Data och Informationsteknik / Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
Erland Holmström Göteborg 29 okt 2013

Övning 2

När du skriver funktioner (metoder) är det en sak du speciellt skall tänka på: funktioner skall vara "totala" dvs de skall hantera alla rätt typade argument. Ofta tänker man bara på de argument som normalt "fungerar". Se tex speciellt upp med tom sträng, negativa tal och noll som argument.

När man bearbetar fält vill man oftast gå igenom hela fältet. Då vet du alltså start och slut och skall välja en for sats. Men ibland, tex när man söker efter något, så vill man sluta i förtid och då är det antagligen lämpligare med en while. I samtliga fältuppgifter måste du undersöka om fältet är tomt.

Strängar och fält påminner om varandra, man kan nästan betrakta en sträng som ett fält med tecken, men de är inte riktigt lika. Fält finns inbyggda i Java och påminner också en del om de primitiva variablerna medan strängar är en äkta klass. Det är inte meningen att man skall lära sig alla metoder och klasser i APIn. Man måste kunna de vanligaste (tex substring, charAt och några fler i String).

Som vanligt kommer ni inte att hinna alla uppgifterna på en övning utan ni får fortsätta själva.

Övningar på strängar, vektorer, metoder och klasser.

1. Skriv ett program som frågar efter ett tal och sedan skriver ut vilket tecken talet hade dvs programmet skall skriva ut
   1 om talet > 0, 0 om talet = 0 och -1 om talet < 0.
2. Skriv funktionen `public static int sign(int tal)` som beräknar tecket på sitt heltals-argument. Och ett huvudprogram som använder sign.

$$\mathrm{sign}(x)=\begin{cases} 1 \text{ om } x>0 \\ 0 \text{ om } x=0 \\ -1 \text{ om } x<0 \end{cases}$$

3. a) Skriv ett predikat (en funktion som returnerar ett boeleskt värde)

   ```
   static boolean isVowel(char ch)
   ```

   som tar ett tecken som parameter, och returnerar true om det är en vokal, och false annars. Programmet skall klara både stora och små bokstäver. Enklast är att omvandla alla till små (eller stora) innan man testar om det är vokal.

   Man kan göra en lösning med en sträng och en while-loop. Man skapar då en sträng som innehåller alla vokaler som man sedan söker igenom med en loop. Om tecknet man skall undersöka finns i strängen är det en vokal, annars inte. (Man kan också använda en case sats.) Skriv också en klass som testar din metod.

   b) Skriv en metod

   ```
   public static int nbrOfVowels(String str)
   ```

   som räknar antalet vokaler i den givna strängen.

4. Skriv två funktioner `indexOfMaxElem` och `maxElement`, som tar en vektor som argument, och ger som resultat dels index för det största värdet i vektorn (om det finns flera så returneras index för det första) och dels det största värdet i vektorn

   Detta är två nästan identiska funktioner. Vad skall returneras om vektorn är tom?

5. Skriv en metod som givet en heltalsvektor skapar och returnerar en ny lika stor vektor som innehåller kvadraten på de tal som finns i indatavektorn. Om man alltså anropar metoden med vektorn {-1, 0, 1, 2, 3, 4, 5} som aktuell parameter så skall man få {1, 0, 1, 4, 9, 16, 25} tillbaka. Specifikationen kan se ut så här:

```
public static int[] arraySqr(int[] arr)
```

För och nackdelar om den ser ut så här?

```
public static void arraySqr(int[] arr)
```

6. Skriv en funktion `String toString4(int i)` som givet ett heltal i intervallet 0..999 returnerar en sträng med talet högerjusterat. Strängen skall alltid vara 4 tecken lång.

   Ex: `toString4(12)` skall ge "`..12`" som resultat dvs en stäng med 2 mellanslag (som jag indikerar med punkter, det kan du göra också om du vill) följt av tecknen 1 och 2. Om indatatalet inte är i intervallet 0..999 skall funktionen returnera talet som en sträng med en '*' först.

