

## Workshop "BlueJ" - OOP von Anfang an - mit java und BlueJ

A. Dietz, R. Punkenburg

---

### Ein einfacher Fahrscheinautomat

Im BlueJ-Projekt FSA1 sind bereits drei Klassen angelegt. Die Klasse **FSA** implementiert einen sehr simplen Fahrscheinautomaten. Die Klassen **Muenzpruefer** und **Drucker** sind funktionsfähig (und kommentiert), aber noch nicht mit dem Fahrscheinautomaten verbunden.

---

#### interaktives Erzeugen von Objekten interaktiver Methodenaufruf Verwenden des Objektinspektors

---

#### 1. Aufgabe

- Erzeugen Sie Objekte vom Typ **Drucker** und **Münzprüfer**. Rufen Sie Methoden für diese Objekte auf.
- Erzeugen Sie ein Objekt vom Typ **Fahrscheinautomat**. Rufen Sie Methoden für dieses Objekt auf. Öffnen Sie vor und nach verschiedenen Methodenaufrufen den Objektinspektor.

#### Konsolenausgaben

Hier werden Konsolenausgaben zur Rückmeldung an den Entwickler verwendet. Dadurch kann die Verarbeitungslogik gut nachvollzogen werden. Wenn die Fachklassen mit einer Benutzeroberfläche verbunden werden, werden alle Konsolenausgaben entfernt bzw. auskommentiert. Eine Ausgabe auf der Konsole wird erzeugt mit: `System.out.println(<String>)`

---

#### Arbeiten mit dem Editor Aufruf der Schnittstellen-Dokumentation

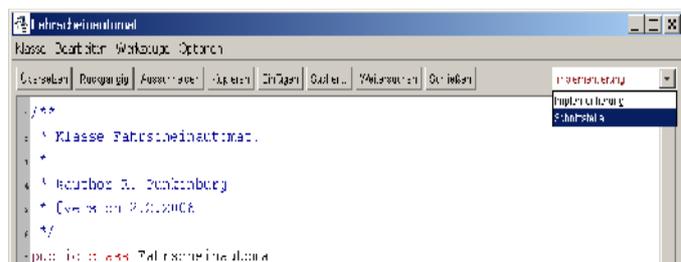
---

#### 2. Aufgabe

- Lesen Sie das README zum Projekt FSA1
- Öffnen Sie für jede der drei Klassen den Quelltexteditor. Schalten Sie das Editorfenster auf "Schnittstelle" um.
- Die Kommentierung der Methoden `wahleFahrschein(...)` ist unvollständig. Was sollte hinzugefügt werden?
- Welche Methode in diesem Projekt ist eine Methode mit Rückgabewert? Von welchem Typ ist dieser Rückgabewert?

#### Dokumentationen

Java verfügt über ein Werkzeug zur automatisierten Generierung von Dokumentationen aus einem entsprechend formatierten Quelltext (das "Hilfsprogramm" javadoc). Alle Bibliotheksklassen sind in diesem Standardformat dokumentiert.



BlueJ bietet die Möglichkeit, vom Quelltexteditor direkt zur Ansicht der Dokumentation umzuschalten. Im Editorfenster können Sie oben rechts zwischen "Implementierung" und "Schnittstelle" umschalten.

Eine kurze Einführung zu den Formatierungsregeln für die Verwendung von javadoc finden Sie in dem Infoblatt [Klassendokumentation.pdf](#).

---

### Visualisierung der Klassenbeziehungen OOP von Anfang an Objektinteraktion

---

Der Fahrscheinautomat soll nun seine Fahrscheine auf dem zur Verfügung stehenden Drucker ausgeben.

#### 3. Aufgabe:

Verwenden Sie die Klasse **Drucker** zum "Ausdrucken" der Fahrscheine.

#### Anleitung

Die Klasse *Fahrscheinautomat* benötigt *Attribut* vom Typ *Drucker*. Im *Konstruktor* muss dieses *Objekt* instanziiert werden. Dann kann dieser "Drucker" verwendet werden, d.h. seine Methoden können aufgerufen werden.

---

### Visualisierung der Klassenbeziehungen OOP von Anfang an Programmieraufgaben, Algorithmen

---

Sie haben sicher schon festgestellt, dass die bisherige Version des Fahrscheinautomaten den Münzprüfer noch gar nicht verwendet (also noch beliebige Münzwerte verbucht).

#### 4. Aufgabe:

Verbessern Sie den Fahrscheinautomaten so, dass nur noch gültige Münzen angenommen werden. Verwenden Sie dazu die Klasse **Muenzpruefer**.

#### Anleitung

Die Klasse *Fahrscheinautomat* benötigt *Attribut* vom Typ *Muenzpruefer*. Im *Konstruktor* muss dieses *Objekt* instanziiert werden. Dann kann vor dem Verbuchen einer Münze ihre Gültigkeit geprüft werden.

---

### Weiterführende Aufgaben Eigene Methoden schreiben Eigene Klassen schreiben

---

#### 5. Aufgabe:

Implementieren Sie eine Methode, durch die ein laufender Vorgang abgebrochen werden kann.

#### 6. Aufgabe:

Die Druckeinrichtung eines realen Fahrscheinautomaten verfügt über eine begrenzte Anzahl von unbedruckten Fahrscheinen. Falls diese verbraucht sind, schaltet der Automat auf "Zur Zeit außer Betrieb". Verändern Sie die Klasse **Drucker** so, dass die Anzahl der noch zur Verfügung stehenden Fahrscheine protokolliert wird. Verändern Sie die Klasse **FSA** so, dass eine Bedienung nur möglich ist, falls noch unbedruckte Fahrscheine zur Verfügung stehen.