

Tentamen i Imperativ Programmering med Grundläggande Objektorientering, DIT012

Joachim von Hacht

Datum: 2017-08-15

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. Du får gärna ge kodexempel eller rita. 4p
- a) Referensvariabel kontra primitiv variabel (jämför dessa).
 - b) Överlagrad (overloaded).

2. Vilka rader nedan ger kompileringsfel? Motivera kortfattat! 4p

```
{
    int i = 0;           // 1
    {
        int i = 0;       // 2
        int j = 0;       // 3
        out.println(i);  // 4
        out.println(j);  // 5
    }
    int i = 0;           // 6
    int j = 0;           // 7
    out.println(i);      // 8
    out.println(j);      // 9
}
out.println(i);         // 10
out.println(j);         // 11
```

3. Givet två grupper av positiva heltal kan man, i vissa fall, öka medelvärdet i *båda* grupperna genom att flytta ett tal från en grupp till en annan. Exempel: 6p

Grupp A	Grupp B	
[3, 1, 2]	[4, 3, 4, 5]	Flytta 3 från B till A
[6, 4, 9, 5]	[4, 4, 3]	Flytta 4 från A till B
[1,1,1]	[2,2,2]	Saknar lösning

Skriv en metod som givet två arrayer med heltal returnerar en sträng som beskriver vilket tal som skall flyttas, t.ex. 3:B->A och 4:A->B enligt ovan. Saknas lösning returneras "No solution".

4. Skriv en metod som givet en matris med heltal returnerar en array med alla negativa tal ur matrisen. Exempel: 6p

	[1, -2, 3,	
Matrisen	-4, 5, -6,	ger resultatet [-2, -3, -4, -6, -8]
	7, -8, 9]	

5. Ett prydnadsord är ett ord som inleds och avslutas med en likadan delsträng t.ex. "onion" som inleds och avslutas med "on". Delsträngen är alltid kortare än hela ordet (äkta delsträng). Ordets grad bestäms av längden på delsträngen, "onion" har alltså grad två. Exempel: 8p

- "underground" är prydnadsord av grad tre därför att det börjar och slutar på "und" som har längden tre.
- "aaaa" är ett prydnadsord av grad tre eftersom delsträngen skall vara kortare än hela ordet

Med hjälp av ett prydnadsord kan man skapa en prydnadssträng på följande sätt:

- a) Ta bort delsträngen från början av prydnadsordet
- b) Det som blir kvar av prydnadsordet lägger man till på slutet av (hela) prydnadsordet lika många gånger som graden för prydnadsordet.

Exempel: "onion" delsträng: "on" (grad två)

- Ta bort inledning "on" ger "ion"
- Lägg till "ion" två gånger på slutet ger "onionionion" som är prydnadssträngen.

Fler exempel:

```
"zvioz" --> "zviozvioz" (delsträng z, zvioz + vioz)
"alfalfa" --> "alfalfalfalfalfalfa" (delsträng alfa,
    alfalfa + lfa + lfa + lfa + lfa)
"aaaa" --> "aaaaaaaa" (delsträng aaa, aaaa + a + a + a, )
"" --> "" (tomma strängen, inte ett prydnadsord)
"jackhammer" --> "jackhammer" (inte ett prydnadsord, se nedan)
```

- a) Skriv en metod som givet en insträng returnerar den inledande/avslutande delsträngen om insträngen är ett prydnadsord. Om ej returneras tomma strängen.
- b) Skriv en metod som givet en insträng skapar en prydnadssträng om insträngen är ett prydnadsord. Om ej returneras insträngen.

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.
- toCharArray(), gör om strängen till en array med tecken
```

- `endsWith(s)`, sant om strängen avslutas med `s`. (V.G.V)
- `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`.

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

```
A a1 = new A(4, new B());
A a2 = new A(6, new B()); // Before, how does it look?
doIt(a1, a2);             // Call
out.println(a1.i + a2.i); // After, how does it look?

void doIt(A a1, A a2){
    a1.get().set(a2);
    a1.get().get().set(7);
}

class A {
    int i;
    B b;
    public A(int i, B b) { this.i = i; b.set(this); this.b = b;}
    B get(){ return b;}
    void set( int i){ this.i = i;}
}

class B {
    A a;
    void set( A a){ this.a = a;}
    A get(){ return a;}
}
```

7. Vi skall modellera ett hisssystem. Hissystemet hanterar flera hisskorgar. Skriv följande klasser:

6p

- En klass Car (hisskorgen). Alla hisskorgar har en riktning och ett våning den skall åka till. Klassen skall ha en konstruktor som initierar instansvariablerna. Klassen skall ha en metod moveTo som sätter vilken våning hissen skall åka till.
- En klass Elevator (hissystemet). Klassen har en array av hisskorgar. Arrayen skall initialiseras m.h.a. en konstruktor som tar ett antal hisskorgar. Klassen skall ha en metod moveTo(carNumber, floor) som bestämmer vilken hisskorg som skall åka till en viss våning.

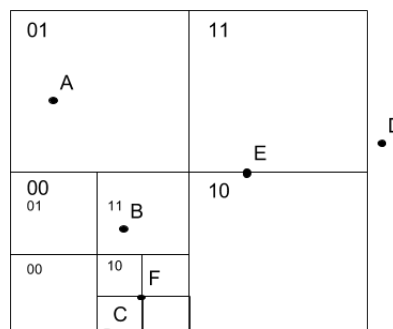
Klasserna skall vara så icke-muterbara som möjligt och dölja så mycket som möjligt av sin data (information hiding). Visa, m.h.a. av dina klasser, hur du skapar några hisskorgsobjekt, hur du skapar ett hisssystem och hur du anropar hisssystemet så att det bestämmer att någon korg skall till en viss våning.

8. Antag att vi arbetar med punkter i ett kvadratisk 2D-område. Man kan koda i vilken del av området en punkt befinner sig genom att upprepade gånger dela området i fyra delkvadrater enligt bilden nedan. Delarna kodas, på alla nivåer, 00, 10, 01, 11 (bara kodning för två nivåer visas).

12p

I vilken delkvadrat punkten befinner sig kodas som den sträng man får om man börjar från hela kvadraten och därefter förflyttar sig till mindre och mindre delkvadrater tills man hittar punkten. Delkvadraten för:

- A kodas 0001. Inledande 00 är koden för hela kvadraten, därefter 01 eftersom punkten är i övre vänstra kvadraten.
- B kodas 000011.
- C kodas 00001000 (avslutande 00 visas inte i bilden).
- D är utanför och kodas som tomma strängen.
- E ligger på en linje och ges 00 eftersom vi inte kan säga vilken delkvadrat den ligger i.
- F ligger också på en linje, ges på samma sätt som E, 000010.



Skriv en metod `String getKey(double a, double b, double width, double height)` som returnerar koden för en punkts delkvadrat i form av en sträng (som ovan). Parametrarna `a` och `b` är punktens koordinater och `width` och `height` är storleken på området (den största kvadraten). Det räcker om metoden klarar tre nivåer (d.v.s. stränglängd max 6 tecken). Det är tillåtet att använda metoden `abs()` från `Math`.

9. Vad skriver följande kodstycker ut? Varför? Du måste motivera!

6p

```
Classifier c = new Classifier();
Object[] o = new Object[]{ new Integer(0),
                           new String(), new Double(0)};
for (int i = 0; i < o.length; i++) {
    out.println(c.classify(o[i]));
}

class Classifier {

    String classify(Integer i){return "It's an integer";}
    String classify(String s){ return "It's a String"; }
    String classify(Object o){ return "Don't know, it's anything";}
}
```