Aufgaben zu Array, Listen, etc

1. Erstellen Sie jeweils ein Array, eine ArrayList und eine Hashtable/Dictionary mit 1 mio. unsortierten Einträgen. Benutzen Sie dazu eine Schleife sowie die Random-Klasse.  
   Messen Sie jeweils die Zeit

* Zum Erstellen der jeweiligen Ausgangsstruktur
* Zum Finden eines beliebigen Wertes. Um Zufallsergebnisse zu vermeiden, führen Sie diesen Vorgang 10-mal hintereinander mit anderen Werten durch.

1. Berechnen Sie die Summe der Zahlen im Array.

**class** **Uebung1**

{

public static void Main ( String[] args )

{

int[] arr = {0, 1, 2, 3};

int summe = #Has to be done

Console.WriteLine( "Summe aller Zahlen = " + summe );

}

}

1. Begutachten Sie das folgende Programm:

**class** **Uebung2**

{

public static void Main ( String[] args )

{

int[] arr = {13, -4, 82, 17};

int[] doppelt;

Console.WriteLine( "Urspruengliches Array: "

+ arr[0] + " " + arr[1] + " " + arr[2] + " " + arr[3] );

// Konstruieren Sie ein Arrayobjekt für doppelt.

Vervollständigen Sie das Programm, so dass ein neues Array doppelt konstruiert wird. Kopieren Sie jetzt die Werte von arr nach doppelt, aber machen Sie die Werte in doppelt zweimal so groß als wie sie in arr sind.

1. Begutachten Sie das folgende Programm:

**class** **Uebung3**

{

public static void main ( String[] args )

{

int[] arrA = { 13, -22, 82, 17};

int[] arrB = {-12, 24, -79, -13};

int[] summe = { 0, 0, 0, 0};

// Addieren Sie die Werte der entsprechenden Slots von arrA und arrB,

// und stellen Sie das Ergebnis **in** den entsprechenden Slot von summe.

Console.WriteLine( "Summe: "

+ summe[0] + " " + summe[1] + " " + summe[2] + " " + summe[3] );

}

}

Vervollständigen Sie das Programm mit vier Zuweisungsanweisungen, so dass jeder Slot von summe die Summe der entsprechenden Slots von arrA und arrB enthält. D.h., addieren Sie Slot 0 von arrA mit Slot 0 von arrB und stellen Sie das Ergebnis in den Slot 0 von summe und so weiter.

1. Vervollständigen Sie das folgende Programm, so dass es die Summe aller Elemente des Arrays berechnet. Schreiben Sie das Programm, so dass es sogar dann funktioniert, wenn die Dimensionen der Zeilen und Spalten geändert werden.

class ArraySumme

{

public static void main ( String[] args ) throws IOException

{

int[][] data = {

new int[]{ 3, 2, 5}, //+10

new int[]{ 1, 4, 4, 8, 13}, //+30

new int[]{ 9, 1, 0, 2}, //+12

new int[]{ 0, 2, 6, 3, -1, -8 } //+ 2

} ; //summe: 54

// Summe deklarieren

// Summe berechnen

for ( int zeile=0; zeile < data.length; zeile++)

{

for ( int spalte = 0; spalte < ???; spalte++)

{

}

}

// Summe ausgeben

Console.WriteLine( );

}

}

1. Arraylists sind sog. doppelt verkettete Listen, bei denen ein Element sowohl den Vorgänger als auch seinen Nachfolger kennt.

Wie würden Sie diese Funktionalität implementieren, wenn Sie für eine Liste von Personen eine eigene verkettete Liste implementieren müssten, ohne eine ArrayList oder sonstige Klasse aus dem Collections-Namespace verwenden dürften.

1. Welche Ausgabe erzeugt folgendes Codeschnipsel?

int [] alpha = new int [2];

int [] beta;

beta = alpha;

Console.WriteLine("alpha equals beta ist " + alpha.Equals(beta));

Console.WriteLine("alpha hat " + alpha.Length +" Komponenten");

Aufgaben zu ArrayLists

1. Gegeben ist folgender Quellcode:

public **class** **ClassA**

{

public int Prop;

public ClassA(int x) {

Prop = x;

}

}

*###################################*

ClassA obj1 = new ClassA(1);

ClassA obj2 = new ClassA(2);

ArrayList col = new ArrayList();

col.Add(obj1);

col.Add(obj2);

*#####################################*

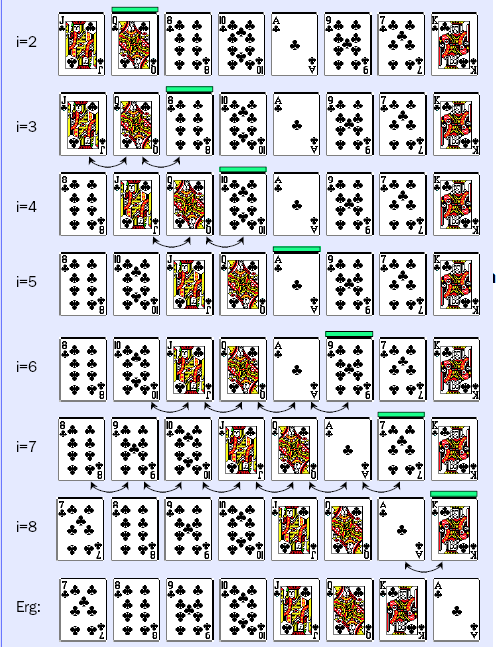
ClassA obj3 = new ClassA(3);

col.Insert(1, obj3);

Was passiert mit dem Element obj2, das vorher die Position des Index 1 einnahm? Iterieren Sie durch die Liste mit Hilfe einer foreach-Schleife und bewerten Sie das Ergebnis.

