

Tentamen i Imperativ Programmering med Grundläggande Objektorientering, DIT012

Joachim von Hacht

Datum: 2017-04-12

Tid: 08.30-12.30

Hjälpmedel: Engelskt-Valfritt språk lexikon

Betygsgränser:

- U: -23
- G: 24-43
- VG: 44-60 (max 60)

Lärare: Joachim von Hacht, tel. 031/772 10 03. Någon besöker ca 09.30 och 11.00,

Granskning: Tentamen kan granskas på studieexpeditionen. Vi ev. åsikter om rättningen eposta mig och ange noggrant vad du anser är fel så återkommer jag (ev. ta en bild och skicka).

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. Förklara med en eller ett par meningar: 4p
- a) Referensvariabel.
 - b) Typkompatibel.

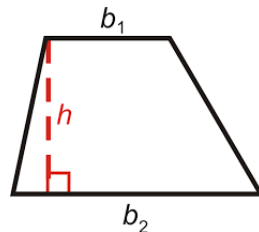
2. Vilket av kodavsnitten a) eller b) nedan leder till kompileringsfel? Varför? Motivera kortfattat. 4p

a)
`public int triple(int i){ return 3 * i; }
public static int inc(int i){ return triple(i) + 1; }`

b)
`public static int triple(int i) { return 3 * i;}
public int inc(int i) { return triple(i) + 1;}`

3. Skriv ett program som beräknar arean för en trapetsoid, se bild. Skriv hela programmet i en metod `void program()`. Arean ges av formel: $A = h(b_1 + b_2)/2$. Programmet skall fungera enligt: 2p

```
Input b1 > 2  
Input b2 > 3  
Input h > 5  
Area = 12.5      (program avslutat)
```



4. Värdet för funktionen $\log(1+x)$, kan beräknas med serien $x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} \dots$ ¹. Skriv en metod `log1p(double x, int nTerms)` som beräknar värdet för x som summan av $nTerms$ termer ut serien. Exempel: 4p

```
out.println(log1p(0.5, 20));      // Ger 0.4054651081081644
```

¹Gäller då x är litet, inget du behöver ta hänsyn till.

5. Skriv en metod `String[] generate(int n)`, som genererar alla strängar med "förfäder" enligt nedan. Parametern `n` anger antal "generationer". Följande metoder ur `Math` för användas: `pow(b,e)` ger `b` upphöjt till `e`, `sqrt(n)`, ger roten ur `n`.

6p

```
ME                //n = 4
ME.dad
ME.mum
ME.dad.dad
ME.dad.mum
ME.mum.dad
ME.mum.mum
ME.dad.dad.dad
ME.dad.dad.mum
ME.dad.mum.dad
ME.dad.mum.mum
ME.mum.dad.dad
ME.mum.dad.mum
ME.mum.mum.dad
ME.mum.mum.mum
```

6. Skriv en boolesk metod, `isOrdered(String order, String target)`, som avgör om ordningen för alla tecken i parametern `order` gäller för alla tecken i `target`, se exempel. Om så ger metoden sant annars falskt. Vi antar att parametern `order` är en sträng med unika tecken och minst två tecken lång. Exempel

8p

```
// Alla a kommer före alla b som kommer före alla c, sant
out.println(isOrdered("abc", "aa bb cc"));
// Alla c kommer inte före alla b, falskt
out.println(!isOrdered("acb", "aa bb cc"));
// Alla b kommer inte före alla c, falskt
out.println(!isOrdered("abc", "aa bb ccc b"));
// Allt nedan skall skriva ut true
out.println(isOrdered("ec", "aa eee becb c dddc"));
out.println(!isOrdered("ax", "aa bb cc"));
out.println(isOrdered("ax", ""));
```

- a) Beskriv i ord grundtanken i din lösning.
b) Implementera lösningen. (SE NÄSTA SIDA)

Färdiga metoder (från olika klasser) som får användas.

`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

`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(ch)` (klassmetod). Metoden returnerar sant om `ch` är 0-9.

`Integer`

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

7. Vad skriver programmet nedan ut? Ge en förklaring genom att rita bilder med variabler, värden och referenser som visar hur det ser ut innan och efter anropet till `b1.copy()`. Du måste rita!

6p

```
A a = new A(9);
B b1 = new B(9, a); // Hur ser det ut nu?
B b2 = b1.copy();   // Anrop
                  // Hur ser det ut nu?

b2.a.i = 37;
out.print(b1.a.i); // Vad skrivs ut?

public class A {
    int i;
    public A(int i) {this.i = i;}
}

public class B {
    int i;
    A a;
    public B(int i, A a) { this.i = i; this.a = a;}
    public B copy() {return new B(i, a);}
}
```

8. Antag att vi skall skriva ett spel. Spelet innehåller en 2D-värld med hinder. Skapa en klass `World` och en klass `Obstacle` för respektive. Följande gäller:

8p

- Alla `Obstacles` har en position som ges av en x- och en y-koordinat. Det skall vara möjligt att ange x och y då man skapar en `Obstacle`.
- `World` skall ha en referens till en array av `Obstacles`. Det skall vara möjligt att ange alla `Obstacles` då man skapar en `World`.
- `World` skall ha en metod `contains(Obstacle o)` som ger sant om världen innehåller hindret o.
- Båda klasserna skall vara så icke-muterbara "som möjligt".
- Visa hur man skapar en värld med några hinder.

9. Skriv en metod `int[] splitInt(int n, int nTerms)` som delar upp parametern `n` i `nTerms` stycken slumpmässiga icke-negativa termer vars summa blir `n`, `nTerms` > 0 . Exempel: `splitInt(7, 4)` kan t.ex. ge något av följande (totalt finns 210 lösningar):

10p

```
[5, 0, 2, 0]    // 5 + 0 + 0 + 2 = 7
[0, 7, 0, 0]
[0, 1, 4, 2]
[1, 2, 3, 1]
```

Du kan anta att du har tillgång till en sorteringsmetod, t.ex. `Arrays.sort()`.

- Beskriv i ord hur din metod är tänkt att fungera.
- Implementera metoden i kod.

För full poäng måste metoden vara effektiv och ge en rimlig sannolikhet för alla utfall (att i exemplet ovan slumpa fyra tal till summan blir 7 ger 0 poäng).

10. Vad skriver programmet nedan ut? Du måste ge en uttömmande förklaring.

8p

```
Base b = new Derived();
out.println(b.get());    // Vad skrivs ut?

public class Base {
    private int i = 5;
    public int get() { return inc(); }
    public int inc() { return i + 1; }
}

public class Derived extends Base {
    private int i = 10;
    public int inc() { return i + 1; }
}
```