

Instuderingsfrågor, del D

Uppgift 1.

I vilka av nedanstående problem behöver man använda sig av fält för att få en elegant lösning? I vilka problem är det onödigt/olämpligt att använda fält?

- a) läsa in och beräkna summan av 20 heltal
- b) läsa in 50 heltal och skriva ut dem sorterade i växande storlek
- c) läsa in 50 heltal och skriva ut det näst största
- d) läsa in och beräkna medianen av ett antal tal
- e) läsa in och beräkna medelvärdet av ett antal tal
- f) läsa in ett antal tal och skriva ut dem igen med undantag av dubletter
- g) läsa in ett antal tal och skriva ut hur många som lästes in före respektive efter det minsta av dem
- h) läsa in ett antal tal och skriva ut dem som är större än medelvärdet av talen

Uppgift 2.

Givet deklarationen

```
int tabell[] = new int[4];
```

Hur många element rymmer fältet? Vilken typ har elementen i fältet? Hur är fältet indexerat?

Uppgift 3.

Nedanstående deklarationssatser resulterar i kompileringsfel. Varför?

- a) **char** vokaler = **new char**[5];
- b) **double**[] m = **double**[10];
- c) **int** odd[] = **new int**{1, 3, 5, 7, 9};

Uppgift 4.

- a) Deklarera en ett fält som innehåller talen 3, 19, 7, 9 och 4.
- b) Deklarera en ett fält som innehåller siffrorna '0' till '9'.
- c) Deklarera ett fält för att lagra 25 objekt av klassen Point.

Uppgift 5.

Rita en bild över hur minnet ser ut efter att var och en av nedanstående deklarationer har gjorts:

- a) **int** [] vekt;
- b) **char**[] chars = {'a', 'b', 'c', 'd'};
- c) **double**[] nr = **new double**[5];
- d) String names = {"Anne", "Ida", "Marie"}

Uppgift 6.

Antag att du i ett program har följande deklarationer

```
int[] a = {3, 5, 8};  
int[] b = {3, 5, 8};
```

Vad blir utskriften när nedanstående programsekvens exekveras? Om utskrifterna skiljer sig åt i de båda fallen, förklara i så fall varför.

- a) **if** (a == b)
 System.out.println("Lika");
else
 System.out.println("Olika");
- b) **if** (Arrays.equals(a, b))
 System.out.println("Lika");
else
 System.out.println("Olika");

Uppgift 7.

Nedanstående deklaration har gjorts

```
int[] arr = {1, 2, 3, 4, 5, 4, 3, 2, 1}
```

Vad skrivs ut av följande kodsegment

- a)

```
int total = 0;  
for (int i = 0; i < arr.length; i = i + 1) {  
    total = total + arr[i];  
}  
System.out.println(total);
```
- b)

```
int total = 0;  
for (int i = 0; i < arr.length; i = i + 2) {  
    total = total + arr[i];  
}  
System.out.println(total);
```
- c)

```
int total = 0;  
for (int i = 1; i < arr.length; i = i + 2) {  
    total = total + arr[i];  
}  
System.out.println(total);
```
- d)

```
int total = 0;  
for (int i = arr.length-1; i > 5; i = i - 1) {  
    total = total + arr[i];  
}  
System.out.println(total);
```
- e)

```
int total = 0;  
for (int i = 0; i < arr.length; i = i + 1) {  
    total = arr[i] - total;  
}  
System.out.println(total);
```

Uppgift 8.

Nedanstående deklaration har gjorts

```
int[] arr = {1, 2, 3, 4, 5, 4, 3, 2, 1}
```

