

Instuderingsfrågor - del 1

identifierare, primära datatyper, tilldelningssatsen,
klassen Math, konstanter

Uppgift 1.

Vilka av följande identifierare är ogiltiga i Java? Motivera.

- | | | |
|----------------|--------------|----------------|
| a) Benny | b) any.time | c) any_time |
| d) intNumber | e) INTNUMBER | f) 1stStreet |
| g) firstStreet | h) now&then | i) nowANDthen |
| j) float | k) tooHot? | l) final score |

Uppgift 2.

Antag att följande deklarationer har gjorts

```
int x = 2, y = 3, z = 4, w = 5;
```

Vad blir resultatet av nedanstående uttryck?

- | | | |
|--------------------|-------------------|--------------------|
| a) $x + w / 2$ | b) $z * 4 - y$ | c) $y + w \% 2$ |
| d) $x + y - z$ | e) $x * z / y$ | f) $x + z * y / w$ |
| g) $y * x - z / y$ | h) $w * x \% - 4$ | i) $14 \% w \% y$ |

Uppgift 3.

Antag att följande deklarationer gjorts:

```
int myNumber = 17;
```

```
int yourNumber = 12;
```

Vad blir utskriften av följande satser:

- a) System.out.println("Antalet är " + myNumber + " styck.");
- b) System.out.println("Antalet är + myNumber + styck.");
- c) System.out.println("Antalet är " + myNumber + yourNumber + " styck.");
- d) System.out.println("Antalet är " + (myNumber + yourNumber) + " styck.");

Uppgift 4.

Antag att följande deklarationer har gjorts

```
int a = 4, b = 7;
```

```
double c = 5.1, d = 6.4;
```

Vad blir resultatet av nedanstående uttryck?

- | | | |
|------------|-------------|----------------|
| a) b / a | b) $b \% a$ | c) $c + a / b$ |
| d) $a * b$ | e) c / a | f) $b + d$ |
| g) $a * d$ | h) $c \% a$ | i) $d \% c$ |

Uppgift 5.

Vilken datatyp har följande literaler?

- | | | |
|---------|-----------|---------|
| a) 44 | b) 65.98 | c) "42" |
| d) true | e) "true" | f) '6' |

Uppgift 6.

Vilka av följande uttryck är ekvivalenta (om något)?

- | | | |
|------------------|----------------|------------------|
| a) $(x + y) * z$ | b) $x + y * z$ | c) $x + (y * z)$ |
|------------------|----------------|------------------|

Uppgift 7.

Nedanstående deklarationssatser innehåller ett antal felaktigheter. Vilka?

```
int a, b;  
int c = 3, d = 4.5;  
double x, a;  
char teck = "y";  
String str = 'en text';
```

Uppgift 8.

Tag bort alla överflödiga parenteser i tilldelningssatserna nedan:

- a) alt = alt + (base + col4) + sum;
- b) pressure = (temp + entropy) * spec2;
- c) gradient = (gradient - (hgt - slope));
- d) eff = (eff + full * (Math.exp(h3 * Math.log(loss))));
- e) x = -b + Math.sqrt((b * b - (4 * (a * c))));

Uppgift 9.

Antag att följande deklarationer har gjorts

```
int i, j, k;  
double x, y;  
boolean a, b;
```

Vilka av nedanstående tilldelningssatser är korrekta och vilka är felaktiga?

- | | | |
|-------------------|---------------|---------------|
| a) x = y; | b) i = k - j; | c) i + j = k; |
| d) a = a; | e) x = 3 + y; | f) b = x < y; |
| g) b = x < y + 1; | h) y = a + b; | i) i = x + y; |

Uppgift 10.

Vad innebär *implicit typomvandling*? Vad innebär *explicit typomvandling*? När måste explicit typomvandling göras och hur gör man detta?

Uppgift 11.

Antag att vi gjort deklarationen

```
final int MAX_SIZE = 50;  
int x;
```

vilka av nedanstående satser resulterar i ett kompileringsfel?

- a) System.out.println(MAX_SIZE);
- b) MAX_SIZE = 50;
- c) x = MAX_SIZE + 2;

Uppgift 12.

Evaluera följande uttryck. Ange datatypen på resultatet.

- | | |
|----------------------------|--|
| a) Math.pow(9, 2) | b) Math.sqrt(9 + 7) |
| c) 3 * Math.round(6.7) + 5 | d) Math.sqrt(Math.pow(5, 2) + Math.pow(12, 2)) |

Uppgift 13.

Skriv följande formler som uttryck i Java:

- | | |
|------------------------------|--|
| a) $\frac{1}{2\pi\sqrt{LC}}$ | b) $\frac{C \cdot h}{(h^2 + d^2)^3}$ |
| c) $\sqrt{\sin x^2 + y^2}$ | d) $\frac{\sqrt{(a+b)(a-b)}}{a^2 + \frac{1}{a^3 - 2}}$ |

Uppgift 14.

Nedanstående program ger ett kompileringsfel i stil med

```
Honest.java: 4: cannot resolve symbol  
symbol : variable JOptionPane
```

Vad är fel?

```
public class Honest {  
    public static void main (String[] arg) {  
        JOptionPane.showMessageDialog(null, "I hate Java!");  
    } //main  
} // Honest
```

Uppgift 15.

Givet följande deklarationer:

```
final int SIZE = 100;  
int index, weight, shoeSize, shoulderBreadth;  
double factor;  
char ch1, ch2;  
boolean start, finished, flag;
```

Skriv tilldelningssatser för att utföra följande:

- Öka `index` med 1.
- Sätter `factor` till summan av `shoeSize` och `shoulderBreadth`.
- Sätter `factor` till roten ur `factor` dividerat med `weight`.
- Sätter `start` till värdet `true`.
- Sätter `flag` till sanningsvärdet av att `SIZE` är större än `index`.
- Sätter `finished` till sanningsvärdet av att `ch1` är lika med `ch2` eller lika med ett av tecknen 'n' eller 'N'.

Uppgift 16.

Givet en heltalsvariabel `date`, som har tilldelats ett datum på formen `yyymmdd` (ex 830803). Skriv tilldelningssatser som tilldelar heltalsvariablerna `year`, `month` och `day` värdet av `yy`, `mm` respektive `dd`.

Uppgift 17.

Antag att följande deklarationer har gjorts:

```
int i, j, k;  
double x;
```

Vilka av nedanstående satser är felaktiga? Ange orsaken!

- | | | |
|----------------------------|----------------------------|---|
| a) <code>i = j / k;</code> | b) <code>i = x / k;</code> | c) <code>i = Math.pow(k, 2);</code> |
| d) <code>x = j / k;</code> | e) <code>x = j + k;</code> | f) <code>i = (int) Math.pow(x, k);</code> |

Uppgift 18.

