

Tentamen i Grundläggande Programvaruutveckling, TDA548

Joachim von Hacht

Datum: 2017-10-26

Tid: 08.30-12.30

Hjälpmedel: Lexikon Engelskt-Valfritt språk.

Betygsgränser:

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

Lärare: Joachim von Hacht . Någon besöker ca 10.00 och 11.30, 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) Referensvariabel respektive primitiv variabel.
 - b) this (i en klassdeklaration)

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

2. Vilka av raderna 1-4 nedan kompilerar ej? Motivera varför! 2p

```
int result = add(div(sub(7, 6), 5), 4);           // 1

int add(int i, int j) { return i + j; }           // 2
int sub(int i, int j) { return i - j; }           // 3
double div(double i, double j) { return i / j; } // 4
```

3. För att beräkna ett närmevärde till roten ur n kan man använda följande iterativa formel: $x_{i+1} = \frac{1}{2}(x_i + \frac{n}{x_i})$, där x_i är en startapproximation till roten ur n och x_{i+1} är nästa, exaktare värde. Skriv en metod som använder formeln för att beräkna ett närmevärde till roten ur n . För tillräcklig noggrannhet räcker det med 5 iterationer. TIPS: Sätt startapproximationen till n . 4p

4. Skriv en metod som givet två arrayer, a1, a2 och en position p, $0 \leq p \leq a1.length$, lägger till alla element i a2 till a1 med start vid position p i a1, om elementet inte redan finns i a1. Original-arrayerna får inte ändras. För full poäng *krävs* en lämplig funktionell nedbrytning! Exempel; 12p

a1	a2	p	resultat
[1, 2, 3, 4, 5]	[55, 66, 3]	0	[55, 66, 1, 2, 3, 4, 5]
[1, 2, 3, 4, 5]	[3, 55, 66]	2	[1, 2, 55, 66, 3, 4, 5]
[1, 2, 3, 4, 5]	[55, 3, 66]	4	[1, 2, 3, 4, 55, 66, 5]
[1, 2, 3, 4, 5]	[55, 3, 66]	5	[1, 2, 3, 4, 5, 55, 66]
[1, 2, 3, 4, 5]	[3, 4, 5]	2	[1, 2, 3, 4, 5]

5. Skriv en metod som returnerar maximalt djup för nästlade parenteser i en sträng. Om strängen är obalanserad (d.v.s. parenteserna matchar inte) så returneras -1. Exempel;

6p

Insträng	Djup
"(a)"	1
"((a)(bcde))"	2
"((a)((bc)d))"	3
"((abc)(d))"	-1

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

Integer

- `valueOf(String s)` (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!

6p

```
A a1 = new A(new int[]{1, 2});
A a2;                                // Före
a2 = doIt(a1);                        // Anrop
                                     // Efter

A doIt(A a) {
    A tmp = new A();
    a.i = tmp.i;
    return tmp;
}

class A {
    int[] i;
    A() { i = new int[]{0, 0};}
    A(int[] i) { this.i = i;}
}
```

7. Skriv följande klasser (se TIPS sist):

- a) En klass Cell som skall motsvara en cell i ett kalkylark (typ Excel). Alla celler har en rad och en kolumn samt ett innehåll. Man skall kunna sätta rad och kolumn då man skapar en cell. Klassen skall ha en metod som flyttar cellen till nästa kolumn och en metod som returnerar kolumnvärdet. 3p
- b) Skapa en klass för kalkylark (typ Excel Sheet). Ett kalkylark har en lista med celler samt ett aktuellt maxindex för antal rader och ett aktuellt maxindex för kolumner. Man skall kunna ange aktuellt maxindex för rader och kolumner då man skapar arket. Då arket skapas skall det fyllas med lika många (tomma) celler (cellobjekt) som ges av aktuellt maxindex för rader och kolumner (d.v.s. $\text{maxRow} * \text{maxCol}$ st.). Cellernas rader och kolumner ges värdena 0-maxindex-1 för både rad och kolumn. 4p
- c) Skapa en metod i Sheet som lägger till en ny kolumn (med tomma celler) givet ett kolumnindex (kolumnen har lika många rader som aktuellt maxindex för rader). Alla celler med kolumnindex lika med eller större än den nya kolumnen flyttas en kolumn till höger. 5p

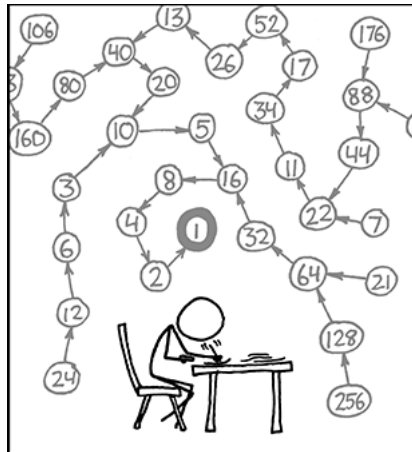
Klasserna skall vara så icke-muterbara som möjligt och dölja så mycket som möjligt av sin data (information hiding).

TIPS: Tillåtet att använda List/ArrayList med metoderna

- add(o), lägger till objektet o sist i listan
- remove(o), tar bort objektet o ur listan
- contains(o), sant om objektet o finns i listan.
- indexOf(o), ger index för objektet
- size(), ger längden på listan
- get(i), ger objektet för index i

8. Tänk dig att du skriver upp alla positiva heltal på ett oändligt stort papper. Från varje tal $n > 1$ ritar du nu en pil till talet $n/2$ om n är jämnt och till $3n+1$ om n är udda. Ett avsnitt ur den graf som bildas visas i bilden nedan. Skriv en metod som, givet två olika heltal beräknar hur långt ifrån varandra (antal pilar, oavsett riktning) de är i grafen. 8p

TIPS: Samma som uppg. 7 tillåtet.



9. Antag att du har koden nedan. Utskrifterna ger true respektive false. Varför? Förklara! 6p

```
Book b1 = new Book("aaa");
Book b2 = new Book("aaa");
out.println(b1.equals(b2));           // True
List<Book> bs = new ArrayList<>();
bs.add(b1);
out.println(bs.contains(b2));         // False!

class Book {
    private final String isbn;
    public Book(String isbn) { this.isbn = isbn; }
    public boolean equals(Book b) {

        if (b == this) { return true;}
        if (b == null) { return false; }
        return b.isbn.equals(isbn);
    }
}
```