

Tentamen för TDA540 Objektorienterad programmering

DAG: 15-04-18

TID: 14:00 – 18:00

Ansvarig: Joachim von Hacht och Christer Carlsson

Förfrågningar: Joachim von Hacht, 0707/311066
Christer Carlsson, ankn 1038

Resultat: erhålls via Ladok

Betygsgränser:

3:a	24 poäng
4:a	36 poäng
5:a	48 poäng
maxpoäng	60 poäng

Siffror inom parentes: anger maximal poäng på uppgiften.

Granskning: Måndag 11/5 kl 12-13 och tisdag 12/5 kl 12-13, rum 6128 i EDIT-huset.

Hjälpmedel: *Cay Horstmann: Java for everyone* eller
Jan Skansholm: Java direkt med Swing.
Understrykningar och smärre förtydligande noteringar får finnas.

Var vänlig och: Skriv tydligt och disponera papperet på lämpligt sätt.
Börja varje uppgift på nytt blad. Skriv ej på baksidan av papperet.

Observera: Uppgifterna är ej ordnade efter svårighetsgrad. Titta därför igenom hela tentamen innan du börjar skriva.

Alla program skall vara väl strukturerade, lätta att överskåda samt enkla att förstå. **Indentera programkoden!!**

Vid rättning av uppgifter där programkod ingår bedöms principiella fel allvarligare än smärre språkfel.

LYCKA TILL!!!!

Uppgift 1.

- a) Betrakta nedanstående program:

```
public class Uppgift1a {
    public static void main(String[] args) {
        int[] vekt = {5, 8, 3, 9, 4};
        mystery(vekt);
        System.out.println(java.util.Arrays.toString(vekt));
    } //main

    public static void mystery(int[] arr) {
        for (int i = 0; i < arr.length - 1; i = i + 1) {
            arr[i] = arr[i] + arr[i + 1];
            arr[i + 1] = arr[i] - arr[i + 1];
            arr[i] = arr[i] - arr[i + 1];
        }
    } //mystery
} //Uppgift1a
```

- i) Vad blir utskriften när programmet exekveras? (2 poäng)
- ii) Förklara med ord (inte mer än en mening) vad metoden `mystery` gör (= vilket syfte metoden har). (1 poäng)
- iii) Skriv om implementationen av metoden `mystery` så att koden blir enklare att förstå. (2 poäng)
- b) Vad skrivs ut när följande program exekveras?

```
public class AClass {
    private int x, y;
    private static int z;
    public AClass(int a, int b, int c) {
        x = a;
        y = b;
        z = c;
    }
    public String toString() {
        return x + ", " + y + ", " + z;
    }
} //AClass
```

```
public class TestClass {
    public static void main(String[] s) {
        AClass a = new AClass(1, 2, 3);
        AClass b = new AClass(4, 5, 6);
        System.out.println(a);
        System.out.println(b);
    }
} //TestClass
```

- c) Filmtokiga Stina håller på och skriver ett Javaprogram för att kunna lagra information om de filmer hon sett. Hon har betygsatt filmerna med en femgradig skala som heltal mellan 0 och 4. Nu vill hon dock, av någon anledning, även kunna uttrycka sin betygssättning med omdömena "Urdålig", "Dålig.", "Godkänd.", "Bra." och "Jättebra.". Hon har därför skrivit nedanstående metod:

```
public static String omvandla(int betyg) {
    if (betyg == 0)
        return "Urdålig!";
    else if (betyg == 1)
        return "Dålig!";
    else if (betyg == 2)
        return "Godkänd!";
    else if (betyg == 3)
        return "Bra!";
    else if (betyg == 4)
        return "Jättebra!";
} //omvandla
```

Det finns emellertid ett problem eftersom hon vid kompilering av metoden erhålls följande kompileringsfel:

```
missing return statement
```

Förklara varför felet uppkommer och skriv om metoden så den blir korrekt.

(2 poäng)

Uppgift 2.

Om man på kapitalet k erhåller räntan p %, har kapitalet efter t år vuxit till $s_t = k \cdot \left(1 + \frac{p}{100}\right)^t$

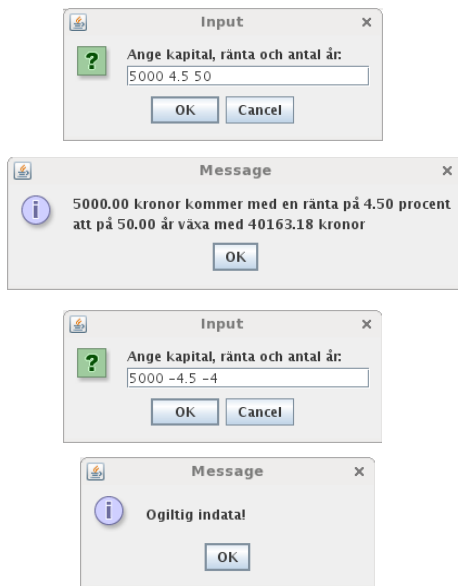
Din uppgift är att skriva ett program som upprepade gånger läser in startkapital (i kronor), ränta (i %) och antalet år, samt beräknar och skriva ut med hur startkapitalet mycket har vuxit.

Du får själv välja om du vill göra in- och utmatning via dialogrutor eller använda `System.in` respektive `System.out` (se exemplen nedan).

