

Tentamen Imperativ programmering med grundläggande objektorientering, DIT012

Joachim von Hacht

Datum: 2016-01-09

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. Någon besöker ca 10.00 och 11.30, tel. 0707/311066

Granskning: Tentamen kan granskas på studieexpeditionen. Vi ev. åsikter om rättningen eposta mig och ange noggrant vad du anser är fel så återkommer jag (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. Vad avses med (förklara med en eller ett par meningar, du får gärna förtydliga med en skiss eller med kod)? 4p
 - a) Primitiv typ.
 - b) Tilldelning.
 - c) Överlagring.
 - d) Alisas-problem.
2. Skriv ett program som beräknar ersättningsresistansen för två parallellkopplade motstånd, R1 och R2. Ersättningsresistansen ges av : $R_{ers} = \frac{R_1 R_2}{R_1 + R_2}$. 2p
Exempelskärning

```
Ange R1 > 10
Ange R2 > 10
Ersättnings R : 5.0
```

3. Ett Fibonacci-ord är en specifik sträng av ettor och nollor t.ex. "01001010". Sekvensen definieras som: 4p

$$fibword(n) = \begin{cases} "0" & n = 0 \\ "01" & n = 1 \\ fibword(n-1) + fibword(n-2) & n > 1 \end{cases}$$

Där + är den vanliga konkateneringsoperatoren.

De 6 första Fibonacci-orden är (citattecken utelämnade): 0, 01, 010, 01001, 01001010, 0100101001001

Skriv en metod som givet ett heltal, $n \geq 0$, returnerar det n :te Fibonacci ordet.
Exempel:

```
out.println(fibonacciWord(0)); // Ger "0"
out.println(fibonacciWord(3)); // Ger "01001"
```

4. Förklara i detalj hur ett metodanrop går till. Utgå från något konkret kodexempel med en (ev. meningslös) metod som har minst två parametrar och ett returvärde. För full poäng krävs att en av parametrarna är av referenstyp. 6p
5. Vi vill undersöka om en robot, efter en given promenad, når en viss position. Följande gäller: 6p
 - Roboten rör sig i ett logiskt (vanligt) koordinatsystem (roboten har en x och y position).
 - Robotens promenad ges av en följd av heltal, $10 \leq n \leq 49$, där tiotalssiffran ger riktning enligt: 1 = norr, 2 = öst, 3 = söder och 4 = väst. Entalssiffran ger antalet steg i given riktning (alltså 0-9).

Skriv en funktion, boolean willHit(int startX, int startY, endX, endY, int[] walk), som givet robotens startposition (startX och startY) samt vägen (walk) svarar sant om roboten stannar på en given slutposition (endX och endY) efter att ha vandrat promenaden. Om ej returneras falskt. Exempel:

```
int[] walk = {13, 24, 32, 22};
out.println(willHit(2, 2, 8, 3, walk)); // Ger true
```

6. Antag att vi vill skapa en datorversion av spelet Yatzy, se bilaga. 4p

- Koda tre (inledningsvis tomma) tänkbara Java-klasser som man skulle kunna använda i ett program.
- Om de tomma klasserna har några uppenbara instansvariabler ange någon eller några av dessa (max 2 rækker).
- Ange en konstruktor för varje klass ovan.

7. Vi vill översätta svenska meningar till "allspråket". Detta sker genom att lägga till "all" efter den första konsonanten i alla ord. Exempel: 6p

"Hur är läget?" -> "Hallur ärall lalläget?"

Vi antar att alla ord i meningen åtskiljs av exakt ett mellanslag. Skriv en metod String toAll(String in) som utför översättningen. Parametern in är den svenska meningen och resultatet är en ny mening med den översatta texten.

OBS! För full poäng krävs att du strukturerar lösningen m.h.a. minst två lämpliga hjälpmetoder!

Tillåtna metoder från String-klassen

- 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.
- 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"]

8. Betrakta program() nedan och ange: Vad skrivs ut? Hur många variabler finns i programmet (hur många olika behållare för värden kommer att användas)? Hur många objekt är inblandade? Du måste motivera, att bara ange några talvärden ger inga poäng! Rita dessutom en bild som visar variabler, referenser och objekt då raden med *) har körts (alltså vilka pilar pekar på vad).

8p

```
class Wrapper {
    Integer i;
    Wrapper(Integer i) {
        this.i = i;
    }
}
class WrapperSwapper {
    void swap(Wrapper v1, Wrapper v2) {
        Integer tmp = v1.i;
        v1.i = v2.i;
        v2.i = tmp;
    }
}
void program(){
    WrapperSwapper ws = new WrapperSwapper();
    int a = 1;
    int b = 2;
    Wrapper v1 = new Wrapper(a);
    Wrapper v2 = new Wrapper(b);
    ws.swap(v1, v2); // <----- *)
    out.println("a= " + v1.i + " b= " + v2.i);
}
```

9. Skapa en klass, `SortedIntegerSet`, för sorterade mängder av heltal, d.v.s. alla värden är unika och sparas i sorterad (stigande) ordning. Klassen skall kunna användas enligt (vi antar att det finns en färdig `toString()`-metod): 12p

```
int[] elems = {2, 4, 6, 3, 8, 2, 2, 6, 7, 9};
SortedIntegerSet s1 = new SortedIntegerSet(elems);
out.println(s1); // [2, 3, 4, 6, 7, 8, 9]
SortedIntegerSet s2 = new SortedIntegerSet(s1);
out.println(s1.remove(9)); // true
out.println(s1.remove(9)); // false
out.println(s2.add(5)); // true
out.println(s2.add(5)); // false
out.println(s1); // [2, 3, 4, 6, 7, 8 ]
out.println(s2); // [2, 3, 4, 5, 6, 7, 8, 9]
SortedIntegerSet s3 = s1.headSet(6);
out.println(s3); // [2, 3, 4] a *new* set with all elements less than 6
```

Ni kan fritt välja representation, om man vill använda `List<Integer>` och `ArrayList` så finns följande metoder (l är ett listobjekt):

```
- l.add(Integer i) // Lägger till i sist i listan,
                  // ger sant som returvärde om det
                  //gick att lägga till
- l.add(int index, Integer i) // Lägger till i vid angivet index,
                              // returvärde som ovan
- l.remove(Integer i) // Tar bort i ur listan, ger sant om
                     // borttagningen lyckades
- l.get(Integer index ) // Returnerar index i
- l.contains(Integer i) // Sant om i finns i listan
- l.size() // Ger aktuell storlek på listan
- l.isEmpty() // Sant om listan är tom
```

10. Skriv en metod, `int minValue(int[][] m)`, som givet en triangulär matris beräknar den minsta summa man kan få genom att vandra från toppvärdet (4:an i exemplet) till något värde på sista raden. Förflyttningar får göras "rakt ned" eller "snett till höger/vänster" en rad i taget. Det är tillåtet att förändra parametern `m`. Exempel 8p

```
int[][] m = {
    {4},
    {3, 2},
    {1, 2, 1},
    {2, 4, 1, 3}
};
Minsta värde: 8 ges av vägen 4, 2, 1, 1
```