Prof. Dr. Franz Schweiggert Institut für Angewandte Informationsverarbeitung 2. Juli 2008
Christoph Ott Blatt 10

Einführung in die Programmierung (SS 2008)

Abgabetermin: 10. Juli 2008

1 Sortieren

Gegeben seien folgende ehrenwerte Personen:

Kermit	Ernie	Bert	Samson	Kruemelmonster	Tiffy	Grobi

1.1 Wie erklär ich Kindern den Insertion-Sort? (1P)

Gehen wir mal davon aus, dass Euer kleiner Bruder Euch fragt, was Ihr denn so an der Uni lernt. Zumindest den Insertion-Sort könnt Ihr ihm erklären. Nehmt Euch 7 kleine Zettel und ein Kind unter 10 Jahren her und schreibt die Namen obiger Personen darauf (auf den Zettel, nicht auf das Kind!). Legt die Zettel in der Reihenfolge wie in obiger Tabelle auf den Tisch und führt besagtem Kind die Vertauschungsschritte des Insertion-Sort Schritt für Schritt vor. Am Ende sollten obige Personen alphabetisch sortiert sein. Wenn Ihr Glück habt, erfahrt Ihr dabei noch was über obige Personen. Auf jeden Fall könnt Ihr gewiss sein, dass besagtes Kind jeglichen Respekt vor Eurem Studium verloren hat. Hat das Kind den Sortieralgorithmus verstanden, so folgt Stufe 2: Erkärt nun auch Eurem Tutor wie der Insertion-Sort funktioniert.

1.2 Programmcode: Insertion-Sort? (1P)

Nun wieder ernsthaft. Nehmt den Insertion-Sort-Algorithmus (Skript S.138) her, versucht ihn zu verstehen und erklärt Eurem Tutor warum dieser Code tut, was Ihr mit den Zetteln demonstriert habt.

1.3 Obige Personen sortieren (3P)

Schreibt eine Methode *insertionSort()*, die einen Array von Strings entgegennimmt und diesen alphabetisch absteigend (d. h. mit 'z' beginnend) sortiert. Überprüft diese Methode, in dem Ihr obige Namen in einem String-Array speichert und diesen Array vor und nach dem Aufruf von *insertionSort()* ausgebt.
Tipp: Strings können nicht mit > und < und sollten auch nicht mit == verglichen werden. Hierzu gibt es die Methoden *compareTo()* und *equals()*, die wie in Beispiel *CompareStrings.java* verwendet werden können.

2 Suchen

2.1 Suchen nach dem eigenen Namen (2P)

Da Ihr nun sowieso schon Eure Zettel gebastelt habt, schreibt auf einen weiteren Zettel Euren eigenen Namen und erklärt Eurem Tutor anschaulich, wie (und unter welchen Umständen) man nach diesem String in obigem Array sowohl linear als auch binär suchen kann. Erklärt Eurem Tutor anschließend die unterschiedlichen Möglichkeiten der Implementierung.

2.2 Nach Personen suchen (3P)

Schreibt eine Methode *binarySearch()*, der der mit Teilaufgabe 1.3 sortierte Array sowie ein weiterer String übergeben wird und die *true* zurückliefert, falls der String im Array enthalten ist; bzw. *false*, falls der String nicht im Array enthalten ist. Wählt dazu bitte eine binäre Such-Methode. Erklärt Eurem Tutor, warum Euer Programm tut, was es tun soll.
Vorsicht: Der übergebene Array ist alphabetisch absteigend sortiert!

Viel Erfolg!