1. Schreiben Sie ein Programm, das zwei sortierte Arraylisten aus Integer-Daten in eine einzige sortierte Liste überführt. Die merge()-Methode des Programms sollte eine Referenz auf beide ArrayLists erhalten und die sortierte neue ArrayList als Referenz zurückgeben.

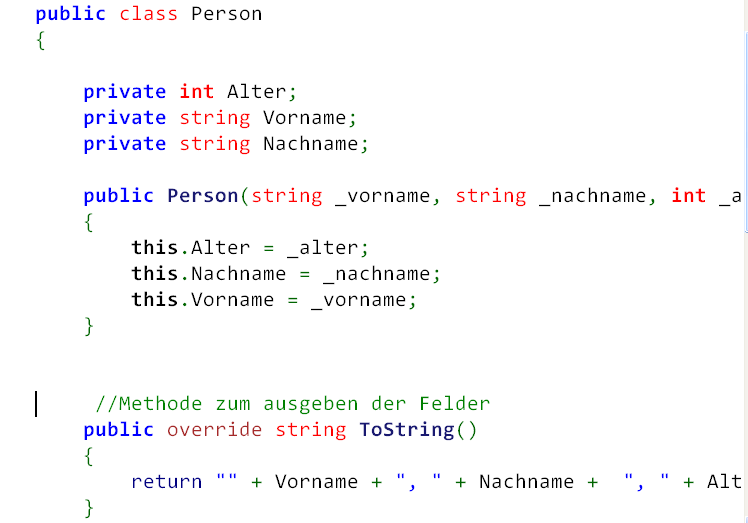
Sortierverfahren

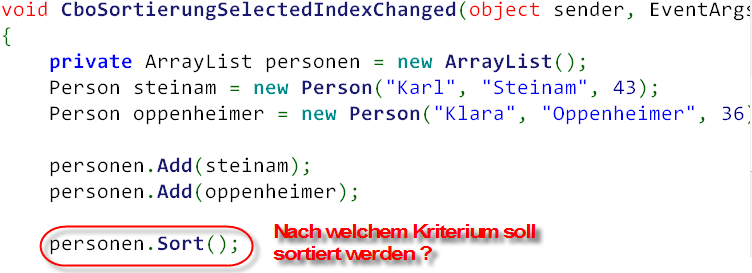
1. Der wohl einfachste Sortieralgorithmus trägt den Namen **Selection Sort** und basiert auf zwei ineinander geschachtelten Schleifen, von denen die äußere alle Elemente bis auf das letzte durchläuft. Ausgehend vom aktuellen Element der äußeren Schleife werden in der inneren Schleife alle Elemente bis zum Ende des Arrays durchlaufen, um dort das jeweils kleinste Element zu suchen. Dieses kleinste Element wird dann mit dem aktuellen Element der äußeren Schleife ausgetauscht.  
     
   Implementieren Sie dieses Sortierverfahren mit Hilfe von C#.   
   Testen Sie das Verfahren an einem int-Array mit den Werten (123,43,567,54, 89,1,43,7)
2. Beim so genannten „Sortieren durch Einfügen“ (Insertion Sort) werden die Elemente beginnend mit dem zweiten Element durchlaufen. Bei jedem Element wird überprüft, ob es nicht besser vor eines der Vorgänger-Elemente passt. Ist das aktuelle Element kleiner als eines seiner Vorgänger-Elemente, wird es einfach vor dieses Vorgänger-Element platziert. Dazu muss allerdings Platz im Array geschaffen werden. Jedes Vorgänger-Element, das kleiner als das aktuelle Element ist, wird dazu einfach bei jedem Schritt eine Position nach rechts gerückt.(siehe Abbildung ). Die Implementierung dieses Algorithmus sieht dann so aus:

Weiterführende Aufgabe

Sortieren und Suchen in Listen

.Sortieren und Suchen in Listen sind Standardaufgaben, die immer wieder anfallen. Sortieren und Suchen hängen eng miteinander zusammen. Listen werden häufig sortiert, um anschließend schneller Suchanfragen auf die sortierte Liste stellen zu können. Collections bieten zum Sortieren bereits eine Sort()-Methode an. Diese muss jedoch in der Lage sein, alle möglichen Objekttypen sortieren zu können. Eine weitere Problematik besteht darin, dass es nur bei einfachen Datentypen relativ klar ist, nach welchem Kriterium sortiert werden soll. Doch nach welchem Kriterium soll ein komplexer Datentyp sortiert werden?





Informieren Sie sich darüber, wie dieses Problem in .NET-Framework gelöst wird und setzen Sie das oben angegebene Beispiel entsprechend um.