Variabeln `size` är av datatypen `double`. Hur skriver man ut dess värde med 3 rätt avrundade decimaler.

Uppgift 19.

Antag att följande deklarationer gjorts

```
int tim = 3;  
int min = 9;  
int sek = 7;
```

som används för att lagra ett antal timmar, minuter respektive sekunder. Hur skriver man om man vill använda dessa variabler för att få en tidsangivelse på formen 03:09:07?

Uppgift 20.

I ett program vill man läsa in tre olika värden från en och samma dialogruta och lagra dessa i heltalsvariabeln `number`, den reella variabeln `freq` respektive heltalsvariabeln `count`. Hur gör man detta?

Uppgift 21.

Varför skall man indentera sina program?

Uppgift 22.

Vad är en algoritm?

Uppgift 23.

Vilka tre enkla styrkonstruktioner kan användas för att uttrycka så gott som alla algoritmer?

Uppgift 24.

Vilka olika faser kan programmeringsarbetet indelas i?

Instuderingsfrågor - del 2

villkor, if-satsen

Uppgift 25.

Skriv Java-kod för uttrycken nedan:

- a) $x > y > z$
- b) x och y är båda mindre än 0
- c) varken x eller y är mindre än 0
- d) x är lika med y men inte lika med z

Uppgift 26.

Vilket/vilka av följande villkorsuttryck testar inte att x är ett tal som ligger mellan 1 och 10?

- a) $1 \leq x \ \&\& \ x \leq 10$
- b) $!(x < 1 \ || \ 10 < x)$
- c) $!(x \leq 1 \ || \ x \geq 10)$
- d) $10 \geq x \ \&\& \ x \geq 1$

Uppgift 27.

Antag att följande deklarationer har gjorts:

```
int x = 4, y = 3, z = 2;
```

Vad skrivs ut av nedanstående sats?

```
System.out.println( !(z == x) || (x > 2) && (y > 3) );
```

Uppgift 28.

Tilldela den logiska variabeln delbar sanningsvärdet av att heltalsvariabeln tal är jämnt delbart med 7.

Uppgift 29.

Skriv Java-satser för följande villkor:

- a) x är större än 3.
- b) y ligger mellan 2 och 5.
- c) r har ett negativt värde och z har ett positivt värde (ett positivt värde är större eller lika med 0).
- d) a och b har samma tecken, dvs båda har negativa värden eller båda har positiva värden.
- e) p och q har samma värde och r har ett annat värde än p och q .

Uppgift 30.

Antag att följande deklarationer har gjorts

```
double x = 2;  
double y = 4;
```

Vad får följande villkorsuttryck för värden?

- a) $x \neq y$
- b) $x < x$
- c) $x \geq (y - x)$
- d) $x == (y + x - y)$

Uppgift 31.

Antag att följande deklarationer har gjorts

```
int x = 2;  
int y = 4;  
int z = 15;
```

Vad blir resultatet av följande uttryck?

- a) $x + 2 < y$
- b) $z - 3 * x \neq y + 5$
- c) $x * y == z - 9$
- d) $z \geq 3 * y$

Uppgift 32.

Antag att följande deklarationer har gjorts

```
int x = 6;  
int y = -2;  
int z = 13;
```

Vad blir resultatet av följande uttryck?

- a) $x - 6 < y \ \&\& \ z == 2 * x + 1$
- b) $!(x - 6 < y \ \&\& \ z == 2 * x + 1)$
- c) $x - 6 < y \ || \ z == 2 * x + 1$
- d) $!(x - 6 < y \ || \ z == 2 * x + 1)$

Uppgift 33.

Skriv ett villkor som avgör huruvida variabeln `number` har ett värde mellan 80 och 90.

Uppgift 34.

Vi har två variabler `weight1` och `weight2`, av datatypen **double**, som representerar vikter av något slag. Vi vill i ett program testa om dessa vikter är lika. Skriv nödvändiga villkorsuttryck för detta om variablerna representerar

- a) vikten på två personer angiven i kilo
- b) vikten på två kanariefåglar angiven i kilo (typiskt väger en kanariefågel ca 0.03 kilo)
- c) vikten på två grävmaskiner angiven i kilo (typiskt väger en en grävmaskin 10000 kilo)

Tips: Reella tal skall inte jämföras på exakthet utan på tillräcklig noggrannhet.

Uppgift 35.

Skriv om nedanstående villkorsuttryck så de blir enklare och tydligare att förstå.

- a) $!(value < 1.0)$
- b) $!(value \geq 0.0 \ \&\& \ value < 1.0)$
- c) $!(smallNumber == 0 \ || \ bigNumber == 10000)$
- d) $!(!(value > 1.0))$

Uppgift 36.

Visa att följande tre uttryck är ekvivalenta.

- a) $x < 10$
- b) $(x < 10) == \text{true}$
- c) $(x \geq 10) == \text{false}$

Vilket av uttrycken tycker du är enklast?

Uppgift 37.

Vad skrivs ut i följande programsekvenser?

- a)

```
if (12 < 12)  
    System.out.println("Never");  
else  
    System.out.println("Always");
```
- b)

```
double value1 = 15.0;  
double value2 = 25.12;  
if (2*value1 > value2)  
    System.out.println("OK");  
else  
    System.out.println("Not OK");
```

Uppgift 38.

Vad skrivs ut i följande programsekvenser?

```
double x = -1.0, y = 4.0, z = 10.0;
if (x > 0)
    if (y > 0)
        z = Math.sqrt(x) + Math.sqrt(y);
else
    z = 0;
System.out.println("z = " + z);
```

Uppgift 39.

Nedanstående programkod ger ett kompileringsfel i stil med

TP.java:7: 'else' without 'if'

Vad är fel?

```
if (x == 0) ;
    x = 100;
else
    x = x + 50;
```

Uppgift 40.

Antag att följande deklARATIONER gjorts

```
boolean x, test;
```

Visa att nedanstående två programsegment är ekvivalenta.

a) if (test)	b) $x = !\text{test};$
$x = \text{false};$	
else	
$x = \text{true};$	

Uppgift 41.

Antag att följande deklARATIONER gjorts

```
double x, y = 15.0;
```

Vilka värden har variabeln x efter att var och ett av nedanstående programsegmenten genomlöpts?

a) $x = 25.0;$	b) if ($y < 15.0$)
if ($y \neq x - 10.0$)	$x = 5.0 * y;$
$x = x - 10.0;$	else
else	$x = 2.0 * y;$
$x = x / 2.0;$	

Uppgift 42.

