

Tentamen i Grundläggande Programvaruutveckling, TDA548

Joachim von Hacht

Datum: 2019-01-07

Tid: 08.00-12.00

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 9.30 och 10.30, tel. 031/7721003

Granskning: Anslås på kurssida.

Instruktioner:

- För full poäng på essäfrågor 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 räcker med enbart relevanta kodavsnitt, övrig kod ersätts med “...” (aldrig import, main-metod, etc....). 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.
- Lösningarna måste klara de fall som anges *samt fall som är principiellt lika*. Lösningar som bara klarar exemplen räcker *inte*. Överkomplicerade lösningar kan ge poängavdrag.
- Färdiga klasser m.m. som får användas anges för varje uppgift. Anges inget får man alltid använda de grundläggande språkliga konstruktionerna, arrayer, egna metoder och egna klasser.

LYCKA TILL...

1. Vad avses med? 4p
- a) Instans
 - b) Sidoeffekt

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

2. Koden nedan ger problem. Varför? Motivera! 2p

```
void program(){
    MyClass m = new MyClass();
    m.check(7);
}

class MyClass {
    static Integer i;
    boolean check(int threshold) {
        return i <= threshold;
    }
}
```

3. En "klump" är en del av en heltals-array där alla element har samma värde. D.v.s. minst två intilliggande index måste ha samma värde. Exempel: 6p

```
[1, 2, 3, 4, 5]           // Ingen klump
[1, 1, 3, 4, 5, 6, 8, 9]   // En klump 1,1
[1, 1, 3, 4, 5, 8, 8, 8]   // Två klumpar 1,1 och 8,8,8
[5, 5, 4, 4, 3, 8, 8, 8]   // Tre klumpar 5,5 och 4,4 och 8,8,8
[1, 1, 1, 1]              // En klump
```

Skriv en metod som givet ett fält returnerar antalet klumpar i fältet.

4. Givet en kvadratisk matris med $m \geq 4$ element och ett heltal $2 \leq n \leq \sqrt{m}$ (i array-form). Skriv en metod som returnerar sant om det finns någon kvadratisk delmatris med sidan n sådan att alla element är 1. För full poäng krävs en lämplig funktionell nedbrytning. List och String inte tillåtna. Exempel: 12p

matris	n	resultat
[0, 1, 0, 1]	2	true
[0, 0, 1, 1]		
[1, 0, 1, 1]		
[0, 1, 0, 0]		
[0, 1, 0, 1]	2	false
[0, 0, 1, 0]		
[1, 0, 1, 1]		
[0, 1, 0, 0]		
[0, 1, 1, 0]	3	true
[1, 1, 1, 1]		
[1, 1, 1, 0]		
[1, 1, 1, 0]		

5. Skriv en metod, String compress(String str), som returnerar en "komprimerad" version av strängen str. Komprimeringen ersätter följder av lika tecken med ett numeriskt värde för antalet tecken samt tecknet, se nedan. Strängen består bara av de engelska bokstäver (a-z, A-Z). Alla metoder i appendix får användas. 8p

str	komprimerad
"aaaaaeefeffw"	"5a4e3f1w"
"qqqqqqqqqqq"	"12q"
""	""

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 kursen, lådor, pilar o.s.v. Strängar kan ritas förenklat som t.ex. "abc".

8p

```
A[] as = { new A(1), new A(2) , new A(3)};
as[0].a = as[1];
as[1].a = as[2];
as[2].a = as[0];
M m = new M(as[0]); // Before
doIt(m);           // Call
                  // After

void doIt(M m) {
    m.a = m.a.a;
}

class M {
    A a;
    public M(A as) { this.a = a;}
}

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

7. Vi skall skapa en objektorienterad modell av ett företag.

10p

- Skriv en klass `Employee` för en anställd. En anställd har ett id och ett namn. All data skall anges då man skapar ett objekt. Vi antar att alla setters/getters vi ev. behöver finns (gäller även nedan).
- Skriv en klass `Office` för ett kontor. Ett kontor har ett id, ett antal anställda och ett maximalt antal anställda. Id och maximalt antal anställda skall anges då ett kontor skapas.
- Skriv en class `Company` för hela företaget. Ett företag har ett antal kontor och anställda. Lägg till en metod `relocate(employee, fromOfficeId, toOfficeId)` som flyttar en anställd från kontoret med id:t `fromOfficeId` till kontoret med `toOfficeId`. Metoden skall returnera `true` om flytten lyckades annars `false` (alla villkor för en flytt skall kontrolleras). Alla metoder i appendix är tillåtna.

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

8. Betrakta koden nedan och ange för varje rad a) - g) *en* av följande.

10p

- Kompilerar ej
- Körningsfel
- Om inget av ovan, ange vad som skrivs ut.

Du måste ge en kort motivering för varje svar (ibland flera satser på varje rad p.g.a. utrymme)!

```
a) A a = new C(); a.doIt(1.0);
b) C c = new B(); c.doIt(1.0);
c) A a = new B(); a.doIt(1);
d) Object o = new C(); D d = (D) o; d.doIt(1.0);
e) IX i = new C(); A a = (A) i; a.doIt(1.0);
f) Object[] os;
   A[] as = {new A(), new A()};
   os = as;
   ((A) os[0]).doIt(5);
g) List<Object> os = new ArrayList<>();
   List<A> as = new ArrayList<>();
   os = as;
   os.add(new B());
   ((B) os.get(0)).doIt(5);

// --- Interfaces and classes
interface IX { void doIt(double d);}

class A implements IX {
    public void doIt(double d) { out.println("doIt A"); }
}
class B extends A {
    public void doIt(int i) { out.println("doIt B"); }
}
class C extends A {
    public void doIt(double d) { out.println("doIt C"); }
}
class D {
    public void doIt(double d) { out.println("doIt D"); }
}
```

APPENDIX

Ur klassen String

- equals(s), avgör om en sträng innehåller samma tecken som en annan.
- 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.

Ur klassen 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.

Ur List/ArrayList

- get(i), ger objektet för index i
- add(o), lägger till objektet o sist i listan
- set(i, o), lägger till objektet vid index i, flyttar övriga till höger.
- remove(o), tar bort objektet o ur listan, returnerar true om detta lyckades annars false
- remove(i), tar bort och returnerar objektet vid index i ur listan
- removeAll(list), tar bort alla element i list.
- contains(o), sant om objektet o finns i listan.
- indexOf(o), ger index för objektet
- size(), ger längden på listan

Klassen Random med metoden nextInt() är alltid tillåten.