
Programmierstarthilfe SS 2009
Fakultät für Ingenieurwissenschaften und Informatik

11. Blatt

Für die Woche vom 6.7. bis zum 10.7.2009 (KW 28)

Organisatorisches

Die Webseiten zur Veranstaltung sind unter <http://www.uni-ulm.de/in/mi/lehre/2009ss/programmierstarthilfe.html> zu finden.

Diese Woche lernen wir die API (*Application Programming Interface*) von Java näher kennen. Eine Sprache an sich ist nämlich nur ein Bruchteil von dem was man zum Programmieren und Arbeiten kennen muss. In APIs sind große Mengen von nützlichen Funktionalitäten vorhanden, die man sonst alle neu schreiben müsste.

1 Üben für die Klausur

In der Klausur wird auf Papier programmiert. Deshalb sollst du die Aufgaben aus diesem Abschnitt auch nur mit Blatt und Stift lösen. Bei jeder Aufgabe steht wieviel Prozent einer Klausur sie ausgemacht hat und wie lange du dafür brauchen solltest. Schalte den Monitor während der Aufgabe aus und miss die Zeit von Start bis Ende.

1.1 Tiefe von Binärbäumen (5%, 6 Minuten)

Start: _____

Die folgende Klasse BaumKnoten sei gegeben:

```
public class BaumKnoten {
    int inhalt;
    BaumKnoten linkerNachfolger;
    BaumKnoten rechterNachfolger;

    public BaumKnoten(int inhalt) {
        this.inhalt = inhalt;
    }
}
```

Schreiben Sie eine Methode `int tiefe(BaumKnote b)` welche die maximale Tiefe eines übergebenen binären Baums ermittelt.

Ende: _____

1.2 Listen (7%, 8 Minuten)

Start: _____

Folgende Klassen sind gegeben:

```

1 class ListenElement {
2     int inhalt;
3     ListenElement nachfolger;
4
5     public ListenElement(int inhalt) {
6         this.inhalt = inhalt;
7     }
8 }
9
10 public class Liste {
11     ListenElement listenKopf;
12
13     public Liste(int inhalt) {
14         this.listenKopf = new ListenElement(inhalt);
15     }
16
17     public void fuegeHinzu(ListenElement element) { ... }
18 }
    
```

Die Methode fuegeHinzu kann als gegeben vorausgesetzt werden.

Implementieren Sie eine Methode ungeradeElementeHerausfiltern(), deren Rückgabewert eine neue Liste ist, die ausschließlich die ungeraden Werte (bezüglich inhalt) der alten Liste enthält. Die Methode soll keine Seiteneffekte auf die eigentliche Liste haben.

Ende: _____

2 Aufgaben

2.1 Funktionen für Strings aus der API

Öffne die API-Dokumentation (<http://java.sun.com/javase/6/docs/api/>) und suche die Klasse String. Suche nun zu den folgenden Problemen eine passende Methode heraus und teste sie anhand eines kleinen Testfalls. Wichtig: Es soll in dieser Aufgabe keine Methode selbst implementiert werden, beschränke dich auf die bereits existierenden Methoden aus der API.

- Wie kann man prüfen, ob ein String leer ist?
- Wie bekommt man die Länge eines Strings?
- Wie entfernt man überflüssige Leerzeichen an Anfang und Ende eines Strings?

- d) Wie wandelt man einen String in eine Text, der nur aus Großbuchstaben besteht, um?
- e) Wie testet man, ob ein String mit einer bestimmten Zeichenkette endet?
- f) Wie vergleicht man einen String mit einem zweiten, ohne dabei auch Klein-/Großschreibung zu achten?
- g) Wie bekommt man den Character eines Strings an der 3. Stelle?
- h) Wie findet man die Stelle des letzten Vorkommens eines Character in einem String?
- i) Wie ersetzt man in einem String alle Vorkommen einer Zeichenkette durch eine andere Zeichenkette?
- j) Angenommen, ein gegebener String würde einen Satz enthalten. Wie bekommt man am einfachsten ein Array der darin enthaltenen Wörter? Überlege dir zunächst, was die Wörter voneinander trennt.

2.2 Die Klasse System

Schauen wir uns nun die API-Dokumentation der Klasse `System` an. Wie kann man das folgende Programm damit implementieren?