Vad menas med att en **if**-sats är nästlad?

Uppgift 43.

Skriv om nedanstående nästlade **if**-satser till enkla **if**-satser genom att använda sammansatta villkorsuttryck.

a) if ($a > c$)	b) if ($a == b$)
if ($b > c$)	$x = y;$
$x = y;$	else
else	if ($a > c$)
$x = z;$	$x = y;$
else	else
$x = z;$	$x = z;$

Uppgift 44.

Förklara skillnaden mellan de båda programsegmenten nedan. Vad har **x** för värde efter att respektive programsegmenten genomlöpts för de fall då **x** har värdet -1 respektive **x** har värdet 1 innan sekvenserna genomlöps?

- | | |
|--|---|
| a) if (x >= 0)
x = x + 1;
else if (x >= 1)
x = x + 2; | b) if (x >= 0)
x = x + 1;
if (x >= 1)
x = x + 2; |
|--|---|

Uppgift 45.

Skriv om nedanstående **if**-sats till en tilldelningssats:

```
if (answer == 'Q')  
    finished = true;  
else  
    finished = false;
```

Uppgift 46.

Nedanstående programsekvens ger ett kompileringsfel i stil med

```
cannot resolve symbol: variable y  
location: System.out.println(y);
```

Vad är fel?

```
String indata = JOptionPane.showInputDialog("Ange talet: ");  
int x = Integer.parseInt(indata);  
boolean correct = true;  
if (x != 0) {  
    correct = false;  
    int y = x+2;  
}  
System.out.println(y);
```

Uppgift 47.

Skriv om nedanstående nästlade **if**-sats till en icke-nästlad **if**-sats.

```
if (x < 0)  
    if (y < 3)  
        if (z == 6)  
            a = x + y + z;
```

Uppgift 48.

Vilken av följande "algoritmer" för betalning i en affär är felaktig?

- | | |
|---|--|
| a) if (har kontanter)
betala med kontanter;
else
betala med check; | b) if (har kontanter)
betala med kontanter;
betala med check; |
|---|--|

Uppgift 49.

Vilka av följande boolska-uttryck är logiskt ekvivalenta med varandra? Antag att **finished** och **dead** är boolska variabler.

- a) (**finished** && !**dead**) || (**dead** && !**finished**)
- b) (**finished** || **dead**) || !(**dead** && **finished**)
- c) **finished** != **dead**

Instuderingsfrågor - del 3

while-satsen, **for**-satsen och **do**-satsen

Uppgift 50.

Vad skrivs ut i nedanstående programsekvenser?

- a)

```
int minutes = 10;
while ( minutes <= 60) {
    System.out.println( minutes + " minutes has passed.");
    minutes = minutes + 20;
}
```
- b)

```
int i = 1;
int formula = 0;
while (i < 10 && formula < 25) {
    formula = formula + i*i;
    i = i + 1;
}
System.out.println(formula);
```
- c)

```
int a = 1, antal = 3;
while (a < 6) {
    System.out.println("Resultat: " + a % antal);
    a = a + 1;
}
```
- d)

```
boolean cont = true;
int a = 1, b = 2;
while (cont) {
    if (a > 3) {
        b = b + 2;
        if ((a >= 6) || (b >= 8))
            cont = false;
    }
    System.out.println("a = " + a + " b = " + b);
    a = a + 1;
}
```
- e)

```
final int LIMIT = 16;
int count = 1;
int sum = 0;
int nextVal = 2;
while (sum < LIMIT) {
    sum = sum + nextVal;
    nextVal = nextVal + 2;
    count = count + 1;
}
System.out.println("Sum is " + sum + "\nCount is " + count);
```

Uppgift 51.

Använd en **while**-sats

- a) för att skriva ut serien 0.25, 0.5, 0.75, . . . , 4.75, 5.0
- b) för att skriva ut serien 1 3 6 10 15 21 28 36
- c) som löper så länge som den boolska variabeln **finished** är **false**
- d) så länge som variabeln **counter** är större än konstanten **LOWER** och mindre än konstanten **UPPER**

Uppgift 52.

Vilka av nedanstående **while**-satser resulterar i en evighetsloop?

- a) **int** i = 1;
 while (i != 25)
 i = i +2;
- b) **double** x = 1.0;
 while (x != 0.1)
 x = x -0.1;
- c) **int** x = 0;
 while (x != 55)
 x = x +2 ;
- d) **int** i = 1, sum = 0;
 while(i < 10 || sum !=15) {
 sum = sum + i;
 i = i +1;
 }
- e) **int** k = 1;
 while (k != 10); {
 System.out.println(k);
 k = k +1;
 }

Uppgift 53.

Betrakta nedanstående programsekvens:

```
int i = 11;  
double sum = 0;  
while (i > 0) {  
  i = i - 1;  
  sum = sum + 1.0 / i;  
}
```

Avsikten är att beräkna serien $1/10 + 1/9 + \dots + 1/2 + 1/1$. Något är dock galet och det uppkommer ett exekveringsfel i stil med

Exception in thread "main" java.lang.ArithmeticException: / by zero

Förklara vad som är fel? Rätta felet!

Uppgift 54.

Vad blir utskrifterna av nedanstående programsegment?

- a) **for** (**int** k = 1; k <= 5; k = k + 1)
 System.out.println(k % 4);
- b) **for** (**int** k = 1; k <= 20; k = k +)
 if ((k % 4) == 0)
 System.out.println(k);
- c) **for** (**int** k = -8; k <= 10; k = k + 4)
 System.out.println(k);
- d) **for** (**int** k = 1; k <= 30; k = k * 2)
 System.out.println(k);
- e) **for** (**int** k = 5; k < 15; k = k + 2)
 System.out.println(k);
- f) **for**(**int** i = 15; i > 5; i = i - 3) {
 System.out.println(i);
}

Uppgift 55.

for-satsen är egentligen en generell villkorsloop. Trots detta bör man enbart använda **for**-satsen som en räkneloop. Varför?

Uppgift 56.

Nedanstående programsegment är ett exempel på missbruk av **for**-satsen. Varför? Hur borde koden se ut?

```
for (int i = 1; i <= 10; i = i + 1) {  
    System.out.println(i);  
    i = i+1;  
}
```

Uppgift 57.

Programsekvenserna i samtliga deluppgifter nedan resulterar i ett kompileringsfel! Varför? Rätta till koden!

