

Prov i Programmering 1 för TE16

Maxpoäng: 70

Provtid: 120 min

String namn = _____

Godkänt: 28

Eclipse

E1 - Nämn 4 st kortkommandon, **specifika för Eclipse**, och förklara vad de gör: (2p)

E2 - Hur gör du smidigast om du vill veta vad för inbyggda metoder en viss variabel har? (1p)

Scanner

S1 - Skriv två metoder som finns inbyggda i klassen Scanner och förklara vad de gör: (2p)

S2 - Skriv en main-metod som ska läsa in heltal från konsolen tills talet 0 blir inskrivet och sedan skriva ut summan av alla tal som blev inskrivna. (3p)*

```
public static void main(String[] args) {
```

```
}
```

Begrepp

B1 - När är det bra att använda en for-loop respektive en while-loop?

(1p)

B2 - Numrera följande datatyper i ordning med den som kan **innehålla** högst numeriskt värde först. (1p)

byte int long float short double

B3 - Ge följande variabler en **lämplig** datatyp och svara på vad som skrivs ut i konsolen:

(4p)*

_____ a = 5.2f * 5d;	System.out.println(a) => _____
_____ b = "J";	System.out.println(b) => _____
_____ c = 2147483647 + 1;	System.out.println(c) => _____
_____ d = 500L;	System.out.println(d) => _____
_____ e = 'e' + 5;	System.out.println(e) => _____
_____ f = null;	System.out.println(f) => _____
_____ g = -30000/5.0 + 7.0f;	System.out.println(g) => _____
_____ h = 500.123 - 0.123;	System.out.println(h) => _____

B4 - Förklara vad konkatenering är och ge skriv ett exempel med kod där du använder det.

(2p)

B5 - Nämn 2 st inbyggda metoder som finns i String och förklara hur de fungerar.

(2p)

B6 - Förklara vad följande metoder har för funktionalitet.

B6a - `Math.sqrt(double d)`

(1p)

B6b - `Math.random()`

(1p)

B7 - Skriv 4 st reserverade nyckelord som inte redan finns med någonstans på provet:

(2p)

B8 - Para ihop följande ord med lämplig förklaring:

(4p)

- | | |
|----------------|--|
| 1. Parameter | ___ Göra något flera gånger. |
| 2. Instansiera | ___ Utvärdera. |
| 3. Iterera | ___ Skapa ett nytt objekt. |
| 4. Exception | ___ Definierat in-värde i en metod. |
| 5. Argument | ___ I följd. |
| 6. Kompilator | ___ Gör om programkod till maskinkod. |
| 7. Evaluera | ___ Värde som skickas med vid ett metदानrop. |
| 8. Sekventiell | ___ Används för felsökning i Java. |
-

B9 - Vad ger ger följande uttryck för utskrift i konsolen?

(4p)*

<code>System.out.println((double) 5 / 0);</code>	=>	_____
<code>System.out.println("Hej" + '!' + 69);</code>	=>	_____
<code>System.out.println(null == (new String[5])[0]);</code>	=>	_____
<code>System.out.println(null == new String());</code>	=>	_____
<code>System.out.println(67 / 7 / 7 - 5d);</code>	=>	_____
<code>System.out.println(new String("hej") == "hej");</code>	=>	_____
<code>System.out.println(("hej" == "hej") + "hej");</code>	=>	_____
<code>System.out.println(Math.PI / 3 > 1.05);</code>	=>	_____

Analys

A1 - Vad gör följande kod och vad ger den för utskrift?

(2p)

```
public static void main(String[] args) {
    System.out.println(fakultet(5));
    System.out.println(fakultet(2));
    System.out.println(fakultet(3)*fakultet(3));
}

private static int fakultet(int n){
    int prod = 1;
    for(int i = n ; i >= 0 ; i--)
        prod *= i;
    return prod;
}
```

A2 - Vad gör följande kod och vad ger den för utskrift?

(2p)

```
public static void main(String[] args) {
    String str = "tfigppu va lav samoI";
    System.out.println(reverse(str));
}

public static String reverse(String str) {
    String revers = "";
    int length = str.length();
    for (int i = 0 ; i < length; i++) {
        revers = revers + str.charAt(i);
    }
    return revers;
}
```

A3 - Vad gör följande kod och vad ger den för utskrift?

(2p)

```
public static void main(String[] args) {
    int tal = (int)(Math.random()*20);
    tal += 6; tal *= 2;
    tal -= 2; tal /= 2;
    tal -= (((tal*2)+2)/2)-6);
    System.out.println(tal);
}
```

A4 - Vad gör följande kod och vad ger den för utskrift?

(3p)*

```
public static void main(String[] args) {

    String tal = "1234"; byte summa = -1;
    char[] siffror = tal.toCharArray();

    for(char siffra : siffror) {
        summa *= Character.getNumericValue(siffra)+256+1;
    }
    System.out.println(summa+512);
}
```

Sortering

S3 - Förklara enkelt med ord hur merge sort fungerar:

(2p)

S4 - Detta är en sorteringsalgorithm som heter bogo sort. Läs koden och förklara hur den sorterar:

(3p)*

```
public class BogoSort {  
  
    public void sort(int[] array) {  
        while (!isSorted(array)) {  
            for (int i = 0; i < array.length; i++) {  
                int randomPosition = (int) (Math.random() * (array.length));  
                int temp = array[i];  
                array[i] = array[randomPosition];  
                array[randomPosition] = temp;  
            }  
        }  
    }  
  
    private boolean isSorted(int[] array) {  
        for (int i = 1; i < array.length; i++) {  
            if (array[i] < array[i - 1]) {  
                return false;  
            }  
        }  
        return true;  
    }  
}
```

S5 - Vilken sorteringsalgorithm är detta ett exempel på? Förklara hur den fungerar.

