

LektionsProblem

Joachim von Hacht

1

Loopar

Skriv, för varje nedan, en while och en for-loop som ger utskriften (samma utskrift för båda loopar:na).

- a) -20, -19, ..., 3
- b) 10, 9, ..., -10
- c) 5, 10, ..., 95, 100
- d) 0.0, 0.3, ..., 8.7, 9.0
- e) $1/2, 1/4, 1/8, \dots, 1/256$ (inte 0.5, 0.25, ...)

Skissa Programmet

```
Input an even integer (0 to quit) > 2
Ok! That was even.
Input an even integer (0 to quit) > 1
That wasn't an even number.
Input an even integer (0 to quit) > 3
That wasn't an even number.
Input an even integer (0 to quit) > 0
You got 1 correct out of 3
```

3

Skissa Programmet (2)

```
Welcome to Number Guess
Guess a number [1-100] > 50
To small
Guess a number [1-100] > 75
To small
...
Guess a number [1-100] > 93
To big
Guess a number [1-100] > 92
To big
Guess a number [1-100] > 91
Correct! Number of guesses needed 8
```

4

Vad gör programmen?

Sammanfatta med en mening!

```

1) void program() {
    int a;
    int b;
    int c;

    a = sc.nextInt();
    b = sc.nextInt();
    c = sc.nextInt();

    int m;
    if (a > b) {
        if (a > c) {
            m = a;
        } else {
            m = c;
        }
    } else {
        if (b > c) {
            m = b;
        } else {
            m = c;
        }
    }
    out.println(m);
}

2) void program() {
    int a;
    int b;
    int c;

    a = sc.nextInt();
    b = sc.nextInt();
    c = sc.nextInt();

    int m = a;
    if (b > m) {
        m = b;
    }
    if (c > m) {
        m = c;
    }
    out.println(m);
}

3) void program() {
    int a;
    int b;
    int c;

    a = sc.nextInt();
    b = sc.nextInt();
    c = sc.nextInt();
    if (b > m) {
        m = b;
    }
    a = sc.nextInt();
    if (a > m) {
        m = a;
    }
    out.println(m);
}

4) void program() {
    int a;
    int b;
    int c;
    int m = 0;

    a = sc.nextInt();
    b = sc.nextInt();
    c = sc.nextInt();

    if (a > b && a > c) {
        m = a;
    } else if (b > a && b > c) {
        m = b;
    } else if (c > a && c > b) {
        m = c;
    }
    out.println(m);
}

5) void program() {
    int is[] = {0, 0, 0};
    int m = 0;

    for (int i = 0; i < 3; i++) {
        is[i] = sc.nextInt();
        if (is[i] > is[m]) {
            m = i;
        }
    }
    out.println(is[m]);
}
    
```

Vad gör metoderna?

Sammanfatta med en mening! Är någon "bättre/sämrre"?

```

1) int doIt(int[] a, int b) {
    int f = -1;
    int i = 0;
    while (i < a.length) {
        if (a[i] == b) {
            f = i;
            break;
        }
        i++;
    }
    return f;
}

2) int doIt(int[] a, int b) {
    int f = -1;
    int i = a.length - 1;
    while (i >= 0) {
        if (a[i] == b) {
            f = i;
            i--;
        }
    }
    return f;
}

3) int doIt(int[] a, int b) {
    int f = -1;
    for (int i = 0; i < a.length; i++) {
        if (a[i] == b) {
            f = i;
        }
    }
    return f;
}
    
```

Vad gör Metoden (1)?

Om du vet, hur kom du fram till detta? Var började du?

```

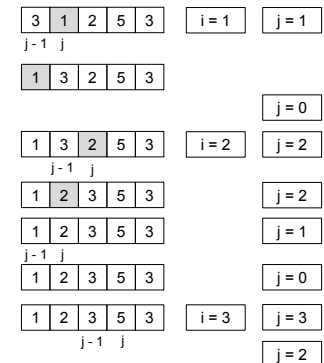
void doIt(int[] arr) {
    for (int i = 1; i < arr.length; i++) {
        int j = i;
        while (j > 0 && arr[j - 1] > arr[j]) {
            int tmp = arr[j - 1];
            arr[j - 1] = arr[j];
            arr[j] = tmp;
            j--;
        }
    }
}
    
```

Analys: Vad gör Metoden (1)?