- a) **for** (**int** k = 1, k <= 10, k = k + 1)
 System.out.println(k);
- b) **int** i;
 for (**int** i = 1; i <= 10; i = i + 1) {
 //här kommer en eller flera programsatser
 }
- c) **for** (**int** j = 1; j <= 10; i = j + 1); {
 System.out.println(j);
 //här kommer eventuellt flera programsatser
}
- d) **int** antalVarv = 0;
 for (**int** i = 1; i <=10; i = i + 1)
 for (**int** i = 1; i <=10; i = i + 1)
 antalVarv = antalVarv + 1;

Uppgift 58.

Vad blir utskrifterna av nedanstående programsegment?

- a) **int** n = 3;
 int m = 6;
 for (**int** i = 1; i <= n; i = i + 1) {
 for (**int** j = 1; j <= m; j = j + 1)
 System.out.print('*');
 System.out.println();
 }
- b) **int** n = 3;
 int m = 6;
 for (**int** i = 1; i <= n; i = i + 1) {
 for (**int** j = 1; j <= i; j = j + 1)
 System.out.print('*');
 System.out.println();
 }
- c) **for** (**int** i = 0; i <= 3; i = i + 1) {
 for (**int** j = 5; j <= 7; j = j + 1)
 System.out.print(j);
 System.out.println(i);
}

Uppgift 59.

Vad blir utskrifterna av nedanstående programsegment?

- a)

```
for (int j = 1; j <= 5; j = j + 1) {  
    for (int k = 1; k <= 5; k = k + 1) {  
        System.out.print(j + " ");  
    }  
    System.out.println();  
}
```
- b)

```
for (int j = 1; j <= 5; j = j + 1) {  
    for (int k = 1; k <= j; k = k + 1) {  
        System.out.print(j + " ");  
    }  
    System.out.println();  
}
```
- c)

```
for (int j = 1; j <= 5; j = j + 1) {  
    for (int k = j; k <= 5; k = k + 1) {  
        System.out.print(j + " ");  
    }  
    System.out.println();  
}
```

Uppgift 60.

Hur många gånger kommer HURRA! att skrivas ut i nedanstående programavsnitt?

- a)

```
for (int i = 1; i <= 3; i = i + 1)  
    for (int j = 1; j <= 3; j = j + 1)  
        for (int k = 1; k <= 3; k = k + 1)  
            System.out.println("HURRA!");
```
- b)

```
for(int i = 1; i <= 3; i = i + 1)  
    for (int j = 1; j <= 3; j = j + 1)  
        for (int k = 1; k <= j; k = k + 1)  
            System.out.println("HURRA!");
```

Uppgift 61.

Använd en **for**-sats för att

- a) skriva ut alla udda värden i intervallet 1-15
- b) skriva ut serien 0.25, 0.5, 0.75, . . . , 4.75, 5.
- c) skriva ut serien 10, 8, 6, 4, 2, 0, -2, -4, -6, -8, -10
- d) skriva ut serien 0, 1, 4, 9, 16, . . . , 81, 100

Uppgift 62.

I vilka av nedanstående problem är det lämpligast att använda en **for**-sats och i vilka problem är det lämpligast att använda en **while**-sats? Motivera!

- a) skapa en iteration som löper från 15 till -15 med steget 1.
- b) skapa en iteration som håller på tills värdet av variabel **x** till absolutbelopp är mindre än 0.0005.
- c) skapa en iteration som löper från 2 till 100 med steget 2.
- d) skapa en iteration som löper så länge som variabeln **position** är mindre än 100 och den boolska **broken** är **false**

Uppgift 63.

Vad skrivs ut när nedanstående kodsegment exekveras?

```
int i = 4;  
do {  
    System.out.println( i = " + i);  
    i = i - 1;  
} while(i > 4);
```

Uppgift 64.

Betrakta nedanstående programsegment:

```
int i = 0, sum = 1;
do {
    boolean more;
    i = i + 1;
    sum = sum * i;
    if (i > 6)
        more = false;
    else
        more = true;
} while(more);
System.out.println(sum);
```

Vid kompilering får följande felutskrift:

```
cannot find symbol
symbol : variable more
```

Förklara vad som är fel och åtgärda felet.

Uppgift 65.

Skriv en programsekvens som läser in ett heltal och upprepar inläsningen tills ett tal i intervallet 1 till 7 erhålls. Använd dialogrutor för kommunikationen med användaren.

Instuderingsfrågor - del 4

metoder, klasser, objekt, top-down design

Uppgift 66.

Vad innebär *top-down-design*? Vad är fördelen med top-down-design?

Uppgift 67.

Vad innebär det att en metod har resultattypen **void**?

Uppgift 68.

Förklara skillnaden mellan formella och aktuella parametrar.

Uppgift 69.

Sant eller falskt?

- a) En metod har exakt en **return**-sats.
- b) En metod har minst en **return**-sats.
- c) En metod har högst en **return**-sats.
- d) En **void**-metod kan aldrig ha en **return**-sats.
- e) När en **return**-sats exekveras i en metod, kommer metoden omedelbart att terminera.
- f) En metod utan parametrar returnerar alltid samma värde.

Uppgift 70.

Betrakta gränssnittet för nedanstående metoder:

```
public static void methodOne (double one)
public static int methodTwo (int one, double two)
```

I ett program har följande deklarationer gjorts:

```
int a, b;
double c, d;
```

Vilka av nedanstående satser ger upphov ett kompileringsfel? Förklara orsaken till kompileringsfelen!

- a) `d = methodOne(c);`
- b) `methodOne(a);`
- c) `a = methodTwo(c, d);`
- d) `a = methodTwo(a, b);`
- e) `d = methodTwo(b, c);`
- f) `b = methodTwo(a);`

Uppgift 71.

Betrakta nedanstående metod:

```
public static int getResult(int n) {
    int product = 1;
    for (int number = 1; number < n; number = number + 1) {
        if (number % 2 == 0)
            product = product * number;
    }
    return product;
}
```

Vilket värde returneras av anropet `getResult(8)`?

Uppgift 72.

Vad blir utskriften av nedanstående kodavsnitt?

```
public static int mystery(int sum) {  
    int temp = sum;  
    while ( temp > 0) {  
        if ( temp / 10 % 2 == 0)  
            return temp;  
        temp = temp / 10;  
    }  
    return temp;  
} //mystery
```

Vad blir värdet returneras av anropet `mystery(1034)`?

Uppgift 73.

Nedanstående metoder ger exekveringsfel. Varför? Rätta felet!

- a) `public static void methodOne (int one) {
 return one + one;
} //methodOne`
- b) `public static boolean methodTwo (int a, int b) {
 if (a > 2*b)
 return true;
} //methodTwo`

Uppgift 74.

