

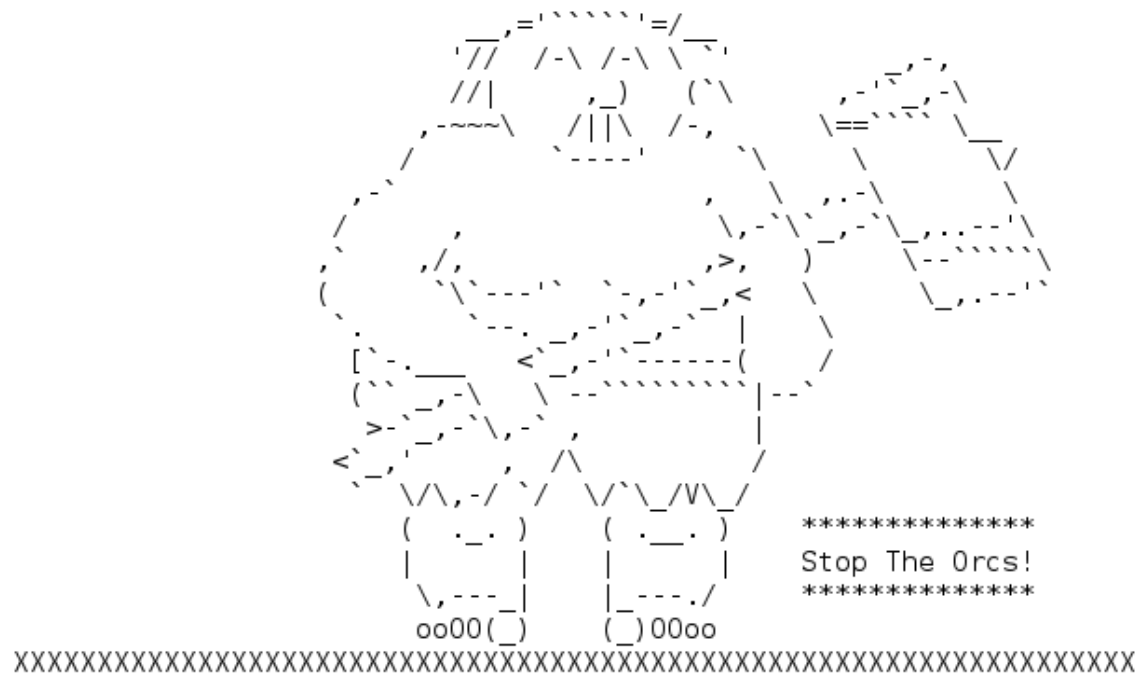
## Anmerkungen

- **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. Vorgegebene Dateien **nicht** ausdrucken!
- Am Anfang jeder Quellcode-Datei muss ein Kommentar stehen, der **Name(n) und Matrikelnummer(n)** der abgebenden Personen erhält.

## Aufgabe 3-1

*Ritter Kunibert sitzt in seiner Burg und wird von Orks belagert. Leider benutzen jene die gut ausgebauten Minenschächte um anzugreifen anstatt sich in eine offene Feldschlacht zu werfen welche einem Ritter natürlich lieber wäre. Normalerweise würde Kunibert die Schächte einfach mit Wasser fluten, aber weil seine Kunigunde darauf besteht dass er täglich badet (man lebt ja im zivilisierten Mittelalter und nicht in der Steinzeit), ist sein Wasservorrat sehr beschränkt und er kann nur einzelne Tunnel durchspühlen. Glücklicherweise besitzt Kunibert mehrere sehr empfindliche Seismographen, welche ihn in die ungefähre Richtung der Orks weisen und eine grobe Bestimmung des Terrains zulassen durch welche sie sich gerade bewegen.*

