

Klassendefinitionen I

Lernziele

- Sie haben auf Ihrem Laptop eine funktionierende BlueJ-Umgebung eingerichtet und können diese verwenden.
- Sie können einfache Klassen basierend auf einer Spezifikation und unter Berücksichtigung der Clean Code-Regeln programmieren und diese in BlueJ testen.

Aufgabe 1

Die Teilaufgabe a) können Sie überspringen, wenn Sie schon gewohnt sind mit einer Entwicklungsumgebung zu arbeiten und den Unterschied von Klassen und Objekten gut kennen. Sie können dann auch bei den übrigen Aufgaben mit Ihrer Entwicklungsumgebung arbeiten, statt BlueJ zu verwenden.

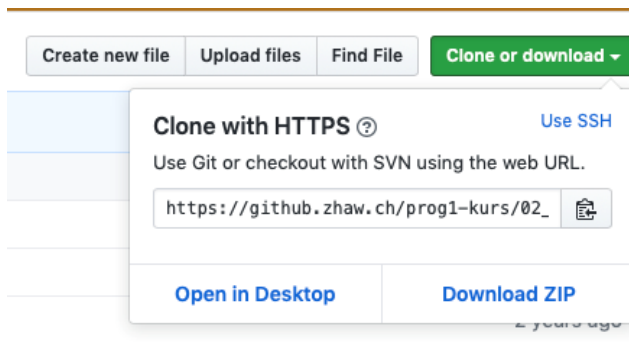
- Falls noch nicht gemacht, installieren Sie eine funktionsfähige BlueJ-Umgebung auf Ihrem Laptop. Benutzen Sie dazu Ihren Package-Manager oder gehen Sie dazu auf <http://www.bluej.org> und dort auf den Download-Bereich. Mit dem BlueJ wird auch Java installiert. Um die aktuell im Terminal verwendete Java-Version zu bestimmen, führen Sie `java -version` in einem Terminal aus.
- Kopieren Sie ebenfalls das Verzeichnis Projekte von der Begleit-CD des Buchs auf Ihren Laptop. Damit erhalten Sie alle Beispiele, die im Buch verwendet werden.
- Prüfen Sie dann, ob alles korrekt funktioniert. Öffnen Sie dazu in BlueJ den naivenFal Ticke-automaten aus Kapitel 2 und interagieren Sie mit diesem, um mit der Verwendung von BlueJ vertraut zu werden.

Aufgabe 2

Studieren Sie das Dokument *14_Anleitung-Arbeiten-mit-Git.pdf* bei den Praktikumsunterlagen auf Moodle. Überspringen Sie Abschnitte, die die Eclipse Entwicklungsumgebung betreffen. Studieren Sie diese Abschnitte, sobald wir auf Eclipse umsteigen.

Forken Sie das Projekt: https://github.zhaw.ch/prog1-kurs/02_Praktikum-1_Buch. Holen Sie die eigene Projektkopie auf Ihren Computer (`git clone`) und öffnen Sie das Projekt in BlueJ..

Hinweis: Falls Sie Schwierigkeiten haben, können Sie mit "Download Zip" eine Kopie des Codes herunterladen und damit arbeiten.



Kompilieren Sie die Klasse, was auf Grund der Fehler nicht funktionieren sollte. Gehen Sie nun wie folgt vor:

- Korrigieren Sie jeweils den gemeldeten Fehler und kompilieren Sie die Klasse erneut, bis sämtliche Fehler korrigiert wurden. Notieren Sie jeden Fehler.

Line 19: `titel = Buchtitel` // hier fehlt das Semicolon
==> `titel = Buchtitel;`
Line 18: `public Buch (String Buchtitel){` // hier fehlt das "int BuchISBN"
==> `public Buch (String Buchtitel, int isbn){`
Line 35: `public void veraendereBestand(int veraenderung) {` // void zu int
==> `public int veraendereBestand(int veraenderung) {`
Line 43: `public String output() {` // void nicht String
==> `public void output() {`

- Sobald die Klasse kompiliert werden kann: Gibt es weitere Fehler, die der Compiler nicht findet? Wenn ja, geben Sie die Fehler an mit jeweils einer Begründung, was falsch ist, und warum der Compiler den Fehler nicht findet. Korrigieren Sie auch diese Fehler im Code.

