

SE1, Aufgabenblatt 1

Softwareentwicklung I – Wintersemester 2012/13

Erster Kontakt mit Klassen und Exemplaren

MIN-CommSy-URL: <https://www.mincommsy.uni-hamburg.de/>

Projektraum: SE-1 CommSy WiSe 12/13

Ausgabedatum: 18. Oktober 2012

Kernbegriffe

Objekte (engl.: objects) sind *Exemplare* (engl.: instances) von *Klassen*. Eine Klasse (engl.: class) definiert für ihre Exemplare, welche *Methoden* (engl.: methods) an ihnen aufrufbar sind. Wir erzeugen Exemplare durch *Konstruktoraufrufe* (engl.: constructor calls) und benutzen sie, indem wir ihre Methoden aufrufen. Die Methoden, die an den Exemplaren einer Klasse aufrufbar sind, bilden die *Schnittstelle* (engl.: interface) der Klasse.

Methoden können *Parameter* (engl.: parameters) haben, mit denen zusätzliche Informationen für eine Aufgabe angegeben werden. Ein Parameter hat einen Namen und einen *Typ* (engl.: type) der festlegt, von welcher Art der Parameter ist. Beispiele für Typen sind die Ganzen Zahlen (in Java u.a. `int`), Zeichenketten (in Java `String`) und Wahrheitswerte (in Java `boolean`).

Wie eine Methode aufzurufen ist, wird über ihre *Signatur* (engl.: signature) beschrieben. In Java legt eine Signatur für eine Methode nur ihren *Namen* und ihre *Parametertypen* fest. Die Signatur einer Methode

```
boolean istAuszahlenMoeglich(int euro, int cent)
```

lautet demnach:

```
istAuszahlenMoeglich(int, int)
```

Der Ergebnistyp und die Parameternamen spielen also für die Signatur keine Rolle.

Ein Exemplar hat einen internen Zustand, welcher sich in den Belegungen seiner *Felder* (engl.: fields) widerspiegelt und von außen nicht unmittelbar einsehbar ist. Er kann aber über Methoden abgefragt und verändert werden. (Eine Besonderheit von BlueJ: der Zustand interaktiv erzeugter Exemplare lässt sich auch direkt anzeigen.) Welche Felder ein Exemplar hat, legt seine Klasse fest. Jedes Exemplar einer Klasse hat jeweils seinen eigenen, von anderen Exemplaren unabhängigen (konkreten) Zustand.

Wir empfehlen euch das Pflegen eines Glossars. Auf den Aufgabenblättern und in der Vorlesung werden viele Begriffe genannt und erklärt. Ihr müsst diese Termini kennen und verstehen, damit ihr die nächsten Übungsaufgaben bearbeiten könnt. Ein Glossar ist dafür ein geeignetes Hilfsmittel.

Lernziele

BlueJ starten können; eigenes Verzeichnis zum Ablegen von Projekten kennen; Objekte interaktiv erzeugen können; Objekte interaktiv manipulieren können; einfache Parameter interaktiv übergeben können; Verhältnis zwischen Klasse und Objekt erklären können; Verhältnis zwischen Methoden und Schnittstelle erklären können.

Vorbereitung

Erstellt in eurem *Home*-Verzeichnis (unter Windows als Laufwerk „Z:“ eingebunden) einen neuen Ordner SE1. Auf diesen könnt ihr von beliebigen Rechnern im Informatikum unter eurer RZ-Kennung zugreifen.

Öffnet einen Internet-Browser und besucht den SE1 Projektraum im MinCommSy. Die URL findet ihr im Kopf des Aufgabenblattes. Ladet die ZIP-Datei *Blatt01_Figuren.zip* in das eben angelegte Verzeichnis SE1 herunter. Ihr findet die Datei im CommSy unter „Materialien“ im Eintrag „Aufgabenblatt 01“.

Startet die Entwicklungsumgebung BlueJ. Unter Windows findet ihr den Programmeintrag im Start-Menü, unter Linux tippt ihr in einem Terminal „bluej“ ein und drückt die Enter-Taste. Öffnet nun die ZIP-Datei, indem ihr den Menü-Eintrag „Projekt“ -> „Projekt öffnen“ auswählt. Ihr solltet nun ein Diagramm mit den vier Klassen `Leinwand`, `Kreis`, `Dreieck` und `Quadrat` sehen.