Sie sollen ein Spiel programmieren, bei dem sich der Computer zufällig einen Minenschacht "merkt". Aufgabe des Spielers ist es, diesen Schacht zu erraten. Zu Beginn soll sich das Spiel etwa wie folgt präsentieren: (Grafiken werden zur Verfügung gestellt, sollten Probleme mit Sonderzeichen auftreten, dürfen diese geändert werden!)



```
|a|b|c|d|e|f|g|h|i|j|k|l|
-----
L( ^ D° L( ^ ° D° ^ ) J° D° ^ ) J
| |#| |#| |#| |#| |#| |#| |
|#| |#| |#| |#| |#| |#| |#|
| |#| |#| |#| |#| |#| |#|
|#| |#| |#| |#| |#| |#| |#|
| |#| |#| |#| |#| |#| |#|
-----
```

Please enter your first guess: █

Der Spieler wählt ein Feld durch Eingabe von Spalten-Buchstaben. Also etwa

Next Guess? d

für Spalte d. Der Buchstabe soll unabhängig von Gross- und Kleinschreibung erfasst werden. Ist das Feld richtig geraten, gewinnt der Spieler und das Programm terminiert mit der Meldung

You win!

Andernfalls gibt der Rechner genau eine der folgenden Hilfen beim nächsten Spielzug aus:

- "Your Seismometer tells you that the orcs are to the left", falls das die zu erratende Spalte weiter links ist,
- "Your Seismometer tells you that the orcs are to the right", falls das die zu erratende Spalte weiter rechts ist,
- "Your Seismometer tells you that the orcs are very far away", falls das die zu erratende Spalte weiter als 5 Spalten weg ist,

- “Your Seismometer tells you that the orcs are under stone”, falls die Orks sich momentan unter Stein ('#') befinden,
- “Your Seismometer tells you that the orcs are under dirt”, falls die Orks sich momentan unter Erde (' ') befinden.

Benutzen Sie die dafür die HINTS Konstanten um diese Meldungen zu erzeugen.

Gibt es mehrere mögliche Hilfen, so soll der Computer zufällig genau eine davon auswählen. Implementieren Sie diese Entscheidung in

```
int calculateHint(int guessedMineShaftId, Random generator){
```

und benutzen Sie nur den übergebenen Generator um Zufallswerte zu erzeugen.

Nach jedem Zug rücken die Orks eine Zeile nach unten vor (auf den Spieler zu.) Sollte der Spieler nicht innerhalb von sechs (6) Versuchen das richtige Feld raten hat er verloren und das Programm terminiert mit der Meldung

```
You lose! The orcs attacked from mine shaft 'c'
```

wobei das Feld natürlich durch das generierte, geheime Feld ersetzt werden soll.

Auf ILIAS (Übungsserien → Übungsserie 3) finden Sie eine Datei GuessOrc.java, die Sie ergänzen sollen. Methoden zur Darstellung des Spielfelds inklusive Spielverlaufs, des Startbildschirms, sowie zur Konvertierung der Eingaben in deren Werte sind gegeben. Ebenso ist eine main-Methode vorimplementiert, die das Spiel startet. Alle vorimplementierten Methoden dürfen nicht verändert werden.

**Hinweis:** Ungültige Eingaben (z.B. “z1346” oder “w”) müssen geeignet behandelt werden!

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

**Beispiele:** Anstatt der gesamten Linie von Orks, schreiben wir in den Beispielen nur die tatsächliche Position als “\*” falls sich die unter Erde befinden, und als “@” falls sie sich unter Stein befinden.

```
|a|b|c|d|e|f|g|h|i|j|k|l|
-----
|#| |#| |#| |#| |#| |#| |
| |#| |#| |@| |#| |#| |#|
|#| |#| |#| |#| |#| |#| |
```

Mögliche Antworten nach geratener Spalte 'l':

1. “Your Seismometer tells you that the orcs are to the left”
2. “Your Seismometer tells you that the orcs are very far away”
3. “Your Seismometer tells you that the orcs are under stone”

```
|a|b|c|d|e|f|g|h|i|j|k|l|
-----
|#| |#| |#| |#| |#| |#| |
| |#| |#| |#| |#| |#| |#|
|#| |#|*|#| |#| |#| |#| |
```

Mögliche Antworten nach geratener Spalte 'a':

1. “Your Seismometer tells you that the orcs are to the right”
2. “Your Seismometer tells you that the orcs are under dirt”

## 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;
}
```