

Anmerkungen

- **Abgabe:** Quellcode zu 3-1 und 3-2 auf Papier, Lösung von 3-3 auf Papier, ausserdem Quellcode zu 3-1 und 3-2 über ILIAS.
- Jede Quellcode-Datei enthält **Name(n)** und **Matrikelnummer(n)**.

Aufgabe 3-1

Sie sollen ein Spiel programmieren, bei dem sich der Computer zufällig ein Feld eines Schachbretts "merkt" und der Spieler dieses Feld in maximal 3 Versuchen erraten muss. Zu Beginn soll sich das Spiel wie folgt präsentieren:

```

      ( )
    <~~~~>
      \__/_
    (____)
      |  |
      |__|
    /_____\
  (_____)
(_____)

*****
*GUESS CHESS*
*****

      ___/""
    |___ 0  }
      /      }
      /      }
      \_____/
      /_____\
    (_____)

- - - - -
8|_|_|_|_|_|_|_|_|
7|_|_|_|_|_|_|_|_|
6|_|_|_|_|_|_|_|_|
5|_|_|_|_|_|_|_|_|
4|_|_|_|_|_|_|_|_|
3|_|_|_|_|_|_|_|_|
2|_|_|_|_|_|_|_|_|
1|_|_|_|_|_|_|_|_|
  a b c d e f g h

You have 3 attempts left. Guess field (e.g. b3): _

```

Der Computer soll nach jeder falschen Eingabe (hier 'g7') Hinweise geben ("nach rechts", "nach rechts und runter" usw.):

```

g7 was wrong. Hints: GO LEFT! GO DOWN!

- - - - -
8|_|_|_|_|_|_|_|_|
7|_|_|_|_|_|_|X|_|
6|_|_|_|_|_|_|_|_|
5|_|_|_|_|_|_|_|_|
4|_|_|_|_|_|_|_|_|
3|_|_|_|_|_|_|_|_|
2|_|_|_|_|_|_|_|_|
1|_|_|_|_|_|_|_|_|
  a b c d e f g h

You have 2 attempts left. Guess field (e.g. b3): _

```

Nach jedem Rateversuch soll das Brett angezeigt und das geratene Feld mit einem 'X' markiert werden. Wenn der Spieler richtig geraten hat, gewinnt er:

You win!

Falls der Spieler das richtige Feld nach 3 Versuchen nicht erraten konnte, gewinnt der Computer:

You lose! Secret field: g5.

Auf ILIAS (Übungsserien → Übungsserie 3) finden Sie eine Datei `GuessChess.java`, die Sie ergänzen sollen. Methoden zur Darstellung des Schachbretts sowie zur Konvertierung der Buchstaben a-h in Zahlen 1-8 und umgekehrt sind gegeben. Ebenso eine `main`-Methode, die das Spiel startet.

Hinweis: Ungültige Eingaben (z.B. "z9" oder "bla") müssen Sie nicht abfangen.

Fakultativ: Sie können das Spiel erweitern indem Sie mehrere Durchgänge erlauben und beim Beenden des Spiels das Ergebnis (gewonnene/verlorene Spiele) anzeigen.

Aufgabe 3-2

Erweitern Sie die Klasse `Book` aus der Übungsserie 2 um eine Instanzvariable `private int price` mitsamt Get- und Set-Methoden. Verwenden Sie dazu die Datei `Book.java` der Musterlösung (siehe auch Anhang von Übungsserie 3 auf ILIAS). Stellen Sie sicher, dass der Preis eines Buches immer grösser oder gleich 0 ist. Tipp: Programmieren Sie die `setPrice`-Methode und Konstruktor(en) so, dass negative Preise stets durch 0 ersetzt werden.

Schreiben Sie anschliessend eine Klasse `Order` für Buchbestellungen. Ein `Order`-Objekt soll aus einer `id`, einem Kundennamen `customerName`, einer Kundenadresse `customerAddress` und maximal 5 `Book`-Objekten bestehen. Zudem soll die Klasse `Order` die Methoden `toString()`, `addBook(...)` und `getTotalPrice()` enthalten.

Schreiben Sie zudem einen Konstruktor `Order()`, der die Instanzvariable `id` automatisch so initialisiert, dass das erste `Order`-Objekt die `id` 1, das zweite die `id` 2, das dritte die `id` 3 usw. erhält. Tipp: verwenden Sie eine `static`-Variable.

Verwenden Sie anschliessend die gegebene Klasse `Test` (ILIAS: Übungsserien → Übungsserie 3) um Ihre Klasse `Order` zu testen. Die Ausgabe von `Test` soll **exakt** so aussehen:

```
$ java Test
Order id: 1, Customer: Sophie Muster, Mittelstrasse 10, 3011 Bern
1, Homo Faber, Max Frisch, 01.01.1957, 0 CHF
2, Harry Potter, J.K. Rowling, 25.07.2000, 45 CHF
3, Krieg und Frieden, Leo Tolstoi, 24.01.1867, 29 CHF
4, Freedom, Jonathan Franzen, 08.06.2010, 39 CHF
4, Freedom, Jonathan Franzen, 08.06.2010, 39 CHF
Total price: 152 CHF

Order id: 2, Customer: Woody Allen, 5th Avenue 7, 10001 New York
5, Goedel, Escher, Bach, Douglas Hofstadter, 05.11.1979, 42 CHF
Total price: 42 CHF
```

Hinweise:

- Die Klasse `Order` muss 5 `Book`-Instanzvariablen haben.
- Programmieren Sie nur `get/set`-Methoden, die tatsächlich verwendet werden.
- Die Klasse `Test` darf **nicht** verändert werden (auch nicht abgegeben).

Aufgabe 3-3

1. Welchen Output erzeugt das folgende Programm? Überlegen Sie sich die richtige Antwort ohne das Programm abzutippen und auszuführen.

```
public class Foo{
    private int a = 0;
    public static int b = 0;

    public void incrementA(){ a++; }
    public void incrementB(){ b++; }

    public String toString(){
        return "a="+a+", b="+b;
    }

    public static void main(String[] args){
        Foo f1 = new Foo();
        Foo f2 = new Foo();
        f2.incrementA();
        f2.incrementB();
        System.out.println(f1);
        System.out.println(f2);
    }
}
```

2. Schreiben Sie die folgende while-Schleife zuerst in eine äquivalente do- und anschliessend in eine äquivalente for-Schleife um:

```
int i=1;
while(i<10){
    i++;
    System.out.println(i);
}
```

3. Die folgende Methode kann zu einem Problem führen. Beschreiben Sie das mögliche Problem und geben Sie an ob es bereits beim Kompilieren oder erst beim Ausführen auftreten kann. Wie kann man es beheben?

```
public static String dreierReihe(int limit){
    String result = "";
    for(int counter = 3; counter != limit; counter += 3){
        result = result + counter + " ";
    }
    return result;
}
```