~~int b besser bestand // klarer~~

~~this. vor Instanzvariabek // zeigt gleich Instanzvariabel~~

~~beim Konstruktor wegen this. kann man gleichen Parameternamen nehmen~~

~~Instanzvariablen sind private / könnte sonst von aus der Klasse darauf
zugegriffen werden.~~

- Hat Ihnen der Compiler Verstösse gegen die Clean Code-Regeln gemeldet? Wieso? Passen Sie den Code gemäss Ihren Erkenntnissen an, damit die Clean Code-Regeln erfüllt sind.

Erzeugen Sie anschliessend ein Buch-Objekt und interagieren Sie mit diesem, um sicherzustellen, dass die Klasse korrekt implementiert ist. Verwenden Sie den Objektinspektor um zu prüfen, wie sich die Datenfelder verändern und experimentieren Sie auch mit der Direkteingabe (Menü *Ansicht* → *Direkteingabe anzeigen*), um in Java“ Objekte zu erzeugen und mit ihnen zu interagieren.

Aufgabe 3

Forken Sie das Projekt: https://github.zhaw.ch/prog1-kurs/02_Praktikum-1_Konto

Nutzen Sie BlueJ um die eigene Projektkopie auf Ihren Computer zu holen und zu bearbeiten. Das Projekt enthält noch keine Klassen. Diese schreiben Sie in dieser Aufgabe.

Implementieren Sie dann eine Klasse, welche in einer Bankapplikation für die Modellierung eines Bankkontos dienen soll. Nachfolgend ist diese Klasse spezifiziert:

- Ein Konto hat einen Kontoinhaber, einen Kontostand (in Franken, ganzzahlig) und einen Zinssatz (in %, ganzzahlig).
- Ein Konto kann auf zwei Arten erzeugt werden:
 - Mit Angabe des Kontoinhabers. Der Kontostand soll initial 0 und der Zinssatz 2 sein.
 - Mit Angabe des Kontoinhabers und des Zinssatzes. der Kontostand soll initial 0 sein.

Hinweis: Eine Klasse kann mehrere Konstruktoren haben, wenn sich diese durch die Parameterlisten eindeutig unterscheiden lassen.

- Der Zinssatz kann durch geeignete Methoden abgefragt und neu gesetzt werden.
- Es gibt je eine Methode, um Geld einzuzahlen und um einen Betrag abzuheben. Sie müssen nicht prüfen, ob die verwendeten Beträge sinnvoll sind oder ob der Kontostand negativ wird.
- Es gibt eine Methode, welche den effektiven Jahreszins basierend auf dem aktuellen Kontostand zurückgibt. Ist der Kontostand z.B. CHF 4000 und der Zinssatz ist 3%, dann wird 120 zurückgegeben. Kümmern Sie sich nicht um allfällige Präzisionsverluste und rechnen Sie „ganzzahlig“.
- Der Zustand eines Kontos kann auf der Kommandozeile ausgegeben werden. Die Formatierung soll dabei genau wie folgt aussehen:

```
Informationen zum Konto:  
Kontoinhaber: Max Meier  
Kontostand: CHF 4000  
Zinssatz: 3%
```

Testen Sie die Klasse nach der Implementierung, um sicherzustellen, dass sie korrekt funktioniert und alle Anforderungen erfüllt.

Beachten Sie bei dieser Aufgabe insbesondere folgende Punkte:

- Die Klasse selbst, die Konstruktoren und jede public Methode soll einen prägnanten Javadoc-Kommentar (`/** ... */`) erhalten. Weitere Kommentare sollten nicht nötig sein.
- Achten Sie auf die Organisation in der Klasse: Erst die Datenfelder, dann Konstruktoren, dann Methoden.
- Beachten sie die Clean Code Regeln. Wählen Sie entsprechend sinnvolle Namen für die Klasse, die Datenfelder, die Methoden und die Parameter. Relevant sind diejenigen Regeln im Dokument «PROG1 Clean Code Regeln» (siehe Moodle), die zu den bereits behandelten Kapiteln aus dem BlueJ-Buch gehören. Die Kapitel sind im Inhaltsverzeichnis aufgeführt.