Prof. Dr. Stefan Brass
M.Sc. Mario Wenzel

Martin-Luther-Universität Halle-Wittenberg
Institut für Informatik
Wintersemester 2018/19

Objektorientierte Programmierung: Hausaufgabenblatt 9

Abgabe: 07.01.2019, 11:00

Dieses Übungsblatt enthält (Teil)Aufgaben mit Bonuspunkten. Diese Punkte gehen auf Ihr Punktekonto, haben aber keinen Einfluss auf die Gesamtzahl der zu erreichenden Punkte. Das Ziel dieser Übungsblattes ist es, dass Sie ihre Kenntnisse zur Aufrufsemantik von Funktionen festigen.
Diese Übungsserie finden Sie im Stud.IP auf der Seite der Veranstaltung unter Lernobjekte → Kurs in ILIAS → Hausaufgabenblatt 9.

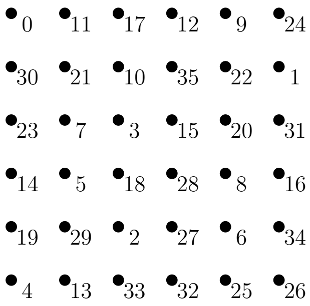
Hausaufgabe 1 (Bonus): (0+4 Punkte)

Bitte beachten Sie: die Einsendung der Aufgabe erfolgt über die ILIAS-Plattform.

In der Vergangenheit haben Sie bereits den Pseudozufallszahlengenerator der Java-Standardbibliothek genutzt. Dieser Generator ist ein sog. Kongruenzgenerator, der die nächste Pseudozufallszahl X_{n+1} wie folgt generiert: $X_{n+1} = (aX_n + c) \bmod m$, wobei X_0 ein initialer Wert ist, der oft mit der Systemzeit belegt wird. Die Zahlen m, c und a müssen geschickt gewählt werden.
Im Folgenden Programmausschnitt ist ein Kongruenzgenerator nach diesem Muster implementiert. Generieren Sie die ersten 8 Zufallszahlen, also werten Sie 8 Mal hintereinander die Funktion `nextRand()` aus. Verbinden Sie die dazu passenden Punkte im Gitter in dieser Reihenfolge miteinander.

```
static int seed = 4;
static int nextRand() {
  seed = (25 * seed + 5) % 36;
  return seed;
}
```

nextRand() no.	Wert
1	
2	
3	
4	
5	
6	
7	
8	



Hausaufgabe 2: (6+2 Punkte)

Bitte beachten Sie: die Einsendung der Aufgabe erfolgt über die ILIAS-Plattform.
Der Weihnachtsmann muss wissen, welche Kinder artig und welche unartig waren, um passend Geschenke oder Kohle zu verteilen. Da ein Kind entweder artig oder unartig war reicht es, eine Liste der unartigen Kinder zu führen. Jedes nicht-unartige Kind war artig. Um das zu organisieren, hat sich der Weihnachtsmann eine Java-Klasse `NaughtyOrNice` geschrieben, die ein `int` Array `naughty` aller unartigen Kinder enthält (der Weihnachtsmann hat zur besseren maschinellen Verarbeitung alle Kinder durchnummeriert). Leider waren viel mehr Kinder unartig, als der Weihnachtsmann sich erhofft hatte (unser Beispiel enthält nur einen Ausschnitt aller unartigen Kinder), und so muss er einen schlauen Algorithmus benutzen, um die viel zu große Liste der unartigen Kinder schnell zu durchsuchen.

```
public class NaughtyOrNice {
  private static int[] naughty = {123, 136, 139, 161, 177,
                                  286, 308, 325, 339, 349,
                                  399, 434, 488, 513, 536,
                                  573, 600, 747, 849, 866};

  public static boolean wasNaughty(int kid) {
    return wasNaughtyBin(kid, 0, naughty.length - 1);
  }

  public static boolean wasNice(int kid) {
    return !wasNaughtyBin(kid, 0, naughty.length - 1);
  }

  private static boolean wasNaughtyBin(int kid, int left, int right) {
    if (left > right) return false;
    int middle = left + (right - left) / 2;
    if (naughty[middle] < kid)
      return wasNaughtyBin(kid, middle + 1, right);
    else if (naughty[middle] > kid)
      return wasNaughtyBin(kid, left, middle - 1);
    else
      return true;
  }
}
```

Als Schnittstelle hat die Klasse `NaughtyOrNice` die öffentlichen Methoden `boolean wasNice(int kid)` und `boolean wasNaughty(int kid)`, mit denen der Weihnachtsmann herausfinden kann, ob ein Kind auf seiner Liste steht.