Betrakta metoden nedan:

```
public static boolean mystery(int x) {  
    if (x % 2 == 0) {  
        return true;  
    }  
    else {  
        return false;  
    }  
} //mystery
```

Förklara vad metod gör och ge ett lämpligare namn på metoden än `mystery`.

Uppgift 75.

Betrakta metoden nedan:

```
public static int mystery(int n, int k) {  
    int answer = 1;  
    for (int i = 1; i <= k; i = i + 1)  
        answer = answer * n;  
    return answer;  
} // mystery
```

Förklara vad metod gör och ge ett lämpligare namn på metoden än `mystery`.

Uppgift 76.

Betrakta nedanstående metod:

```
public static int pick(boolean test, int x, int y) {  
    if (test)  
        return x;  
    else  
        return y;  
}
```

Vad blir värdet av anropet

`pick(false, pick(true, 0, 1), pick(true, 6, 7))`

Uppgift 77.

- a) Skriv en metod

int lastDigit(**int** number)

som returnerar den sista siffran i parametern **number**. Till exempel skall anropet lastDigit(62349) returnera värdet 9 och anropet lastDigit(-234) returnera värdet 4.

- b) Skriv en metod

int firstDigit(**int** number)

som returnerar den första siffran i parametern **number**. Till exempel skall anropet firstDigit(62349) returnera värdet 6 och anropet firstDigit(-234) returnera värdet 2.

- c) Skriv en metod

int nrOfDigits(**int** number)

som returnerar antalet siffror i parametern **number**. Till exempel skall anropet nrOfDigits(62349) returnera värdet 5 och anropet nrOfDigits(-234) returnera värdet 3.

- d) Skriv en metod

public static int largestDigit (**int** value)

som tar en heltalsparameter **value** och returnerar ett nytt heltal som utgör av den största siffran som ingår i talet **value**.

Ett anrop largestDigit(2077) skall returnera värdet 7 och ett anrop largestDigit(-932) skall returnera värdet 9.

Uppgift 78.

- a) Skriv en metod som beräknar summan $\sum_{k=i}^n k^2$.

- b) Skriv en metod uttrycket $1 + x + x^2 + x^3 + \dots + x^n$.

Uppgift 79.

Vad är skillnaden mellan en klass och ett objekt?

Uppgift 80.

Vad är skillnaden mellan en instans av en klass och ett objekt av en klass?

Uppgift 81.

Förklara vad en konstruktor är för något. Vad används konstruktorer till?

Uppgift 82.

Måste man deklarera en konstruktor i en klass? Motivera ditt svar!

Uppgift 83.

Vad är attribut?

Uppgift 84.

Vad innebär inkapsling? Vilka är fördelarna med inkapsling?

Uppgift 85.

Förklara vad som menas med var och en av följande variabler i Java:

- a) Enkel variabel
- b) Referensvariabel
- c) Lokal variabel
- d) Instansvariabel
- e) Klassvariabel

Uppgift 86.

Betrakta nedanstående program:

```
public class Strange {  
    int value1 = 2, value2 = 4;  
    public static void main(String[] arg) {  
        System.out.println(2 * value1 + 5 * value2 - 4);  
    } //main  
} //Strange
```

När programmet kompileras fås följande felmeddelande

non-static variable value1 cannot be referenced from a static context

non-static variable value2 cannot be referenced from a static context

Förklara vad som är felaktigt i programmet och ge förslag på hur felet skall rättas till.

Uppgift 87.

Vad är en statisk metod? Varför är `main`-metoden statisk?

Uppgift 88.

Hur sker parameteröverföring i Java?

Uppgift 89.

Vad är en *synlighetsmodifierare*? Redogör för skillnaden mellan synlighetsmodifierarena **private**, **public**, **protected** och utelämnad.

Uppgift 90.

Vad innebär det att två eller flera metoder är överlagrade?

Uppgift 91.

I klassen `ClassA` definieras nedanstående fyra konstruktorer. Men det blir ett kompileringsfel. Varför?

```
public ClassA(int x) { /*code not shown here*/ }  
public ClassA(double z) { /*code not shown here*/ }  
public ClassA(int x, int y) { /*code not shown here*/ }  
public ClassA(int y) { /*code not shown here*/ }
```

Uppgift 92.

Vi har en klass

```
public class ClassC {  
    private int count;  
    public void init() {  
        count = 1;  
    } //init  
    public void increment() {  
        count = count + 1;  
    } //increment  
    public int getCount() {  
        return count;  
    } //getCount  
} //ClassB
```

Vad blir utskriften när klassen `MainClass` nedan exekveras?

```
public class MainClass {  
    public static void main( String[] args) {  
        ClassC theObject = new ClassC();  
        System.out.println(theObject.getCount());  
        theObject.init();  
        theObject.increment();  
        theObject.increment();  
        System.out.println(theObject.getCount());  
    } //main  
} //MainClass
```

Uppgift 93.

Betrakta nedanstående klass

```
public class Dummy {  
    private static String name;  
    public Dummy( String name ) {  
        this.name = name;  
    }//constructor  
    public void print() {  
        System.out.println( "The name is: " + name );  
    }//print  
    public String toString() {  
        return name;  
    } // toString  
} // Dummy
```

Vad blir utskriften när klassen Utskrift nedan exekveras?

```
public class Printout {  
    public static void main( String[] a ) {  
        Dummy kk1 = new Dummy( "First" );  
        kk1.print();  
        Dummy kk2 = new Dummy( "Second" );  
        kk2.print();  
        System.out.println( kk1 + " " + kk2 );  
    }//main  
}//Printout
```

Uppgift 94.

Vi har en klass

```
public class ClassB {  
    public final int A = 365;  
    private int b;  
    public double c;  
    public void m1(int b) {  
        this.b = b;  
    }//m1  
    public void m2() {  
        b = b + 1;  
    }//m2  
    private double m3() {  
        return b*c;  
    }//m3  
}//ClassB
```

I klassen ClassMain nedan används klassen ClassB ovan. Dock innehåller klassen ClassMain ett antal felaktiga satser. Vilka?

```
public class ClassMain {  
    public static void main( String[] args) {  
        ClassB obj = new ClassB();  
        obj.b = 10;  
        obj.c = 5.5;  
        obj.m1(12);  
        obj.m1(7.8);  
        double x = obj.m2();  
        System.out.println(obj.m3());  
    }//main  
}//ClassMain
```

Uppgift 95.

