

Tentamen i Grundläggande Programvaruutveckling, TDA548

Joachim von Hacht

Datum: 2018-08-20

Tid: 14.00-18.00

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 15.00 och 17.00, 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) Returtyp
 - b) this

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

2. Koden nedan kompilerar inte. Motivera varför! 2p

```
int sumAndMul(){  
    int sum = 0 ;  
    for( int i = 0 ; i < 10 ; i++){  
        sum += i;  
    }  
    return sum * i;  
}
```

3. Skriv en metod som givet en heltalsarray returnerar en array med alla partialsummor i arrayen. Exempel: 6p

Array	Array med partialsummor (resultatet)
[1, 1, 2]	[1, 2, 4]
[1, 2, 3, 4, 5]	[1, 3, 6, 10, 15]
[0, 0, 0]	[0, 0, 0]
[1, 2,-3]	[1, 3, 0]
[9]	[9]

4. Givet en rektangulär matris med positiva heltal. Skriv en metod som returnerar antalet unika rader i matrisen. Metoden skall använda arrayer för matriser. För full poäng krävs en lämplig funktionell nedbrytning. Skriv en kort kommentar vid varje metod, vad är syftet med metoden? Exempel: 12p

```
[ 1, 1, 1, 1  
 1, 2, 4, 5  
 1, 5, 1, 2 ]  ger svaret 3
```

```
[ 1, 1, 1, 1  
 1, 1, 1, 1  
 1, 5, 1, 2 ]  ger svaret 2
```

```
[ 5, 5, 5  
 5, 5, 5 ]  ger svaret 1
```

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, slutar och börjar på "aaa"

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

- Ta bort delsträngen från början av prydnadsordet
- 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. Se tillåtna metoder för String och StringBuffer i appendix.

b) Skriv en metod som givet en insträng skapar en prydnadssträng om insträngen är ett prydnadsord. Om ej returneras insträngen.

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. 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 skriva ett program för ett kortspel.

12p

- a) Skriv en klass Card för kort. Ett kort har en kortfärg och ett värde. Värde och färg skall kunna anges då kortet skapas.
- b) Skriv en klass Deck för hela kortleken. Leken skall ha en lista med kort och en metod, pick() som returnerar ett kort ur leken. Då kortleken skapas får den en lista med kort. Använd List/ArrayList för alla listor, inga arrayer, gäller även nedan. Se också appendix.
- c) Skriv en klass Player. En spelare har ett namn och en hand (en lista med kort) samt ett värde för max antal kort på hand (i listan). Namn och max antal kort anges då spelaren skapas. Klassen skall dessutom ha två metoder
 - i. hasFullHand som anger om spelaren har full hand (och inte kan lägga till fler kort)
 - ii. addCartToHand som lägger till ett kort till spelarens hand
- d) Skriv en klass CardGame som skall representera hela spelet. Klassen har en kortlek och en lista med spelare. Dessa anges då spelet skapas. Dessutom skall det finnas en aktuell spelare. Lägg till en metod pickCard som tar ett kort från leken och ger detta till aktuell spelare (om möjligt). Är det inte möjligt så byter spelet till nästa spelare.

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

8. Vad kommer att skrivas ut av programmet nedan? Motivera! Vad kommer programmet att skriva ut om man i tur och ordning avkommenterar raderna a)-c) nedan (vi återkommenterar efter varje körning)? Motivera varje fall. Om programmet inte kompilerar för något fall motivera detta.

8p

```
Sub s = new Sub();
s.m();
out.println(s.getI());
s.n();
out.println(s.getI());

public class Super {
    private int i;
    public void m() { i++; }
    public void n() {
        i += 2;
        //m();          // a)
        //this.m();      // b)
        //super.m();     // c)
    }
    public int getI() { return i; }
    public void setI(int i) { this.i = i; }
}

public class Sub extends Super {
    public void m() { setI(getI() + 5); }
}
```

APPENDIX

```
// Tillåtna metoder för uppg. 5
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.

// Tillåtna metoder för uppg. 7
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
```