Vad skrivs ut av nedanstående kodsegment

- a)

```
for (int i = 0; i < arr.length; i = i + 1) {  
    arr[i] = arr[i] + i;  
}  
System.out.println(java.util.Arrays.toString(arr));
```
- b)

```
for (int i = 0; i < arr.length - 1; i = i + 1) {  
    arr[i] = arr[i + 1];  
}  
System.out.println(java.util.Arrays.toString(arr));
```
- c)

```
for (int i = 1; i < arr.length; i = i + 1) {  
    arr[i] = arr[i - 1];  
}  
System.out.println(java.util.Arrays.toString(arr));
```
- d)

```
for (int i = arr.length-1; i > 0; i = i - 1) {  
    arr[i] = arr[i] - arr[i - 1];  
}  
System.out.println(java.util.Arrays.toString(arr));
```
- e)

```
for (int i = 0; i < arr.length / 2; i = i + 1) {  
    arr[i] = arr[i + arr.length / 2];  
}  
System.out.println(java.util.Arrays.toString(arr));
```

Uppgift 9.

a) Vad blir utskriften när nedanstående programsekvens exekveras?

```
int[] vekt = {3, 5, 8, 23, 11, 27};
int i = 0;
while (vekt[i] <= vekt[i+1])
    i = i + 1;
System.out.println(i);
```

b) Vad blir utskriften när nedanstående programsekvens exekveras?

```
int[] vekt = {4, 9, 36, 45, 49, 53};
int i = 0;
while (vekt[i] <= vekt[i+1])
    i = i + 1;
System.out.println(i);
```

Uppgift 10.

Vad blir utskriften av följande kodsegment?

```
int[] arr = new int[9];
arr[0] = 1;
arr[1] = -1;
for (int i = 2; i < 9; i = i + 1)
    arr[i] = arr[i - 1] - arr[i - 2];
System.out.println(java.util.Arrays.toString(arr));
```

Uppgift 11.

Nedanstående kodavsnitt resulterar i kompilerings- eller exekveringsfel. Ange vad som är fel i respektive kodavsnitt.

- a) `int[] values = new int[10];`
`for (int k = 1; k <= values.length; k = k + 1) {`
 `values[k] = k * k;`
}
- b) `int[] values;`
`for (int k = 0; k < values.length; k = k + 1) {`
 `values[k] = k * k;`
}
- c) `double[] values = {1.5, 2.0, 3.5, 4.0, 5.5, 4.5, 3.0, 2.5, 1.5};`
`for (int k = 0; i < values.length; k = k + 1) {`
 `values[k] = values[k] + values[k + 1];`
}

Uppgift 12.

Betrakta metoden `mystery` nedan:

```
public static int mystery(int[] arr) {
    int x = 0;
    for (int i = 0; i < arr.length; i = i + 2) {
        x = x + arr[i];
    }
    return x;
} //mystery
```

Vad skrivs ut av nedanstående kodavsnitt?

```
int[] nums = {3, 6, 1, 0, 1, 4, 2};
System.out.println(mystery(nums));
```

Uppgift 13.

Betrakta metoden `fib` nedan:

```
public static int[] fib(int n) {  
    int[] a = new int[n];  
    a[0] = 1;  
    a[1] = 1;  
    for (int i = 2; i < n; i = i + 1) {  
        a[i] = a[i - 1] + a[i - 2];  
    }  
    return a;  
} //fib
```

Vad blir utskriften av följande sats?

```
System.out.println(java.util.Arrays.toString(fib(7)));
```

Uppgift 14.

Betrakta metoden `partialSum` nedan:

```
public static int[] partialSum(int[] arr) {  
    int[] sum = new int[arr.length];  
    for (int j = 0; j < sum.length; j = j + 1)  
        sum[j] = 0;  
    for (int j = 0; j < sum.length; j = j + 1) {  
        for (int i = 0; i <= j; i = i + 1)  
            sum[j] = sum[j] + arr[i];  
    }  
    return sum;  
} //partialSum
```

Vad blir utskriften av följande kodavsnitt?

```
int[] myStuff = {1, 4, 1, 3};  
System.out.println(java.util.Arrays.toString(partialSum(myStuff)));
```

Uppgift 15.

Vad skrivs ut av nedanstående program? Motivera, lämpligen genom att rita händelseförloppet!

```
public class ClassA {  
    private static void m1(int n, int[] a) {  
        n = 2;  
        a[0] = 3;  
    } //m1  
  
    public static void main (String[] args) {  
        int[] v = {1, 2, 3};  
        int x = 5;  
        m1(x,v);  
        System.out.println("x= " + x);  
        System.out.println("v[0] = " + v[0]);  
        System.out.println("v[1] = " + v[1]);  
        System.out.println("v[2] = " + v[2]);  
    } //main  
} //ClassA
```

Uppgift 16.