Vollziehen Sie für die Aufrufe `wasNice(286)`, `wasNice(760)` und `wasNaughty(536)` die rekursiven Aufrufe von `wasNaughtyBin` nach und geben Sie jeweils an, mit welchen Parametern die Funktion aufgerufen wurde und welchen Wert Sie zurückgibt. Geben Sie auch an, mit welchem Wert die Variable `middle` jeweils belegt wird und welchen Rückgabewert der ursprüngliche Aufruf hat.

wasNice	286	—	—	—	
wasNaughtyBin	kid	left	right	middle	Rückgabewert
wasNaughtyBin	286				siehe Unten
wasNaughtyBin	286				siehe Unten
wasNaughtyBin	286				siehe Unten
wasNaughtyBin	286				

wasNice	760	—	—	—	
wasNaughtyBin	kid	left	right	middle	Rückgabewert
wasNaughtyBin	760				siehe Unten
wasNaughtyBin	760				siehe Unten
wasNaughtyBin	760				siehe Unten
wasNaughtyBin	760				siehe Unten
wasNaughtyBin	760			—	

wasNaughty	536	—	—	—	
wasNaughtyBin	kid	left	right	middle	Rückgabewert
wasNaughtyBin	536				siehe Unten
wasNaughtyBin	536				

Bonus: Damit der Algorithmus funktioniert, den der Weihnachtsmann hier benutzt, muss das Array, in dem gesucht wird, gewisse Bedingungen erfüllen. Kreuzen Sie eine oder mehrere Bedingungen an, die zwingend notwendig sind, damit der Algorithmus funktioniert:

- Das Array darf nur positive Zahlen enthalten
- Das Array darf nur Zahlen enthalten, die gleich viele Stellen haben
- Das Array muss eine gerade Länge haben
- Das Array muss sortiert sein
- Das Array muss duplikatfrei sein

Hausaufgabe 3: (2+6 Punkte)

Bitte beachten Sie: die Einsendung der Aufgabe erfolgt nur im YAPEX.
Open exercises via code → Freigabecode: `5cool663j24f-c91a`

In dieser Aufgabe ist Ihre Kreativität gefragt. Sie sollen einen Weihnachtsbaum zeichnen. Ihr Programm bekommt eine ungerade Zahl $s \geq 3$ als Eingabe. Dies ist die Breite des Weihnachtsbaums. Geben Sie einen Weihnachtsbaum aus, der s Zeichen breit ist, sich nach oben je Zeile um 2 Zeichen verjüngt und einen Stamm von 3 Zeichen Breite und 2 Zeilen Höhe hat. Für die Eingabe $s = 11$ wäre dies also eine gültige Ausgabe:

```
     #
    ###
   #####
  #######
 #########
###########
    ###
    ###
```

Bonus: Hier ist Ihre Kreativität gefragt. Schmücken Sie den Baum mit anderen nicht-Leerzeichen Ihrer Wahl. "Behängen" Sie Ihren Baum mit Kugeln, Zuckerstangen und Girlanden und setzen Sie eine Spitze auf. Sie können auch den Baumstamm hübscher machen, indem Sie ihn durch andere Zeichen ersetzen. Für kreativen Schmuck und Gestaltung vergeben wir bis zu 6 zusätzliche Punkte.
Unsere YAPEX-Tests überprüfen lediglich pro Zeile die Anzahl der Leerzeichen und darauffolgenden nicht-Leerzeichen. In diesem Rahmen können Sie Ihre Ausgabe gestalten. Sie sollten jedoch den Baum algorithmisch generieren und verschönern, da wir über versteckte Tests auch weitere Baumgrößen in unsere Bewertung einbeziehen.

Das OOP-Team wünscht Ihnen und Ihren Liebsten erholsame und besinnliche Weihnachten, einen guten Rutsch ins neue Jahr und schöne Feiertage.