Analys utifrån in (kör koden papper/penna)

```

for (int i = 1; i < arr.length; i++) {
    int j = i;
    while (j > 0 && arr[j - 1] > arr[j]) {
        int tmp = arr[j - 1];
        arr[j - 1] = arr[j];
        arr[j] = tmp;
        j--;
    }
}
    
```



ETC.

Analys: Vad gör Metoden (1)?

Analys inifrån ut

```
void doIt(int[] arr) {  
    for (int i = 1; i < arr.length; i++) {  
        int j = i;  
        while (j > 0 && arr[j - 1] > arr[j]) {  
            int tmp = arr[j - 1];  
            arr[j - 1] = arr[j];  
            arr[j] = tmp;  
            j--;  
        }  
    }  
}
```

3) Upprepa om "vi är i" arrayen (j > 0) och värdet till vänster är större

1) Byter plats på värdet i j-1 och j

2) Flyttar vänster

4) Upprepa för alla element (utom första)

Vad gör Metoden (2)?

Analysera genom att "köra" med papper och penna!

```
void doIt(int[][] arr) {  
    for (int row = 0; row < arr.length; row++) {  
        for (int col = row + 1; col < arr[row].length; col++) {  
            int tmp = arr[row][col];  
            arr[row][col] = arr[col][row];  
            arr[col][row] = tmp;  
        }  
    }  
}
```

Vad gör Metoden (3)?

Analysera genom att "köra" med papper och penna (arr är sorterad i stigande ordning).

```
int doIt(int[] arr, int value) {  
    int lo = 0;  
    int hi = arr.length - 1;  
    while (lo <= hi) {  
        int m = lo + (hi - lo) / 2;  
        if (value < arr[m]) {  
            hi = m - 1;  
        } else if (value > arr[m]) {  
            lo = m + 1;  
        } else {  
            return m;  
        }  
    }  
    return -1;  
}
```

Analys: Vad gör Metoden (3)?

```
int doIt(int[] arr, int value) {  
    int lo = 0;  
    int hi = arr.length - 1;  
    while (lo <= hi) {  
        int m = lo + (hi - lo) / 2;  
        if (value < arr[m]) {  
            hi = m - 1;  
        } else if (value > arr[m]) {  
            lo = m + 1;  
        } else {  
            return m;  
        }  
    }  
    return -1;  
}
```

loop

arr	value	lo	hi	m	arr[m]
[1, 4, 5, 7]	3	0	3	$0+(3-0)/2=1$	4
			0	$0+(0-0)/2=0$	1
		1	0	lo > hi!	
				return -1	

arr	value	lo	hi	m	arr[m]
[1, 4, 5, 7]	5	0	3	$0+(3-0)/2=1$	4
		2	3	$2+(3-2)/2=2$	5
				return 2	

Utfifrån in med penna och papper

Harmoniska Serien

$$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} \dots + \frac{1}{n}$$

Vilket är minsta steget?

- 1) Skriv en metod som summerar n termer ur serien (namn? indata? utdata?).
- 2) Skriv en metod som returnerar max antal termer som kan summeras utan att överskrida ett visst gränsvärde (namn? indata? utdata?).

13

Vilken är idén ... (1)

Implementera en metod som kan derivera ett polynom!

$$f(x) = x^4 - 5x^2 + 12x - 3$$

$$f'(x) = 4x^3 - 10x + 12$$

$$f''(x) = 12x^2 - 10$$

$$f'''(x) = 24x$$

$$f^{(4)}(x) = 24$$

$$f^{(5)}(x) = 0$$

$$f^{(6)}(x) = 0$$

$$\text{etc.} = 0$$

$$\text{etc.} = 0$$

Vilken är idén ... (2)

Implementera en metod som avgör om en mening är ett palindrom

Några exempel på palindrom;

"Sirap i paris" Läs->"sirapiparis"<-läs

"Madam I'm Adam"

"Was It A Rat I Saw?"

