

Prov i Programmering 1 för TE17C

Maxpoäng: 40

Provtid: 70 min

String namn = "Tomas";
Godkänt: ~16

Scanner

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

- .nextInt() - läser in ett heltal av typen int
- .nextLine() - läser in text inkl. blanksteg tills radbrytning sker, t.ex. enter i konsolen

S2 - Skriv kod för att läsa in ett heltal från konsolen och sedan skriva ut alla positiva heltalssiffror som är mindre än det inskrivna talet. (4p)

```
public static void main(String[] args) {  
    Scanner input = new Scanner(System.in);  
    int number = input.nextInt();  
    for (int i = 0; i < number; i++) {  
        System.out.println(i);  
    }  
    input.close();  
}
```

Exempel på körning:

```
<terminated>  
5  
0  
1  
2  
3  
4
```

Begrepp

B1 - När är det bra att använda en for-loop, en while-loop, en do-while loop och en switch-case? (4p)

for: när man vet exakt hur många iterationer man ska genomföra.

while: när man har ett okänt antal iterationer

do-while: som en while-loop men den körs garanterat minst en gång, oavsett villkoret

switch-case: lämpligt när man har ett bestämt antal (få) möjliga utfall, t.ex. en meny

B2 - Ge följande variabler en lämplig datatyp.

```

boolean a = true;
String b = "K";
int c = 10;
Long d = 21474836470032654;
char e = 'w';
String f = "Spelfredag på fredag!";
int g = -30000;
double h = 500.123;

```

B3 - Förklara vad konkatenering är och skriv ett exempel med kod där du använder det.

(2p)

konkatenering = slå ihop två eller flera strängar

"Hej " + "Tomas!" = "Hej Tomas!"

B4 - När är det lämpligt att använda sig utav arrayer? Hur kan man initiera en array? Skriv kod.

(2p)

```
int[] hej = {1, 3, 4, 0};
```

Lämpligt när man vill lagra eller arbeta med stora mängder data av samma typ, t.ex strängar eller int

B5 - Förklara vad följande metoder har för funktionalitet.

B5a - Math.sqrt(double d)

inbyggd metod som tar andra roten ur "d" (1p)

t.ex \sqrt{d}

B5b - Math.pow(double d, double f)

(1p)

inbyggd metod som tar talet "d" upphöjt till talet "f", t.ex d^f

- ✓ 1. Parameter
- 2. Instansiera
- 3. Iterera
- 4. Modulus
- 5. Argument
- 6. Kompilator
- 7. Evaluera
- 8. Sekventiell
- 9. Kommentar
- 10. boolean /int
- 11. Index
- 12. Algoritm
- 13. Konstant
- 14. Programblock

13 Alltid samma.

11 En plats i en array.

14 Kod som innesluts av måsvingar.

1 Definierat in-värde i en metod.

kan vara sant eller falskt
Division som rundar svaret nedåt.

6 Gör om programkod till maskinkod.

5 Värde som skickas med vid ett metodanrop.

12 Ett recept.

9 Text som inte påverkar din kod.

4 Division med rest.

8 I följd.

7 Utvärdera.

3 Göra något flera gånger.

2 Skapa ett nytt objekt.

blev fel i ord 10

Analys

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

(3p)

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

↑
argument

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

} körs lika många gånger
som argumentet,
↑
minska i med 1 varje iteration

* $prod = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = \underline{\underline{120}}$

$prod = 2 \cdot 1 = \underline{\underline{2}}$

☆ $prod = (3 \cdot 2 \cdot 1) \cdot (3 \cdot 2 \cdot 1) = \underline{\underline{36}}$

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

(2p)

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

↑
skriver ut reverse strängen
↑
anropar metoden reverse med argumentet 'str'

```
public static String reverse(String str) {  
    String reverse = "";  
    int length = str.length();  
    for (int i = length-1 ; i >= 1 ; i--) {  
        reverse = reverse + str.charAt(i);  
    }  
    return reverse;  
}
```

← skapar tom sträng
← längden på strängen str
↑
inbyggd sträng-metod som tar en viss bokstav från strängen str och lägger in den i en ny sträng reverse
↑
villkoret slutar på index 1, dvs man skippar utropestecknet

Utskriften blir: "Tomas val av uppgift"

Sortering

S1 - Nämn 4 olika sorteringsalgoritmer och beskriv kortfattat med ord hur de fungerar.
Du behöver inte skriva någon kod:

(4p)

- Quick sort - tar ett valfritt element i en lista
sortera alla mindre till vänster och
sen alla större till höger. Forsätt ta
ett nytt valfritt element och upprepa
- Codetail sort - sorterar från vänster till höger
och sedan från höger till vänster.
upprepa tills färdig
- Insertion sort - sorterar ett tal i bagen i fallande
ordning från höger till
vänster
- Bubble sort - se nedan

S2 - Vilken sorteringsalgoritm är detta ett exempel på? Förklara även hur den fungerar.

(2p)

```
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;  
                }  
            }  
        }  
    }  
}
```

} byter plats på två angränsande tal
om högra talet är mindre
än det vänstra

Ovanstående är Bubble sort kod. Den fungerar
så att man går från vänster till höger
och jämför två angränsande tal med
varandra. Om högra talet är mindre
än vänstra talet så byter talen plats
med varandra. Slutet börjar den om
från början och kör tills alla tal är sorterade.

Felsökning

(Koden är inte färgkodad eftersom det blir för mycket hjälp)

F1 - Peka ut eller ringa in alla syntaxfel i följande kod.

(2p)

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

```
    int a = 5, b = 1, c = 7;
```

```
    while (a < c) {
```

```
        c++;
```

```
        a++;
```

```
        if (a % 9 == 3) {
```

```
            a++
```

```
        }
        System.out.println(b);
```

```
        b++;
```

```
    }
```


```
    int tot = a + b + c;
```

```
    System.out.println("Tot:" tot);
```

```
}
```

 [] ← ska vara hakparenteser

 == ← jämförelseoperator

 saknas ;

 + ska vara ett +

hade det stått

`System.out.printf("Tot: ", tot);`

hade det varit korrekt