Betrakta nedanstående klass

```
public class ClassD {  
    private int x, y, z;  
    public ClassD(int x, int y, int z) {  
        this.x = x;  
        this.y = y;  
        this.z = z;  
    } //constructor  
    public void modify(int x, int y) {  
        z = x + y;  
        this.x = z;  
        this.y = z + x;  
    } //modify  
    public void printout() {  
        System.out.println("x = " + x);  
        System.out.println("y = " + y);  
        System.out.println("z = " + z);  
    } //printout  
} //ClassD
```

Vad blir utskriften när klassen `MainClass` nedan exekveras?

```
public class MainClass {  
    public static void main( String[] args) {  
        ClassD cd = new ClassD(1, 2, 3);  
        cd.modify(10, 10);  
        cd.printout();  
    } //main  
} //MainClass
```

Uppgift 96.

Betrakta nedanstående program:

```
public class Wrong {  
    public int compute(int a, int b) {  
        return 2*a + 5*b - 4;  
    } //compute  
  
    public static void main(String[] arg) {  
        int tal1 = 2, tal2 = 4;  
        int res = compute(tal1, tal2);  
    } //main  
} //Wrong
```

När programmet kompileras fås följande felmeddelande

`Wrong.java:7: non-static method compute(int,int ) cannot be referenced from a static context`

Förklara vad som är felaktigt i programmet och ge förslag på hur felet skall rättas till.

Instuderingsfrågor - del 5

en-dimensionella fält och klassen String

Uppgift 97.

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 98.

Givet deklarationen

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

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

Uppgift 99.

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 100.

- 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 101.

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 102.

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 103.

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 104.

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 105.

- 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 106.

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 107.

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 108.

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 109.

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 110.

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 111.

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 112.

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;  
    } //f  
    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]);  
    } //main  
} //Dummy
```

Uppgift 113.

Antag att du i ett program har deklarerat och tilldelat elementen i ett fält av typen

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

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

Uppgift 114.

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 115.

Antag att följande klass finns:

```
public class Punkt {  
    private double x, y;  
    public Punkt(double x, double y) {  
        this.x = x;  
        this.y = y;  
    }  
    //Flera metoder  
} //Punkt
```

Vid körning av nedanstående programsegment erhålls felmeddelandet `NullPointerException`. Förklara vad felet beror på. Rätta felet.

```
Punkt[] triangel;  
triangel[0] = new Punkt(3.0, 0.0);  
triangel[1] = new Punkt(0.0, 4.0);  
triangel[2] = new Punkt(0.0, 0.0);
```

Uppgift 116.

Skriv en metod

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

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 117.

Skriv en metod

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

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 118.

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 119.

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 120.

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 121.

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 122.

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 123.

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 124.

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 125.

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 126.

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 127.

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 128.

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 129.

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 130.

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 131.

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 132.

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 133.

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".

Instuderingsfrågor - del 6

fler-dimensionella fält och ArrayList

Uppgift 134.

Vilka av nedanstående deklarationer är korrekta och vilka resulterar i kompileringsfel?

- a) `double[][] matris = new double(5)(4);`
- b) `double[][] temp= new int[3][9];`
- c) `int[][] speed = {{1, 3, 5 }, {7, 9}};`
- d) `char[][] letters = new char[2][];`

Uppgift 135.

Följande deklaration har gjorts

```
char[][] letters = {{ 'a', 'b', 'c', 'd'}, { 'x', 'y', 'z' }};
```

Vilket värde har

- a) `letters[0][1]`
- b) `letters[0][3]`
- c) `letters[1][3]`
- e) `letters[0]`

Uppgift 136.

Antag att du i ett program har följande deklarationer

```
int[][] vekt = {{3, 5, 8}, {2, 4, 9, 6, 3}, {9, 5}, {2, 4, 8, 2}};
```

Vad blir utskriften när nedanstående programsekvens exekveras?

```
System.out.println(vekt.length);
System.out.println(vekt[0].length);
System.out.println(vekt[1].length);
System.out.println(vekt[2].length);
```

Uppgift 137.

Betrakta nedanstående kodsegment

```
int[][] mat = new int[3][4];
for (int row = 0; row < mat.length; row++) {
    for (int col; col < mat[row].length; col++) {
        if (row < col)
            mat[row][col] = 1;
        else if (row == col)
            mat[row][col] = 2;
        else
            mat[row][col] = 3;
    }
}
```

Vilket innehåll har fältet `mat` efter det att kodsegmentet har exekverats?

Uppgift 138.

Vad skrivs ut när nedanstående program exekveras?

```
public class Test {
    public static void main(String[] args) {
        int[][] values = {{3, 4, 5, 1}, {33, 6, 1, 2}};
        int v = values[0][0];
        for (int row = 0; row < values.length; row++)
            for (int col = 0; col < values[row].length; col = col + 1)
                if (v < values[row][col])
                    v = values[row][col];
        System.out.print(v);
    } //main
} //Test
```

Uppgift 139.

Vad skrivs ut när nedanstående program exekveras?

```
public class Test {
    public static void main(String[] args) {
        int[][] values = {{3, 4, 5, 1}, {33, 6, 1, 2}};

        for (int row = 0; row < values.length; row++) {
            java.util.Arrays.sort(values[row]);
            for (int col = 0; col < values[row].length; col = col + 1)
                System.out.print(values[row][col] + " ");
            System.out.println();
        }
    }
} //main
} //Test
```

Uppgift 140.

Vad blir utskriften av nedanstående programsatser?

```
import java.util.Arrays;
public class Test {
    public static void main(String[] args) {
        int[] oldArray = {1, 2, 3, 4, 5, 6, 7, 8, 9};
        int[][] newArray = new int[3][3];
        int row = 0;
        int col = 0;
        for (int index = 0; index < oldArray.length; index = index + 1) {
            newArray[row][col] = oldArray[index];
            row = row + 1;
            if ((row % 3) == 0) {
                col = col + 1;
                row = 0;
            }
        }
        for (int i = 0; i < newArray.length; i = i + 1)
            System.out.println(Arrays.toString(newArray[i]));
    }
} //main
} //Test
```

Uppgift 141.

Vad skrivs ut när nedanstående program exekveras?

```
public class Test {
    public static void main(String[] args) {
        int[][] values = {{3, 4, 5, 1}, {33, 6, 1, 2}};
        for (int row = 0; row < values.length; row++) {
            System.out.println(calculate(values[row]));
        }
    }
} //main

public static int calculate(int[] list) {
    int v = list[0];
    for (int i = 1; i < list.length; i++)
        if (v < list[i])
            v = list[i];
    return v;
} //calculate
} //Test
```

Uppgift 142.

Vad skrivs ut när nedanstående program exekveras?