"Can a get a hot dog?"

Vilken är idén ... (3)

Skriv en metod som returnerar frekvenserna för ett stort antal värden i intervallet [0, n].

Class interval	Tally	Frequency
0 - 39	I	1
40 - 79		5
80 - 119		12
120 - 159		8
160 - 199		4
200 - 239	I	1
Sum =		31

När ...???

1. När i programmet kan implicita typomvandlingar förekomma?
2. När i programmet kan kopiering av data (värden) förekomma?

Sidoeffekter...???

När och hur kan sidoeffekter uppstå i ett program?

Hur (1)...???

Hur (på vilket sätt) kan man i ett program skapa objekt?

Räkna upp alla sätt du känner till!

Namn...???

Vilka "saker" kan ha samma namn i ett program?

Hur många...?

Hur många objekt finns i koden? Hur kommer vi åt dessa?

```
void program() {
    new Random().nextInt(10);
    int s = sum(new int[]{1,2,3});
    for( char ch : "abc".toCharArray()){
        // ...
    }
    out.println("123" + "abc");
}

int sum( int[] arr){    // Hmm, sum, same name as ...
    int sum = 0;        // ... here ?!
    for(int i = 0 ; i < arr.length; i++){
        sum += i;
    }
    return sum;
}
```

Vad händer ...(1)???

Vad händer på respektive rad? Vad skrivs ut på rader med out.println(..). Vilka rader kompilerar inte?

```
Object o = new int[3];           // 1
out.println(o[0]);               // 2
int[] i = o;                     // 3
int[] j = (int[]) o;             // 4
out.println(j[0]);               // 5
out.println(o == new int[3]);    // 6
out.println(o.equals(new int[3])); // 7
out.println(new int[3].equals(o)); // 8

String s1 = o;                   // 9
String s2 = (String) o;          // 10
```

Vad händer...(2)?

Förklara (i detalj)!

```
int x;
int y;
x = y = 1;

out.println(x, y);
```

Före efter...(1)

```
int[] a;           // Före anrop
a = getArr(3, 5);  // Anrop
out.println(a);    // Efter anrop
```

```
int[] getArr( int size, int value ){
    int[] a = new int[size];
    for( int i = 0 ; i < a.length ; i++){
        a[i] = value;
    }
    return a;
}
```

Rita bild med
variabler,
referenser och
objekt. Före och
efter anrop

Före efter... (2)

```
A a = new A();
a.arr = new int[]{1, 2, 3};
a.d = 1.5;
A b = new A();
b.arr = new int[]{4,5,6};
b.d = 2.5;           // Before

a = doIt( b);         // Call
                        // After
```

Rita bild med
variabler,
referenser och
objekt. Före och
efter anrop

```
A doIt( A a ){
    A tmp = new A();
    tmp.arr = a.arr;
    tmp.d = a.d;
    return tmp;
}
```

```
class A {
    int[] arr;
    double d;
}
```

Före efter... (3)

```
A[] as = new A[3];
A a1 = new A();
a1.i = 1;
a1.s = "aaa";

A a2 = new A();
a2.i = 2;
a2.s = "bbb";

as[0] = a1;
as[1] = a2;           // Before
as[2] = doIt(as);     // Call
                        // After
```

Rita bild med
variabler,
referenser och
objekt. Före och
efter anrop

```
A doIt( A[] as ){
    as[0].s = as[1].s;
    return as[0];
}
```

```
class A {
    int i;
    String s;
}
```

Före efter... (4)

```
B b1 = new B("b");
A a1 = new A( b1 ); // Before
B b2 = doIt( a1 );  // Call
                        // After
```

Rita bild med
variabler,
referenser och
objekt. Före och
efter anrop

```
B doIt( A a ){
    a.b.s = "c";
    return a.b;
}
```

```
class A {
    int i;
    B b;
    A( B b ){
        this.b = b;
    }
}
```

```
class B {
    int i;
    String s;
    B ( String s ){
        this.s = s;
    }
}
```

