

Objektsammlungen II & Bibliotheksklassen I

Lernziele

- Sie sammeln Erfahrung mit Objektsammlungen flexibler und fester Grösse.
- Sie setzen bei einfachen Problemstellungen, zu deren Lösung eine oder mehrere Objektsammlungen eingesetzt werden müssen, die jeweils passende Objektsammlung ein.
- Sie können mit (mehrdimensionalen) Arrays umgehen und setzen diese situationsgerecht ein.
- Sie können eine bisher unbekannte Klasse mit Hilfe der zugehörigen Klassendokumentation in Ihrem Programm gezielt einsetzen.
- Sie können eine Klasse von Grund auf so programmieren, dass Sie eine gegebene Spezifikation erfüllt.

Aufgabe 1 (auf Papier!)

Der Aufruf `holeDatenreihe()` gibt eine Datenreihe als Text zurück. Der Text besteht aus durch Leerzeichen getrennte Ganzzahlen. Speichern Sie diese Datenreihen in einer geeigneten Objektsammlung und geben Sie anschliessend die Zahlen der Datenreihe so aus, dass pro Ausgabezeile genau eine Ganzzahl ausgegeben wird.

Hinweis: Die Methoden der Klasse `String` könnten hier hilfreich sein!

```
//TODO: Objektsammlung deklarieren und erzeugen
public String[] sammlung;
while(hatMehrDaten()) {
    String datenreihe = holeDatenreihe();
    //TODO: datenreihe in Objektsammlung speichern
    sammlung = datenreihe.split(" ");
    //TODO: datenreihe ausgeben (eine Zahl pro Zeile)

    for(String zahl : sammlung)
    {
    }
} System.out.println(zahl);
}
```

Aufgabe 2

Importieren Sie für diese Aufgabe das BlueJ-Projekt 04_Praktikum-2_Messen.

In dieser Aufgabe sollen Sie messen, wie lange es im Durchschnitt dauert, eine bestimmte Anzahl Zufallszahlen zu generieren und zu sortieren. Studieren Sie dazu zuerst die Schnittstelle der Klasse Messkonduktor. Die Interna der Klasse brauchen Sie nicht zu interessieren.

- a) Die Methode `messungenDurchfuehren()` gibt einen Array zurückgibt. Ist das notwendig bzw. sinnvoll?
Nein es ist nicht sinnvoll, da das Array als referenz übergeben wird, und die Werte auch ohne `return` im Array verändert werden.

Erweitern Sie anschliessend die Klasse `MessApp` um folgende Funktionalität:

- b) Führen Sie 10 Messreihen à 20 Messungen durch. Die Laufzeiten sollen in einem zweidimensionalen Array gespeichert werden.
Pro Messung sollen 400'000 Zufallszahlen generiert und sortiert werden.
- c) Berechnen Sie nun anhand der Messergebnisse folgende Mittelwerte und geben Sie diese aus:
- i) Mittelwert pro Messreihe
 - ii) Mittelwert pro Messung (über alle 1. Messungen pro Messreihe, über alle 2. Messungen pro Messreihe etc.)

Betrachten Sie die Messergebnisse und vergleichen Sie diese mit den Messungen von Kollegen. Stellen Sie Unterschiede fest?

Betrachten Sie die erhaltenen Mittelwerte. Gibt es grössere Abweichungen zwischen diesen? Woran könnte das liegen?

Aufgabe 3

Importieren Sie für diese Aufgabe das BlueJ-Projekt 05_Praktikum-1_Pruefung.

Die Klasse `Pruefungsergebnis` ist gegeben. Vervollständigen Sie die Klasse `Pruefungsverwaltung` an den bezeichneten Stellen so, dass die nachfolgenden Anforderungen eingehalten sind.

Ausgabeformat: Für das folgende Codestück:

```
Pruefungsverwaltung verwaltung = new Pruefungsverwaltung();  
verwaltung.speichereErgebnis(new Pruefungsergebnis("Max Muster", 5.3333));  
verwaltung.speichereErgebnis(new Pruefungsergebnis("Susi Mueller", 3.74));  
verwaltung.speichereErgebnis(new Pruefungsergebnis("Heinz Moser", 4));  
verwaltung.druckeAntworttexte();
```

Sie müssen dabei folgende Überlegungen anstellen:

Herzliche Gratulation Max Muster! Sie haben an der Pruefung eine 5.5 erzielt und somit bestanden!
Susi Mueller, Sie haben an der Pruefung eine 3.5 erzielt und sind somit durchgefallen!
Herzliche Gratulation Heinz Moser! Sie haben an der Pruefung eine 4.0 erzielt und somit bestanden!

- Was für eine Datenstruktur verwenden Sie zum Speichern der Pruefungsergebnis-Objekte?

Es ist gute Praxis das Lösen einzelner Probleme in eigene Methoden auszulagern. Ganz getreu dem Grundsatz „teile und herrsche“. Dies hilft Ihnen dabei die Komplexität zu minimieren und macht den Code besser lesbar.

Zum Ausprobieren der Prüfungsverwaltung verwenden Sie die Klasse ZufaellicheNotengebung . Diese muss allerdings noch ergänzt werden, so dass sie folgende Rahmenbedingungen erfüllt:

- Die Methode fuehreAus der Klasse ZufaellicheNotengebung soll für Tests ein Set von Prüfungsergebnissen mit zufällig gewählten Noten erzeugen und anzeigen. Ergänzen Sie die Klasse um den hierfür noch zusätzlich notwendigen Code.
- Die Noten in der Ausgabe müssen auf 0.5 gerundet werden.
- Die Noten müssen als Zahl ausgegeben werden.
- Der Ausgabetext muss sich auf das Prüfungsergebnis (bestanden, nicht bestanden) und den Kandidaten beziehen.

Weiterführend (Optional)

Erweitern Sie das Programm so, dass die Testergebnisse über die Konsole vom Benutzer eingelesen werden.

Aufgabe 4

Folgende Methode der Klasse Pruefungsverwaltung ist gegeben:

```
private double rundeAufHalbeNote(double note) {  
    return Math.round(note * 2) / 2.0;  
}
```

Wieso ist es wichtig 2.0 zu schreiben und wieso würde 2 nicht das gewünschte Ergebnis liefern?

Wenn man nur 2 angeben würde, wäre das Resultat ein Int, da round als Resultat einen Int oder long zurück gibt. Je nach zahl wäre dann das Resultat verfälscht. Bsp. 2/3