- a) Zu Beginn soll eine Begrüßung ausgegeben werden. Was hat das mit der Klasse `System` zu tun?
- b) Dann soll ein Zeilenumbruch ausgeführt werden. Beachte: Bei Linux erzeugt das Zeichen `line-feed (\n)` einen Zeilenumbruch, bei MacOS das Zeichen `carriage-return (\r)` oder `line-feed` (aber nicht beides zusammen) und bei Windows die Kombination beider Zeichen `(\r\n)`. Wir wollen, dass das Programm überall funktioniert. Tipp: Die Systemvariable für das systemspezifische Zeilenumbruchzeichen heißt `line.separator`.
- c) Lass nun den Garbage-Collector laufen und informiere dich darüber, was dieser tut.
- d) Miss die Zeit, die der Garbage-Collector benötigte. Das geht am einfachsten, indem du vorher und nachher die Systemzeit speicherst und die Differenz bildest.
- e) Wenn der Garbage-Collector zu lange braucht (nimm dir da eine beliebige Millisekundenzahl), dann soll eine Fehlermeldung ausgegeben werden. Fehlermeldungen schreibt man als guter Programmierer nicht auf die Standardausgabe, sondern auf die Standard-Fehler-Ausgabe.
- f) Wenn kein Fehler auftrat soll das Programm normal enden, wenn doch, soll es mit `Exit-Status -1` enden. Der Fehlercode signalisiert anderen Programmen, dem Benutzer und dem System in fortgeschritteneren Anwendungen, ob dein Programm fehlerfrei lief.

2.3 Klasse zur Generierung von Passwörtern

Es soll eine Klasse `PasswordGenerator` zur Erzeugung von zufälligen Passwörtern geschrieben werden. Der Konstruktor dieser Klasse soll drei `boolean`-Werte übergeben bekommen, die angeben, ob kleine Buchstaben, große Buchstaben sowie Ziffern vorkommen dürfen. Hierbei muss mindestens eine Zeichenart aktiviert sein, ansonsten sollen standardmäßig zumindest die kleinen Buchstaben verwendet werden. Eine Methode `generatePassword(int length)` soll anschließend entsprechende, zufällige Passwörter erzeugen und diese als `String` zurückgeben.

```

1 // PasswordGenerator-Objekt erzeugen mit
2 // kleinen und grossen Buchstaben
3 PasswordGenerator meinPasswortMacher = new
4   PasswordGenerator(true,true,false);
5 // Passwort der Laenge 8 erzeugen
6 String passwort1 = meinPasswortMacher.generatePassword(8);
    
```

a) Für diese Aufgabe benötigst du Zufallswerte. Bisher haben wir dafür die Methode `Math.random()` genutzt. Dieses Mal soll jedoch die fertige Java-Standardklasse `Random` benutzt werden. Informiere dich in der API-Dokumentation im Internet über ihre Verwendung und ihre Methoden. Überlege dir anschließend, wie du die Klasse sinnvoll für dein Problem benutzen kannst.

Suche im Internet nach „java api class random“ oder folge dem Link:
<http://java.sun.com/javase/6/docs/api/java/util/Random.html>

b) Implementiere die Klasse `PasswordGenerator` und teste sie. Achte hierbei auf die verschiedenen Zeichenarten, die man - auch kombiniert - auswählen kann.

c) Erzeuge eine weitere Methode `generatePasswordArray(int passwordLength, int passwords)`, die ein Array von Strings mit erzeugten Passwörtern zurückgeben soll. Hierbei soll die Länge sowie die Anzahl der zu erzeugenden Passwörter übergeben werden.

d)** Oft sind zufällige Passwörter sehr schwer zu erraten, aber auch kaum zu merken. Aus diesem Grund ist es hilfreich, festzulegen, dass sich bei Buchstaben im Passwort immer Vokale und Konsonanten abwechseln, um das damit erzeugte Passwort aussprechbar zu machen. Schaffst du es, trotzdem auch noch Ziffern unterzumischen, falls diese aktiviert wurden?

3 Ergebnissicherung

Thema	Gelernt habe ich	Verstanden	Wiederholen/erklären?

4 Für nächste und die letzte Woche

Nächste Woche werden wir den bisherigen Stoff in Probeklausuren üben. Schau dir die bisherigen Blätter noch einmal an und bring ein paar Stifte mit.

In der letzten Woche werden wir Fragen zum Stoff beantworten. Lies das Skript noch einmal durch und schreibe dir Fragen, die noch offen sind, auf.

Falls noch Nachfrage besteht können wir in der letzten Woche auch nochmal eine Probeklausur anbieten.