Aufgabe 1.1 Exemplare erzeugen und Methoden aufrufen

- 1.1.1 Klassen können als *Baupläne* für die Erstellung von Exemplaren bezeichnet werden. Erzeugt interaktiv ein Exemplar von `Kreis`, indem ihr im Kontextmenü der Klasse `Kreis`, zu erreichen über die rechte Maustaste, den Eintrag `new Kreis()` auswählt (und den vorgegebenen Namen erst einmal akzeptiert).
Hinweis: Falls dies bei euch nicht klappt, müsst ihr wahrscheinlich einmal auf den Button „Übersetzen“ klicken. Danach sollte es funktionieren.
- 1.1.2 Unten auf der Objektleiste erscheint das gerade erzeugte Exemplar. Ruft nun die Methode `sichtbarMachen` für diesen Kreis auf, indem ihr im Kontextmenü den entsprechenden Eintrag auswählt. Wenn ihr das tut, wird die Methode `sichtbarMachen`, die in der Klasse definiert ist, von diesem Kreisexemplar ausgeführt. Es sollte ein Fenster erscheinen (eventuell müsst ihr es in den Vordergrund holen), in dem der Kreis sichtbar ist.
- 1.1.3 Führt weitere Methoden an diesem Kreis aus: verschiebt ihn z.B. oder ändert seine Farbe. Auf diese Weise ändert ihr den Zustand des Kreises. Es gibt Methoden, die aufgerufen werden können, ohne weitere Informationen zu benötigen. Andere Methoden brauchen zusätzliche Angaben als Parameter. Achtung: Zeichenketten (Typ `String`) müssen von doppelten Anführungszeichen umschlossen sein. Beispiel: `"Hallo"`
- 1.1.4 Erzeugt mehrere grafische Objekte und zeichnet auf diese Weise ein Bild. Erklärt den Betreuern mit den Begriffen aus dem Einleitungstext, wie ihr dabei vorgegangen seid.
- 1.1.5 Vergleicht die Methoden der verschiedenen Figurentypen. Was sind die Gemeinsamkeiten, was die Unterschiede?
- 1.1.6 Was gehört neben der Farbe noch zum Kreiszustand? Verwendet die *Inspect*-Funktion von BlueJ (Doppelklick auf das Exemplar), um die Felder des Kreises anzuzeigen.

Aufgabe 1.2 Was ist eine Schnittstelle?

Vermutlich habt ihr zuhause einen *Fernseher* (auch wenn ihr keinen habt, wisst ihr vermutlich, was ein Fernseher ist). Dieser Fernseher ist ein Exemplar eines Bauplanes, über den der Hersteller des Gerätes verfügt. Wenn wir diesen Fernseher benutzen, dann verhält er sich wie ein Objekt in der Objektorientierung: Er hat einen Zustand, und wir können diesen Zustand manipulieren. Dazu hat ein Fernseher auch eine Schnittstelle zum Benutzer, über die er seine Leistungen anbietet.

- 1.2.1 Wenn ihr die Schnittstelle eures Fernsehers objektorientiert beschreiben solltet, welche Methoden hätte er dann? Haben diese Methoden Parameter? Wenn ja, welche? Wenn nein, warum nicht? Haltet eure Überlegungen **schriftlich fest**. Ihr könnt hierfür ein Blatt Papier oder einen Texteditor verwenden.
- 1.2.2 Wisst ihr, wie ein Fernseher im Detail funktioniert? Wie wird die Leistung erbracht, die sich uns durch die bequeme Schnittstelle des Fernsehers anbietet? Welcher Teil dieser Leistung des Fernsehers ist für uns sinnlich erfahrbar/sichtbar, welcher nicht? Was geschieht intern, „hinter den Kulissen“? Was gehört zum internen Zustand eines Fernsehers? Ist die Umsetzung dieses internen Zustands für uns als Benutzer des Fernsehers relevant? Haltet auch hier eure Überlegungen **schriftlich fest**.
- 1.2.3 Versucht, eine möglichst knappe Definition des Begriffs Schnittstelle anhand eurer vorigen Überlegungen **schriftlich** zu formulieren.

Aufgabe 1.3 Was sind Klassen und Exemplare?

- 1.3.1 Findet mindestens drei Beispiele in eurer direkten Umgebung, auf die die Begriffe *Klasse* und *Exemplar* der objektorientierten Programmierung gut passen. **Schreibt sie auf**, damit ihr sie mit euren Betreuern diskutieren könnt.
- 1.3.2 Findet Beispiele, bei denen die Begriffe nicht gut passen oder nicht gut anwendbar sind. Das dürfte schon schwieriger sein. **Schreibt auch diese auf**.
- 1.3.3 **Formuliert schriftlich** mindestens drei Sätze, die die Beziehung der Begriffe *Klasse*, *Exemplar* und *Schnittstelle* im objektorientierten Sinne beschreiben. Sie können auf eure Beispiele bezogen sein oder einfach nur die genannten Begriffe korrekt miteinander in Beziehung setzen.
- 1.3.4 Worin unterscheiden sich Exemplare? Stellt euch beispielsweise eine leere und zwei volle Kekspackungen derselben Marke vor. **Beschreibt schriftlich** und begründet.