Vad skrivs ut av nedanstående program? Motivera, lämpligen genom att rita händelseförloppet!

```
public class Dummy {  
    public static void f(int x, int[] y, int[] z) {  
        x = 1;  
        y[0] = 2;  
        z = new int[5];  
        z[0] = 555;  
    }  
    public static void main(String[] args) {  
        int x = 111;  
        int[] y = { 222, 333, 444, 555 };  
        int[] z = { 666, 777, 888, 999 };  
        f(x, y, z);  
        System.out.println(x);  
        System.out.println(y[0]);  
        System.out.println(z[0]);  
    }  
}
```

Uppgift 17.

Antag att du i ett program har följande deklaration

```
int[] tal = new int[100];
```

Skriv ett programsegment som skriv ut elementen i fältet tal i omvänd ordning.

Uppgift 18.

Antag att du i ett program har följande deklaration

```
int[] vekt = new int[50];
```

Skriv ett programsegment som tilldelar de jämna delementen (vekt[0], vekt[2], etc) värdet 1 och de udda delementen värdet 0.

Uppgift 19.

Skriv en metod

```
public static int countOdd(int[] vekt)
```

som som tar ett heltalsfält **vekt** och returnera hur många av elementen i fältet **vekt** som är udda tal.

Exempel:

Antag att följande deklaration har gjorts

```
int[] vekt = { 12, 7, 3, 9, 18, 1, 5, 22, 13 } ;
```

Ett anrop av countOdd(vekt) returnerar värdet 6.

Uppgift 20.

Skriv en metod

```
public static int computeOddSum(int[] vekt)
```

som som tar ett heltalsfält **vekt** och returnera summan av alla element i fältet **vekt** som är udda tal.

Exempel:

Antag att följande deklaration har gjorts

```
int[] vekt = { 12, 7, 3, 9, 18, 1, 5, 22, 13 } ;
```

Ett anrop av computeOddSum(vekt) returnerar värdet $7+3+9+1+5+13 = 38$.

Uppgift 21.

Skriv en metod

```
public static int countCloseTo(double[] values, double target, double tolerance)
```

som tar ett fält **values** av reella tal samt två reella tal **target** och **tolerance** och returnerar antalet element i fältet **values** som inte skiljer sig från värdet av **target** med mer än **tolerance**.

Uppgift 22.

Skriv en metod

```
public static double[] doubleValues(double[] original)
```

som tar ett fält **original** av reella tal och returnerar ett nytt fält som innehåller samma element som i fältet **original** men där elementen har multiplicerats med 2.

Uppgift 23.

Skriv en statisk metod

```
public static boolean reverse(boolean[] original)
```

som tar ett boolskt fält **original** och returnerar ett nytt fält som innehåller samma element som i fältet **original** men där elementen har omvänd ordning, dvs. det första elementet i fältet **original** kommer sist i det nya fältet, det andra elementet i fältet **original** kommer näst sist i det nya fältet, osv.

Uppgift 24.

Skriv en metod

```
public static double[] removeFirst(double[] original)
```

som tar ett fält **original** av reella tal och returnerar ett nytt fält som innehåller samma element som i fältet **original** förutom att det första elementet är borttaget.

Uppgift 25.

Skriv en metod

```
public static int findFirstTarget(int[] values, int target)
```

som tar ett heltals fält **values** samt ett heltal **target** och returnerar index för första förekomsten av talet **target** i fältet **values**. Om ingen förekomst av **target** finns i **values** skall värdet -1 returneras.

Uppgift 26.

Skriv en metod

```
public static int[] removeAllTargets(int[] original, int target)
```

som tar ett heltalsfält **original** samt ett heltal **target** och returnerar ett nytt fält som innehåller samma element som **original** förutom att alla förekomster av **target** är borttagna. Använd metoden **removeAt** från föregående uppgift.

Uppgift 27.