```
public class Test {
    public static void main(String[] args) {
        int[][] data = {{1, 2}, {3, 4}}, {{5, 6}, {7, 8}};
        System.out.print(calculate(data[0]));
    } //main
    public static int calculate(int[][] m) {
        int v = m[0][0];
        for (int i = 0; i < m.length; i++)
            for (int j = 0; j < m[i].length; j++)
                if (v < m[i][j])
                    v = m[i][j];
        return v;
    } //calculate
} //Test
```

Uppgift 143.

Vad skrivs ut när nedanstående program exekveras? Beskriv i ord vad metoden `calculate` gör!

```
public class Test {
    public static void main(String[] args) {
        int[][] matrix = {{1, 2}, {1, 2, 3, 4, 5}, {1, 2, 3}};
        System.out.println(calculate(matrix));
    }
    public static int calculate(int[][] m) {
        int longest = 0;
        for (int row = 0; row < m.length; row = row + 1) {
            if (m[row].length > longest)
                longest = m[row].length;
        }
        return longest;
    } //calculate
} //Test
```

Uppgift 144.

Vad skrivs ut när nedanstående program exekveras? Beskriv i ord vad metoden `calculate` gör!

```
public class Test {
    public static void main(String[] args) {
        int[][] matrix = {{1, 2}, {1, 2, 3, 4, 5}, {1, 2, 3}};
        System.out.println(calculate(matrix));
    }
    public static int calculate(int[][] m) {
        int longest = 0;
        int index = -1;
        for (int row = 0; row < m.length; row = row + 1) {
            if (m[row].length > longest) {
                longest = m[row].length;
                index = row;
            }
        }
        return index;
    } //calculate
} //Test
```

Uppgift 145.

Vad skrivs ut när nedanstående program exekveras? Beskriv i ord vad metoden `calculate` gör!

```
public class Test {
    public static void main(String[] args) {
        int[][] matrix = {{1, 2}, {1, 2, 3, 4, 5}, {1, 2, 3}};
        System.out.println(java.util.Arrays.toString(calculate(matrix)));
    }
    public static int[] calculate(int[][] m) {
        int longest = 0;
        int index = -1;
        for (int row = 0; row < m.length; row = row + 1) {
            if (m[row].length > longest) {
                longest = m[row].length;
                index = row;
            }
        }
        return java.util.Arrays.copyOf(m[index], m[index].length);
    } //calculate
} //Test
```

Uppgift 146.

Skriv en metod

```
public static boolean exist(int[][] m)
```

som tar ett 2-dimensionellt heltalsfält `m` och returnerar värdet **false** om fältet `m` inte existerar (dvs har värdet **null**) eller innehåller 0 rader, annars returneras värdet **true**.

Uppgift 147.

Skriv en metod

```
public static int largestSumOfRow(int[][] m)
```

som tar ett 2-dimensionellt heltalsfält `m` och returnerar den största radsumman i fältet `m`. Du får anta att fältet `m` existerar, dvs inte är **null**, samt att `m` innehåller minst en rad och att ingen rad är tom, dvs alla rader har minst ett element.

Exempel:

Antag att följande deklaration har gjorts

```
int[][] matrix = {{1, 2, 3, 4, -9}, {2, 3, 1, 3, 1}, {5, 5, 5, 6, 5}};
```

Ett anrop av `largestSumOfRow(matrix)` returnerar värdet 26.

Uppgift 148.

Skriv en metod

```
public static int indexOfLargestRow(int[][] m)
```

som tar ett 2-dimensionellt heltalsfält `m` och returnerar index för den rad i fältet `m` som har störst radsumma.

Exempel:

Antag att följande deklaration har gjorts

```
int[][] matrix = {{1, 2, 3, 4, -9}, {2, 3, 1, 3, 1}, {5, 5, 5, 6, 5}};
```

Ett anrop av `indexOfLargestRow(matrix)` returnerar värdet 2.

Uppgift 149.

Skriv en metod

```
public static int[] getLargestRowt(int[][] m)
```

som tar ett 2-dimensionellt heltalsfält *m* och returnerar en kopia av den rad i fältet *m* som har största radsumma.

Exempel:

Antag att följande deklaration har gjorts

```
int[][] matrix = {{1, 2, 3, 4, -9}, {2, 3, 1, 3, 1}, {5, 5, 5, 6, 5}};
```

Ett anrop av `getLargestRow(matrix)` skall ge fältet `{5, 5, 5, 6, 5}`.

Uppgift 150.

En *enhetsmatrix* eller *identitetsmatrix* är en matrix vars element är 1 i diagonalen och 0 i övrigt. En enhetsmatrix kan ha godtycklig storlek men är alltid kvadratisk, dvs. den har lika många kolumner som rader. Figurerna nedan visar en två enhetsmatriser av storleken 3 respektive 4.

1	0	0
0	1	0
0	0	1

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Du skall skriva en metod

```
public double[][] identityMatrix(int size);
```

som skapar och returnerar en enhetsmatrix av storleken *size*.

Uppgift 151.

Skriv en metod

```
public static int[][] transponat(int[][] mat)
```

som returnerar transponat matrisen av *mat*.

Transponat matrisen A^T till en matrix *A* definieras genom att $a^T_{ij} = a_{ji}$.

Uppgift 152.

Vad är fel med nedanstående kodsegment?

- a) `ArrayList<int> values = new ArrayList<int>();`
- b) `ArrayList<Double> values = new ArrayList();`
- c) `ArrayList<String> values = new ArrayList<String>;`
- d) `ArrayList<Integer> values;`
`for (int i = 1; i <= 10; i = i + 1) {`
 `values.add(i * i);`
`}`
- e) `ArrayList<Integer> values = new ArrayList<Integer>();`
`for (int i = 1; i <= 10; i = i + 1) {`
 `values.set(i - 1, i * i);`
`}`

Uppgift 153.

Betrakta nedanstående metod:

```
public static ArrayList<Integer> mystery(int n) {  
    ArrayList<Integer> list = new ArrayList<Integer>();  
    for (int i = 1; i <= n; i = i + 1) {  
        list.add(new Integer(i * i + 3));  
    }  
    return list;  
}
```

Vad blir utskriften av nedanstående sats?

```
System.out.println(mystery(6));
```

Uppgift 154.

Vad skrivs ut av nedanstående kodavsnitt?

```
List<String> list = new ArrayList<String>();
list.add("A");
list.add("B");
list.add("C");
list.add("D");
for (String str : list) {
    System.out.println(str);
}
```

Uppgift 155.

Vad skrivs ut av nedanstående kodavsnitt?

```
ArrayList<Integer> list1 = new ArrayList<Integer>();
ArrayList<Integer> list2 = new ArrayList<Integer>();
list1.add(1);
list1.add(2);
list1.add(5);
list2.addAll(list1);
list1.addAll(list2);
System.out.println(list1);
```

Uppgift 156.

Vad skrivs ut av nedanstående kodavsnitt?

