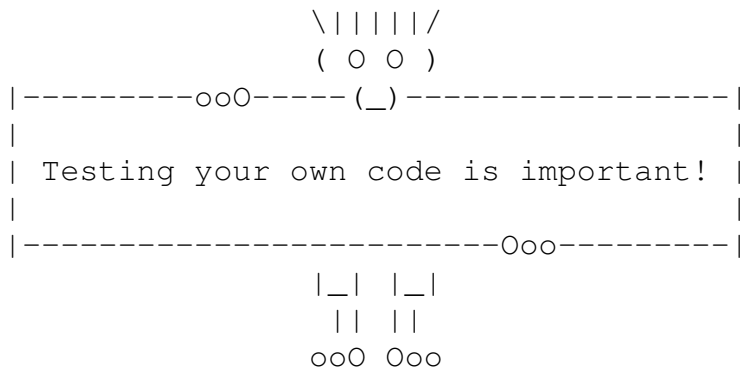
Prof. Dr. Christoph Bockisch
MSc Steffen Dick
Fachbereich Mathematik und Informatik
AG Programmiersprachen und -werkzeuge

Philipps Universität Marburg

Übungen zur Vorlesung
Objektorientierte Programmierung: Wintersemester 2021/2022

Nr. 4, Abgabe bis 22.11.2021

Aufgabe 4.1: Der Schilder-Klaus — 4 Punkte



Das ist Klaus. Klaus macht gerne Schilder mit wichtigen Nachrichten. Nachrichten, die auf den Schildern von Klaus vorkommen, können eine beliebige Länge bis zu 84 Zeichen haben.

Wir wollen Klaus mit seinen Schildern helfen, indem wir eine Funktion schreiben, die Klaus mit seinem neuesten Schild auf der Konsole anzeigt.

a) Schreiben Sie eine Funktion `void showBillboard(String message)`, die, mithilfe einer `while`-Schleife, Klaus mit seiner Nachricht auf der Konsole ausgibt.

   Hinweis: Sie dürfen die obige ASCII Art von Klaus kopieren. Denken Sie allerdings daran, dass die Schilder größer werden können, allerdings nicht seine Arme! Überlegen Sie sich, wo die Finger platziert werden müssten, damit Klaus keine längeren Arme bekommt. Sie dürfen außerdem **mehr als eine** `while`-Schleife verwenden.

b) Erstellen Sie ein Schild mit Klaus, auf dem `"Testing your own code is important"` steht. Erstellen Sie ein zweites Schild, auf dem `"Testing your program with all kinds of different input values is important!"` zu finden ist. Sehen die Schilder von Klaus auch genau so aus, wie erwartet?

c) Testen Sie `showBillboard` mit `""` und einem beliebigen String, der länger als 84 Zeichen lang ist.

Aufgabe 4.2: Graf Zahl | 4 Punkte

Implementieren Sie die folgenden Funktionen, die eine ganze Zahl nehmen und in das angegebene Format konvertieren.

Sie dürfen dabei nicht auf vordefinierte Methoden zurückgreifen, die diese Konvertierung übernehmen.

a) `String toBinary(int n)`, die eine positive Zahl entgegennimmt und sie in Binärdarstellung zurückgibt.

b) `String toOctal(int n)`, die eine positive Zahl entgegennimmt und sie in Oktaldarstellung zurückgibt.

c) `String toTwosComplement(int n)`, die eine Zahl zwischen -128 und 127 entgegennimmt und die Zahl in ihrer 8 Bit Zweierkomplement Darstellung zurückgibt

d) `void my_tests()`, die jede der obigen Funktionen ausreichend testet. Schreiben Sie so viele Tests für jede Funktion, wie Sie als nötig empfinden und geben Sie das Resultat auf der Konsole aus.

Beispiel:

```
1  ...
2  String binaryTest1 = "1";
3  String binaryTest1Result = toBinary(1);
4  if(binaryTest1.equals(binaryTest1Result)){
5    System.out.println("Binary test 1 passed");
6  }
7  else{
8    System.out.println("Binary test 1 failed");
9    System.out.println("Expected: "+binaryTest1);
10   System.out.println("Actual: "+binaryTest1Result);
11 }
12 ...
```

Hinweis: Sollten Sie Probleme bei der Konversion ins Zweierkomplement haben, können Sie

```
Integer.toBinaryString((<Number> & 0xFF) + 256).substring(1)
```

verwenden, um Ihre Konversion zu überprüfen.

Aufgabe 4.3: I wanna play a game | 4 Punkte

In dieser Aufgabe geht es darum, das Spiel mit den folgenden Regeln zu implementieren:

Es gibt genau zwei Haufen von Kieselsteinen, die eine unterschiedliche Anzahl an Kieselsteinen haben können. Beide Spieler müssen jede Runde eine Anzahl von Kieselsteinen aus dem Haufen mit den meisten Kieselsteinen entnehmen. Jeder Spieler hat in jedem Zug die Wahl, ob zwei oder drei Kieselsteine aufgenommen werden sollen, dabei können allerdings nicht mehr Kieselsteine aufgenommen werden als vorhanden sind. Das Spiel geht solange, bis ein Spieler keine Kieselsteine mehr aufnehmen kann, also, wenn beide Haufen weniger als zwei Kieselstein enthalten. Verloren hat dann, wer keine Kieselsteine mehr aufnehmen konnte.

a) Schreiben Sie eine Methode `boolean canIWin(int pileA, int pileB)`, die `true` zurückgibt, falls es eine Abfolge von Zügen gibt, mit der der aktuelle Spieler gewinnen können.

   Hinweis: Implementieren Sie diese Methode rekursiv: Ich kann gewinnen, wenn ich x aus Haufen y nehme und mein Gegner nicht gewinnt.

b) Schreiben Sie eine Methode `void whatShouldIDo(int pileA, int pileB)`, die Ihnen eine Zugempfehlung auf der Konsole ausgibt.

   Hinweis: Bei der Implementierung dieser Methode dürfen Sie davon ausgehen, dass Sie aktuell am Zug sind.

c) Überprüfen Sie Ihre Implementierungen mit geeigneten Werten und simulieren Sie ein Spiel, das mindestens 5 Runden anhält.

   Geben Sie bei Ihrer Abgabe die nötigen Aufrufe zum Testen und Ihre Simulation mit an.