(Lätt) Mikrotenta (1)

1. Vad betyder tilldelning resp. synlighetsområde? 2p
Förklar kort och koncist (får rita, skriva kod)
2. Kompilerar följande (om ej varför)? 2p

```
import static java.lang.Math.*;
int[] arr = new int[ sqrt(9) ]; ?
```
3. Skriv metod sum(m, n) som summerar alla heltal i 2p
intervallet [m, n]
4. Skriv en metod som vänder på en heltals-array. Exempel 2p

```
[ 1,2,3 ] -> [ 3,2,1 ]
```


Original arrayen får inte ändras
5. Skriv en metod som returnerar det största värdet i en 2p
kvadratisk flyttalsmatris (double).
6. Rita en bild med minst 2 av varje: variabler, referenser och 2p
objekt. Skriv kod som åstadkommer det du ritat.

Mikrotenta (2)

1. Vad betyder värdeanrop resp. vad är en uppräkningsstyp? 2p
Förklar kort och koncist (får rita, skriva kod)
2. Kompilerar följande (om ej varför)? 2p

```
int i = 0; double j = 1.5;  
out.println(2 * i + (j > 0));
```
3. Skriv metod fac(n) som beräknar n! ($n! = n*(n-1)*(n-2)...*1$) 2p
4. Skriv en metod som returnerar alla jämna tal i en heltalsarray. 2p
Exempel

```
[1, 2, 3, 6, 8, 11] -> [2, 6, 8]
```
5. Skriv metod swap(int[][] m, int r1, int r2) som byter plats på rad r1 och r2 i matrisen m. 2p
6. Skriv en metod som raderar ett viss tecken ur en sträng (alla strängmetoder du kan är tillåtna). Exempel 2p

Sträng	Att ta bort	Resultat
"abeabdbdabra"	'a'	"bebdbdbr"

Varför använder vi ...

1. (primitiva/ref) Typer?
2. (konstanta) Variabler?
3. Tilldelning, inkrementering/dekrementering?
4. Synlighetsområden?
5. Uttryck?
6. (styrande) Satser?
7. Arrayer?
8. Referenser?
9. Metoder (parametrar, returtyp)?
10. Klasser?
11. Funktionell abstraktion?
12. Funktionell nedbrytning?
13. Testning?
14. Strängar?
15. Samlingar?

30

Ordlistan t.o.m. vecka 5

API, Kompilering, Exekvering, (Lat) Evaluering, Sats, Literal, Uttryck, Deklaration, Lokal/Instans variabel, Initiering, Tilldelning, Operator, Synlighetsområde, Överlagrad, Värde/Referens-likhet, Sidoeffekt, Kommentar, Nästlade, Block, Indentering, Nyckelord, (Primitiv/Referens) Typ, Typfel, Implicit/explicit typomvandling, Syntaxfel, Element, Index, Indexera, Punktnotation, Datastruktur, Traversera, Instansiering, null, Metod (huvud, kropp, parameterlista, returtyp), Argument, Värdeanrop, Anropsstack, void, Utparameter, Selektion, Iteration, Redundant/Duplicerad kod, Terminera, Off-by-one fel, Uppräkningsstyp, Omslagstyp, Boxing/unboxing, Gränssnitt, Skräpsamling, Alisas-problem, Samlingar, NullPointerException (NPE), ...

31

Ordlistan vecka 6-

Klass, objekt, instans, instansmetoder, this, (överlagrad) konstruktor, mutera, informationsgömning, åtkomst, get/set-metoder, icke-muterbar, klassen Object, överskuggning (override), annotering, klassmetoder, klassvariabler, konstanter, main-metoden, statisk/deklareras typ, dynamisk/objekttyp, kompatibla referenstyper, implementera (gränssnitt)

32

En enda instans!

Skapa en klass som det bara finns (kan finnas) en enda instans av (kallas Singleton klass)