```
ArrayList<String> list = new ArrayList<String>();
list.add("red");
list.add("green");
list.add("red");
list.add("blue");
list.remove("red");
System.out.println(list);
```

Uppgift 157.

Vad skrivs ut av nedanstående kodavsnitt?

```
ArrayList<String> list = new ArrayList<String>();
list.add("red");
list.add("green");
list.add("red");
list.add("black");
for (int i = 0; i < list.size(); i = i + 1)
    if (list.get(i).equals("red")) {
        list.remove(element);
        i = i - 1;
    }
System.out.println(list);
```

Uppgift 158.

Vad skrivs ut av nedanstående kodavsnitt?

```
ArrayList<String> list = new ArrayList<String>();
list.add("red");
list.add("green");
list.add("red");
list.add("black");
for (int i = 0; i < list.size(); i = i + 1)
    if (list.get(i).equals("red")) {
        list.set(i, "yellow");
        i = i - 1;
    }
System.out.println(list);
```

Uppgift 159.

Skriv en metod

```
public static ArrayList<Integer> append(ArrayList<Integer> a, ArrayList<Integer> b)
```

som tar två listor a och b, och returnerar en ny lista som är en hopslagning av listorna a och b.

Exempel:

Om listan **a** är [1, 4, 9, 16] och listan **b** är [9, 7, 4, 9, 11] så returnerar ett anrop av metoden **append(a, b)** listan [1, 4, 9, 16, 9, 7, 4, 9, 11].

Uppgift 160.

Skriv en statisk metod

```
public static int nrOfTargets(ArrayList<Integer> list, Integer target)
```

som tar en lista **list** samt ett värde **target** och returnerar hur många gånger värdet **target** förekommer i listan **list**.

Uppgift 161.

Skriv en statisk metod

```
public static ArrayList<Integer> reverse(ArrayList<Integer> original)
```

som tar en lista **original** och returnerar en ny lista som innehåller samma element som i lista **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.

Instuderingsfrågor - del 7

arv, grafiska komponenter, händelsestyrd programmering

Uppgift 162.

Vad innebär arv och vad är fördelarna med arv?

Uppgift 163.

Vad är en arvshierarki?

Uppgift 164.

Vad innebär det att en subclass överskyddar en metod i superklassen?

Uppgift 165.

Till vad används operatoren **instanceof**?

Uppgift 166.

Vad är det unika med klassen **Object**?

Uppgift 167.

Vad är ett grafiskt användargränssnitt?

Uppgift 168.

Program som använder grafiska användargränssnitt har en annan logiskt uppbyggnad än program med textbaserad in- och utmatning. På vilket sätt?

Uppgift 169.

Vilka grafiska standardbibliotek finns i Java?

Uppgift 170.

Objekt som tillhör subclasser till klassen **Container** kallas för *behållare*. Vad innebär det att ett grafiskt objekt är en behållare?

Uppgift 171.

Vad är ett fönster? Vilken **Swing**-klass används normalt för att skapa ett fönster?

Uppgift 172.

Hur ser det koordinatsystem ut som används i Java för att adressera positioner i ett fönster? Vilken enhet används i koordinatsystemet?

Uppgift 173.

Vad används objekt av klassen **JPanel** till?

Uppgift 174.

Antag att vi skapat ett fönster enligt nedanstående deklaration

```
JFrame myWindow = new JFrame();
```

Skriv Javasatser för att

- a) sätta bakgrundsfärgen på fönstret till grön.
- b) göra fönstret synligt.
- c) sätta fönstrets höjd till 400 pixels och fönstrets bredd till 200 pixels.
- d) göra fönstret stängningsbart via stängningsrutan i fönstrets ram.

Uppgift 175.

Vad är det för skillnad mellan ett objekt av typen JLabel och ett objekt av typen JButton?

Uppgift 176.

Antag att vi har skapat en etikett med nedanstående deklaration

```
JLabel etikett = new JLabel("Välkommen");
```

Skriv satser för att

- a) läsa texten på objektet etikett.
- b) placera texten centrerad i objektet etikett.
- d) sätta färgen på objektet etikett till gul.
- b) sätter bakgrunden på objektet etikett till ogenomskinlig.

Uppgift 177.

Antag att vi har skapat en knapp med nedanstående deklaration

```
JButton knapp = new JButton("Press");
```

Skriv satser för att

- a) ändra texten på objektet knapp till "Tryck".
- b) sätta färgen på texten i objektet knapp till blå.
- c) tar reda på om objektet knapp är tryckbar.
- d) gör objektet knapp otryckbar.

Uppgift 178.

Vad finns det för relation mellan objekt av klasserna JButton och ActionEvents?

Uppgift 179.

Vilket lyssnarinterface måste man implementera för att fånga händelser av klassen ActionEvents? Vad måste göras för att händelser som genereras av ett objekt skall omhändertas? I vilken metod är det som händelser av klassen ActionEvents fångas och tas om hand?

Uppgift 180.

- a) I klassen ActionEvents finns en instansmetod getSource(). Vad gör denna metod?
- b) I klassen ActionEvents finns en instansmetod getActionCommand(). Vad gör denna metod?

Uppgift 181.

I vilket paket finns de klasser som handhar händelsehantering?

Uppgift 182.

Vad är skillnaden mellan ett JTextArea-objekt och ett JTextField-objekt?

Uppgift 183.

Vad är en layout manager och hur används den?

Uppgift 184.

Hur läggs komponenterna ut i den behållare när FlowLayout används?

Uppgift 185.

Hur läggs komponenterna ut i den behållare när GridLayout används?

Uppgift 186.

Hur läggs komponenterna ut i den behållare när BorderLayout används?

Uppgift 187.

Vilket standardpaket i Java innehåller layout manager klasserna?

Uppgift 188.

Vilken layout manager har JFrame default?

Uppgift 189.

Vilken layout manager har JPanel default?

Uppgift 190.

Hur beskrivs en färg i Java? Hur skapas en färg i Java?

Uppgift 191.

Texter som skrivs ut i grafiska komponenter kan ges olika typsnitt, vilken klass handhar typsnitt?

Uppgift 192.

Om man vill förändra utseendet av en Swing-komponent, d.v.s. en subclass till klassen JComponent, vilken metod måste man då överlagra?

Uppgift 193.

För att rita olika slags figurer och skriva texter i ett grafiskt objekt behövs en "penna", vilken klass tillhör denna "penna"?

Uppgift 194.

Om man vill rita om en komponent från ett program, vilken metod skall man då anropa?