

Tentamen i Grundläggande Programvaruutveckling, TDA548

Joachim von Hacht/Magnus Myreen

Datum: 2017-08-14

Tid: 14.00-18.00

Hjälpmedel: Lexikon Engelskt-Valfritt språk.

Betygsgränser:

- U: -23
- 3: 24-37
- 4: 38-47
- 5 : 48-60 (max 60)

Lärare: Joachim von Hacht/Magnus Myreen. Någon besöker ca 15.00 och 16.30, tel. 031/7721003

Granskning: Anslås på kurssida.

Instruktioner:

- För full poäng på essä-frågor krävs ett läsbart, begripligt och heltäckande svar. Generellt 1p för varje relevant aspekt av problemet. Oprecisa eller alltför generella (vaga) svar ger inga poäng. Konkretisera och/eller ge exempel. Det är aldrig någon risk att vara övertydlig!
- Det räcker med enbart relevanta kodavsnitt, övrig kod ersätts med “...” (aldrig import, main-metod, etc....)
- Överkomplicerade lösningar kan ge poängavdrag.
- Vi utgår från att användaren alltid skriver rätt och/eller gör rätt (d.v.s ingen felhantering behövs). Om felhantering skall ingå anges detta specifikt.

LYCKA TILL...

1. Vad avses med? 4p
- a) Klassmetod
 - b) Typfel

Förklara med en eller ett par meningar, du får gärna förtydliga med en skiss eller med kod.

2. Vilka av raderna 1-5 nedan kompilerar ej? Motivera varför! 2p

```
out.println(any(1, 1));           // 1

double any(int i, double d) {    // 2
    return i + d;                 // 3
}

double any(double d, int i) {    // 4
    return d + i;                 // 5
}
```

3. Skriv en metod, sumFib, som givet antalet termer, $n > 0$, returnerar summan av n termer från serien $\frac{1}{1} + \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \frac{1}{8} + \frac{1}{13} + \dots$ 4p

Exempel:

```
out.println(sumFib(10)); // Ger 3.3122872225813405
```

4. Skriv en metod som givet en icke tom array med heltal returnerar en array med alla "ledare" i den givna arrayen. En ledare definieras som ett element där alla element till höger om är mindre än ledaren. Elementet längst till höger räknas alltid som en ledare. För full poäng krävs en lösning *utan* nästlade loopar. Exempel: 8p

| Original-array | Resultat-array (ledarna) |
|-------------------------|--------------------------|
| [14, 9, 11, 7, 8, 5, 3] | [14, 11, 8, 5, 3] |
| [1, 2, 3, 4, 5] | [5] |
| [5, 4, 3, 2, 1] | [5, 4, 3, 2, 1] |
| [5, -4, 3, 2, 1] | [5, 3, 2, 1] |
| [1, 1, 1, 1] | [1] |
| [1] | [1] |

5. Skriv en metod, `sameEnds`, som givet en sträng returnerar den längsta icke-överlappande identiska sekvens som inleder och avslutar strängen. Om någon sådan sekvens inte existerar returneras tomma strängen. 6p

| Insträng | Inledande/Avslutande sekvens |
|-------------|---------------------------------|
| "a" | "" (överlappar, sekvens saknas) |
| "aa" | "a" |
| "aaa" | "a" |
| "abXab" | "ab" |
| "ababXabab" | "abab" |
| "ababXabcb" | "" (sekvens saknas) |

Tillåtna metoder från olika klasser:

`String`

- `charAt(int i)`, ger tecknet vid index `i`.
- `indexOf(char ch)`, ger index för tecknet `ch`, -1 om tecknet saknas.
- `length()` ger längden av strängen.
- `substring(int start, int end)`, ger en delsträng från `start` (inkl.) till `end-1`.
- `substring(int start)`, ger en delsträng från `start` (inkl.) till strängens slut.
- `toArray()`, gör om strängen till en array med tecken
- `split(String str)`, delar upp en sträng i en array av delsträngar utifrån ett visst tecken. Returnerar en `String`-array (`String[]`)
Exempel `"aaa:bb:cccc:dd".split(":")` -> `["aaa", "bb", "cccc", "dd"]`

`StringBuilder`

- `append(String s)`, lägger till strängen `s` sist i `StringBuilder`-objektet.
- `append(char ch)`, som ovan
- `setLength()`, sätter aktuell längd, `setLength(0)` raderar alla tecken.
- `toString()`, omvandlar `StringBuilder`-objektet till en `String`.