För att erhålla full poäng på uppgiften:

- skall programmet utformas på så sätt att inläsningen upprepas tills användaren avbryter exekveringen (vid användning av dialogrutor genom att användaren trycker på Cancel-knappen och vid användning av `System.in` genom att användaren lämpligen ger ctrl z)
- skall startkapital, ränta och antalet år läsas i en inläsningssats (dvs ett `Scanner`-objekt skall användas)
- skall programmet ge felutskrift om ett ogiltigt indata ges (dvs negativa värden, även på räntan trots att det numera är möjligt med negativ ränta)
- skall decimala tal i utskriften anges med *exakt* 2 decimaler
- skall det i programmet innehålla en metod
private static double increase(**double** capital, **double** rate, **double** year)
som beräknar hur mycket tillväxten på kapitalet blir

Med användning av dialogrutor



Med användning av System.in resp System.out

Ange kapital, räntesats och antal år: 5000 4.5 50
5000.00 kronor kommer med en ränta på 4.50 procent
att på 50.00 år växa med 40163.18 kronor

Ange kapital, räntesats och antal år: 5000 -4.5 -4
Ogiltig indata!

Ange kapital, räntesats och antal år:

(9 poäng)

Uppgift 3.

Skriv en metod

```
public static boolean containsAsSequence(int[] source, int[] target)
```

som tar två heltalsfält `source` och `target`. Metoden skall returnera **true** om elementen i `target` finns som en sekventiell följd av element i `source`.

Exempel:

Antag att följande deklaration har gjorts

```
int[] list1 = {1, 6, 2, 1, 4, 1, 2, 1, 8};
```

```
int[] list2 = {1, 2, 1};
```

```
int[] list3 = {2, 1, 2};
```

Anropet `containsAsSequence(list1, list2)` ger då värdet **true**, medan anropet `containsAsSequence(list1, list3)` ger värdet **false**.

Om parametern `source` eller `target` är **null** skall en `IllegalArgumentException` kastas.

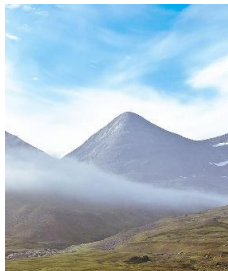
(6 poäng)

Uppgift 4.

En digital färgbild representeras som ett tvådimensionellt fält av bildpunkter, där varje bildpunkt utgörs av *tre* heltalsvärden i intervallet 0-255. De enskilda värdena i en bildpunkt representerar intensiteten av färgerna rött, grönt och blått.

En enkelt sätt att göra ett fotomontage med bra kvalitet är att fotografera objekt/föremålet man vill klistra in i ett annat fotografi, mot en enfärgad bakgrund.

I figurerna nedan har vi till vänster fotografiet med föremålet som vi vill klistra in i det mittersta fotografiet och till höger har vi resultatet.



Din uppgift är att skriva en metod

```
public static int[][][] montage(int[][][] foreground, int[][][] background)
```

som tar två digital färgbilder `foreground` samt `background` och returnerar en ny digital färgbild i vilken den enfärgade bakgrunden i bilden `foreground` har ersatts med motsvarande bildpunkter i bilden `background`. Du får anta att `foreground` och `background` har samma storlek. Vidare gäller att bakgrundsfärgen i bilden `foreground` är maximalt grön, dvs bildpunktens värde är [0, 255, 0].

(7 poäng)

Uppgift 5.

Skriv en metod som givet en sträng `s` och ett heltal `n` genererar en ny sträng med längden `n`, bestående av slumpvis valda tecken ur `s`. Om `s` är den tomma strängen sätts resultatet till den tomma strängen.

Du får använda `String`-metoderna som anges vid fråga 6.

(6 poäng)

Uppgift 6.

Vi vill översätta svenska meningar till "P-språket". Detta sker genom att alla vokaler dubbleras och ett 'p' läggs in mellan vokaler.

Example: "Hur är läget?" -> "Hupur äpär läpågepet?"

Skapa en klass PLanguage med följande metoder:

- a) **private boolean** isVowel(**char** ch) ger sant om ch är en vokal annars falskt. (2 poäng)
- b) **public** String toP(String swe) givet en sträng på svenska returneras översättningen till P-språket. Använd metoden från a) i din lösning. (4 poäng)

För att lösa uppgifterna är det tillåtet att använda följande metoder från klassen String:

- char** charAt(**int** i) ger tecknet vid index i.
- int** indexOf(**char** ch) ger index för tecknet ch, -1 om tecknet saknas.
- int** length() ger längden av strängen.
- String subString(**int** start, **int** end) ger en delsträng från start till end-1.
- String subString(**int** start) ger en delsträng från start till strängens slut.
- String[] split(String str) delar upp en sträng i ett fält av delsträngar utifrån ett visst tecken.
Exempel: "aaa:bb:cccc:dd".split(":") -> { "aaa", "bb", "cccc", "dd" }

Uppgift 7.

Skapa en objektmodell av spelet *Trivial Pursuit*, se bilaga.

- a) Ange 4 lämpliga klasser. (4 poäng)
- b) Hitta något lämpligt attribut (instansvariabel) för varje klass. (2 poäng)
- c) Skapa konstruktörer för varje klass och visa hur du m.h.a dessa kan koppla ihop hela modellen. Skapa en metod buildTP som utför arbetet och returnerar hela modellen. (4 poäng)

Uppgift 8.

Om man har två grupper av tal kan man ibland öka medelvärdet i båda grupperna genom att flytta ett tal från ena gruppen till den andra. Skriv en metod som givet två talgrupper A och B (i form av **int[]**) skriver ut vilket (om något) tal som skall flyttas vart.

Exempel:

Talgrupper

A = {3, 1, 2};

B = {4, 3, 4, 5};

Utskrift

Flytta 3 från B till A!

Vi antar att grupperna innehåller max 10 positiva heltal och att maxvärdet för talen är 100. Om ingen ökning av medelvärdet är möjligt skrivs "Omöjligt" ut.

(7 poäng)