Antag att följande deklarationer gjorts

```
String str = "The three did feed the deer";
```

Vad blir resultatet av nedanstående uttryck?

- | | | |
|-----------------------|-------------------------|--------------------------|
| a) str.length() | b) str.charAt(5) | c) str.indexOf('e') |
| d) str.indexOf("did") | e) str.substring(4,9) | f) str.toUpperCase() |
| g) str.indexOf("two") | h) str.lastIndexOf('e') | i) str.replace('e', 'a') |

Uppgift 28.

Antag att följande deklarationer gjorts

```
String s1 = "Harry";
```

```
String s2 = " Arne";
```

```
String s3 = "Sofie";
```

Vad blir resultatet av nedanstående uttryck?

- | | | |
|---------------------|---------------------|--------------------------|
| a) s1.compareTo(s2) | b) s2.compareTo(s3) | c) s1.compareTo("Harry") |
|---------------------|---------------------|--------------------------|

Uppgift 29.

Antag att följande deklarationer gjorts

```
String s = "a good time";
```

```
String s1 = "time";
```

```
String s2 = s.substring(7, 11);
```

```
String s3 = s2;
```

Vad blir resultatet av nedanstående uttryck?

- | | | |
|------------------|----------------------|--------------------------------|
| a) s1 == s2 | b) s1 == s3 | c) s2 == s3 |
| d) s1.equals(s2) | e) s1.equals("TIME") | d) s1.equalsIgnoreCase("TIME") |

Uppgift 30.

Vad blir utskriften av följande programsegment?

```
String s1 = "fabrik";
```

```
String s2 = "snus" + s1;
```

```
s1 = s1.concat("ör");
```

```
System.out.println(s2);
```

```
System.out.println(s1);
```

Uppgift 31.

Betrakta nedanstående metod:

```
public static String mystery(String str) {  
    String output = "";  
    for (int j = 1; j < str.length(); j = j + 2) {  
        output = output + str.charAt(j);  
    }  
    return output;  
} // mystery
```

Vad blir utskriften av nedanstående sats?

```
System.out.println(mystery("computer"));
```

Uppgift 32.

Betrakta metoden nedan:

```
public static String scramble(String word, int howFar) {  
    return word.substring(howFar + 1, word.length())  
        + word.substring(0, howFar);  
}  
//scramble
```

Vad blir utskriften av satsen

```
System.out.println(scramble("computer", 3);
```

Uppgift 33.

Antag att följande deklaration har gjorts

```
String str = "Detta ÄR en STRÄNG med BÅDE små Och STORA bOkStÄvEr!";
```

skriv ett kodavsnitt som översätter alla små bokstäver i variabeln `str` till stora bokstäver och alla stora bokstäver till små bokstäver.

Uppgift 34.

Vad blir utskriften när `main`-metoden nedanstående klass exekveras?

```
public class AClass {  
    public static void changer(String x, int y) {  
        x = x + "peace";  
        y = y * 2;  
    }  
    //changer  
  
    public static void main(String[] args) {  
        String s = "world";  
        int n = 6;  
        System.out.println(s + " " + n);  
        changer(s, n);  
        System.out.println(s + " " + n);  
    }  
    //main  
}  
//AClass
```

Uppgift 35.

Ett företag använder email-adresser enligt följande två former:

```
John.Doe@BestBuy.com  
John_Doe@BestBuy.com
```

Företagen vill nu byta email-adresserna till formen:

```
Doe.J@BestBuy.net
```

Din uppgift är att skriva en metod

```
public static String fixEmail(String oldEmail)
```

som gör detta. Du får anta att parametern `oldEmail` är en sträng på formen

```
<first-name> . <last-name> @ <company-name> . com  
<first-name> _ <last-name> @ <company-name> . com
```

Uppgift 36.

Skriv en metod

```
public static String middle(String str)
```

som tar en sträng `str` och returnera en ny sträng som innehåller det mittersta tecknet i strängen `str` om `str` har ett udda antal tecken och de två mittersta tecknen om `str` har ett jämt antal tecken. Till exempel skall anropet `middle("middle")` returnera strängen "dd".