`Character`

- `isDigit()` får användas (klassmetod). Metoden returnerar sant för tecknen 0-9.

`Integer`

- `valueOf(String s)` får användas (klassmetod). Metoden omvandlar en sträng till ett heltal. OBS! Att metoden kastar ett undantag för tomma strängen

6. Rita en bild som visar variabler, värden, referenser och objekt samt hur dessa förhåller sig till varann före, respektive efter anropet av metoden `doIt`. Rita som vi ritat under föreläsningarna, lådor, pilar o.s.v. Ni *måste* rita! Ange dessutom vad som skrivs ut. 8p

```
int[] a = new int[]{1, 2, 3};
M m1 = new M(a, "abc");
M m2 = new M(a, "def"); // Before: How does it look after this line?

doIt(m1, 0, m2.a[2], m2.s); // The call

out.print(m2); // After: What will be printed? How does it look?

void doIt(M m, int i, int v, String s) {
    m.a[i] = v;
    m.s = s;
}

class M {
    int[] a;
    String s;
    M(int[] a, String s) {
        this.a = a;
        this.s = s;
    }
    public String toString(){
        return s + Arrays.toString(a);
    }
}
```

7. Skriv följande klasser: 8p
- a) En klass `Particle`. Objektet skall ha en position i 2D och en färg. Klassen skall ha en konstruktor som sätter alla värden och en konstruktor som kan skapa en kopia givet ett annat `Particle`-objekt.
 - b) En klass `ParticleSystem`. Klassen skall ha en array med `Particle`-objekt. Arrayen skall initieras m.h.a. en konstruktorparameter. Klassen skall på samma sätt som `Particle` ha en konstruktor som skapar en `ParticleSystem`-objekt givet ett annat `ParticleSystem`-objekt (en kopia). Efter anropet på denna skall det finnas två helt separata `ParticleSystem`-objekt med samma data.

Klasserna skall vara så icke-muterbara som möjligt och dölja så mycket som möjligt av sin data (information hiding).

Visa, m.h.a. av din kod, hur du skapar några `Particle`-objekt och hur du skapar ett `ParticleSystem`-objekt samt en kopia av detta.

8. Nedan har ni klassen Split.

8p

```
class Split {
    Split left, right;
    public Split(Split left, Split right) {
        this.left = left;
        this.right = right;
    }
    public int sum() {
        int res = 1;
        if (left != null) {
            res = res + left.sum();
        }
        if (right != null) {
            res = res + right.sum();
        }
        return res;
    }
}
```

a) Vad skrivs ut vid första utskriften i koden nedan ?

3p

b) Vad skrivs ut vid andra utskriften i koden nedan?

7p

```
Split t = new Split(new Split(null,new Split(null,null)),null);
System.out.println(t.sum());          // Första utskriften
Split s = new Split(null,null);
for(int i=0; i < 6; i++) {
    s = new Split(s,s);
}
System.out.println(s.sum());          // Andra utskriften
```

9. Skriv en metod som givet ett godtyckligt positivt heltal skriver ut ett mönster enligt exemplen nedan. Nedan ser man utskriften för indata 8 och indata 15. Tips: det lönar sig att skapa en matris inne i metoden. 10p

```
// Indata: 8
```

```
1.....  
1.444443.  
1.5....3.  
1.5.87.3.  
1.5..7.3.  
1.5666.3.  
1.....3.  
12222222.  
.....
```

```
// Indata: 15
```

```
1.....  
1.444444444444443.  
1.5.....3.  
1.5.888888887.3.  
1.5.9.....7.3.  
1.5.9.XXXX.7.3.  
1.5.9.X..X.7.3.  
1.5.9.X.X.X.7.3.  
1.5.9.XXX.X.7.3.  
1.5.9.....X.7.3.  
1.5.9XXXXXX.7.3.  
1.5.....7.3.  
1.56666666666.3.  
1.....3.  
122222222222222.  
.....
```