelte Wert ausgegeben wird.

H1.2 Entwurfsmuster „Zustand“ implementieren

Implementierung bitte im Paket `aufgabe2`.

Es soll eine Waschmaschine implementiert werden. Das Verhalten dieser Waschmaschine folgt dem folgenden Zustandsdiagramm:



Es sind folgende Methoden an der Klasse Waschmaschine vorhanden:

- `getZustand()`: der Aufruf dieser Methode liefert folgende Ausgabe auf der Console: „Waschmaschine ist im Zustand <Name des jeweiligen Zustands>“. Ein Zustandswechsel findet nicht statt.
- `ein()`
- `aus()`
- `ende()`
- `abbrechen()`
- `programm(zahl)`, wobei `zahl` ein Integer-Wert ist und das ausgewählte Programm angeben soll. In dieser Aufgabe ist die konkrete Zahl jedoch nicht von Bedeutung (muss aber beim Aufruf natürlich trotzdem übergeben werden).
- `start()`
- `pause()`

Technology
Arts Sciences
TH Köln

Fakultät für Informations-, Medien- und Elektrotechnik
Institut für Nachrichtentechnik

Prof. Dr. Hans W. Nissen
Tel.: (0221) 8275 – 2489
Email: hans.nissen@th-koeln.de

- `weiter()`

Der Aufruf einer dieser Methoden in einem gewissen Zustand kann also zu einem Zustandsübergang führen – wie im Zustandsdiagramm spezifiziert.
Falls der Aufruf einer Methode in einem Zustand entsprechend des Zustandsdiagramms nicht vorgesehen ist, soll bei Aufruf dieser Methode in diesem Zustand folgende Ausgabe auf der Console erfolgen: „Methode <Name der Methode> ist im Zustand <Name des Zustands> nicht möglich!“

a) Implementieren Sie die Klasse Waschmaschine mit dem obigen Zustandsdiagramm entsprechend des Zustandsmusters aus der Vorlesung.

b) Erstellen Sie eine Klasse mit einer main-Methode, in der Sie ein Objekt der Klasse Waschmaschine erzeugen. Testen Sie Ihre Implementierung durch die folgenden Aufrufe in der main-Methode:

- Aktueller Zustand der Maschine
- Waschmaschine ein
- Aktueller Zustand der Maschine
- Programm 3 wählen
- Aktueller Zustand der Maschine
- Waschmaschine starten
- Aktueller Zustand der Maschine
- Pause
- Aktueller Zustand der Maschine
- Weiter
- Aktueller Zustand der Maschine
- Ende
- Aktueller Zustand der Maschine

- Waschmaschine aus
- Aktueller Zustand der Maschine
- Waschmaschine ein
- Programm 4 wählen
- Abbrechen
- Aktueller Zustand der Maschine

- Programm 2 wählen
- Waschmaschine starten
- Programm 2 wählen
- Abbrechen

- Pause
- Pause
- Abbrechen
- Aktueller Zustand der Maschine

Technology
Arts Sciences
TH Köln

Fakultät für Informations-, Medien- und Elektrotechnik
Institut für Nachrichtentechnik

Prof. Dr. Hans W. Nissen
Tel.: (0221) 8275 – 2489
Email: hans.nissen@th-koeln.de

- Aus
- Aktueller Zustand der Maschine