(3p)*

```
public class sortering3 {  
  
    public static void sort(int[] array) {  
        int inner, outer;  
        int temp;  
  
        int h = 1;  
        while (h <= array.length / 3) {  
            h = h * 3 + 1;  
        }  
        while (h > 0) {  
            for (outer = h; outer < array.length; outer++) {  
                temp = array[outer];  
                inner = outer;  
  
                while (inner > h - 1 && array[inner - h] >= temp) {  
                    array[inner] = array[inner - h];  
                    inner -= h;  
                }  
                array[inner] = temp;  
            }  
            h = (h - 1) / 3;  
        }  
    }  
}
```

S6 - Vilken sorteringsalgorithm är detta ett exempel på? Förklara hur den fungerar.

(3p)

```
public class sortering2 {
    public void sort(int[] arr) {
        int n = arr.length;
        int temp = 0;
        for (int i = 0; i < n; i++) {
            for (int j = 1; j < (n - i); j++) {
                if (arr[j - 1] > arr[j]) {
                    temp = arr[j - 1];
                    arr[j - 1] = arr[j];
                    arr[j] = temp;
                }
            }
        }
    }
}
```

S7 Vilken sorteringsalgorithm är detta ett exempel på? Förklara hur den fungerar.

(3p)

```
public class sortering1{
    private int[] numbers;
    private int number;

    public void sort(int[] values) {
        if (values == null || values.length == 0)
            return;
        this.numbers = values;
        number = values.length;
        sort(0, number - 1);
    }

    private void sort(int low, int high) {
        int i = low, j = high;

        int pivot = numbers[low + (high - low) / 2];

        while (i <= j) {
            while (numbers[i] < pivot)
                i++;
            while (numbers[j] > pivot)
                j--;

            if (i <= j) {
                exchange(i, j);
                i++; j--;
            }
        }
        if (low < j)
            sort(low, j);
        if (i < high)
            sort(i, high);
    }

    private void exchange(int i, int j) {
        int temp = numbers[i];
        numbers[i] = numbers[j];
        numbers[j] = temp;
    }
}
```

Felsökning

F1a - Peka ut alla syntaxfel i följande kod.

(2p)*

```
public static void main(String[] args) {
    String str = "Hej jag heter inte Baok!";
    char[] konsonanter = "bcd fghjklmnpqrstvwxyz".toCharArray();
    for (char c : str) {
        boolean konsonant;
        for (char k : konsonanter) {
            if (Character.toLowerCase(c) == k)
                konsonant = true; break;
        }
        if (konsonant) {
            System.out.print(c);
            System.out.print(Character.toChars(111));
            System.out.print(Character.toLowerCase(c));
        } else
            System.out.print(c);
    }
}
```

F2 - Peka ut de syntaxfel som du ser i följande kod:

(2p)

```
public
class
felsok2
{
    public
    static
    void
    main
    (
    String
    []
    args
    )
    {
        System
        .
        out
        .
        println
        (
        "
        hej
        "
        )
    ;
    }
}
```

F3 - Peka ut de syntaxfel du ser i följande kod:

(2p)

```
public static void main(String[] args) {
    int[] numbers = new int[6];
    numbers = {1,2,3,4,5,6};
    int sum = 0, i = 0;
    for(; i < 6 ;){
        sum += numbers[i];
        i++;}
    String PHRASE = "Summan är: ";
    System.out.println(PHRASE += sum);
}
```

Operatorer

O1 - Skriv siffrorna 1-8 bredvid de villkor som blir sanna när b1, b2, b3 och b4 antar värdena i tabellen nedan.

fall	värde b1	värde b2	värde b3	värde b4
1	true	true	true	true
2	true	true	false	true
3	true	false	true	true
4	true	false	false	true
5	false	true	true	true
6	false	true	false	true
7	false	false	true	true
8	false	false	false	true

O1a - `if(!b2){` (1p)

```
    System.out.println("true");  
}
```

O1b - `if(b1 && !b2){` (1p)

```
    System.out.println("true");  
}
```

O1c - `if(!(b1 && !b2)){` (1p)

```
    System.out.println("true");  
}
```

O1d - `if((!b1 && b2) || (!!b4 || (!b2 && b3)){` (1p)

```
    System.out.println("true");  
}
```

O1e - `if(b1 || b2 || b3){` (1p)

```
    System.out.println("true");  
}
```

O1f - `if((5 > 0 || 27 != 27) && 'c' != null){` (1p)

```
    System.out.println("true");  
}
```

O1g - `if(b2 && b1 && b4 && !b2){`
(1p)
 `System.out.println("true");`
`}`

O2 - Gör minst två if-satser som tillsammans innehåller alla sex jämförelseoperatorer och alla de fyra logiska operatorerna. (Du måste instansiera de **variabler** du använder i dina villkor!) (4p)