   Man kan använda en case sats men det blir otympligt, speciellt om man vill generalisera till större tal. Kan man använda en sträng med blanka ur vilken man tar en delsträng lika lång som antalet mellanslag som behövs? Titta gärna också lite på printf men tanken är inte att använda den här.

   Generalisera till `String toStringn(int i, int n)`som skapar sträng med n tecken.

7. Antag att du har en sträng med okänt innehåll. Det kan tex bero på att en användare just skrivit in strängen till en variabel. Nu vill du veta om bokstaven 'a' finns i strängen.
   a) Skriv ett program som tar reda på det och i så fall skriver ut på vilken plats i strängen tecknet först finns. Skriv först en version som använder metoden String.charAt och en loop.
   b) och sedan en version som använder String.indexOf. (kan ske i samma program)

   Exempel: Om enOkändSträng innehåller "Kalle har en dålig dag" så skall ditt program skriva ut en etta.

   c) Du vill dessutom kopiera de tre sista tecknen i strängen till en ny sträng med hjälp av String.substring. Gör det.

8. Skriv en metod som tar ett fält och ett tal *nbr* som parameter och skriver ut fältet med *nbr* tal per rad och i kolumner. Utskriften skall alltid inledas med en tom rad.

   Om *nbr* är 5 så skall alltså ett (endimensionellt) fält skrivas ut med 5 tal per rad enligt:

```
...1 ...2 ...6 ..24 .120
.720 .540 ..20 ...8 .880
```

   Långa fält kan "tabelleras" med `nbr` värden per rad genom att man lägger in

```
if ( (i+1)%nbr == 0 ) {
  System.out.println()
}
```

9. Skriv först två metoder som omvandlar mellan celsius och fahrenheit och omvänt.

```
public static double toCelsius(double x)
public static double toFahrenheit(double x)
```

   Omvandlingen sker med

```
fahrenheit = (9/5)*celsius +32
```

   Skriv sen ett program som upprepade gånger frågar användaren åt vilket håll han/hon vill omvandla. Svar skall ges med 'c' om man vill omvandla till Celsius och 'f' om man vill omvandla till fahrenheit grader. Om användaren skriver 's' vill han sluta och alla andra tecken skall ge en felutskrift och möjlighet att försöka igen. Efter att man skrivit 'c' eller 'f' så skall man uppmanas att mata in motsvarande gradtal och sedan skall programmet

skriva resultatet och fråga igen vad man vill göra

10. Den här uppgiften behöver inte redovisas men det är viktigt att du gör den.

Till Java hör ett stort klassbibliotek (kallat API = Application Programmers Interfaces) med mer än 1800 klasser) och ett än större antal metoder. Dokumentationen som beskriver hur klasserna används av en programmerare är tillgänglig på HTML-format från Oracle

http://docs.oracle.com/javase/6/docs/api/

Det finns länk på hemsidan så du slipper skriva. Lägg adressen till dina bokmärken.

När du labbar måste du ofta slå upp saker i APIn. Botaniosera på webben. Det är naturligt att inte förstå så mycket av det man ser/läser men titta på ändå.

Det i särklass viktigaste paketet är java.lang som innehåller klasser för in- och utmatning och några andra I java.lang finns det sk "wrapper" klasser för de primitiva typerna som tillhandahåller ett stort antal metoder för att tex omvandla mellan tal och text och en matteklass, Math, för diverse matematiska funktioner, tex absolutbelopp, avrundning och för att generera slumptal. Vill man använda något i klassen java.lang kan man göra det direkt – den är alltid tillgänglig.

java.lang.Math:

Starta din browser och studera paketet java.lang. Gå till klassen java.lang.Math. (Observera att det också finns ett paket som heter java.math men det är något annat.)

Skriv ner specifikationerna för hur man gör följande operationer

a) Genererar ett slumptal. Skriv ut ett slumptal.

b) Avrundar.

c) logaritmen (vilken logaritm är det?)

d) Antag att p är deklarerad som en int. Vad beräknar uttrycket och vilken typ har resultatet?

```
(int)Math.floor(Math.sqrt((double)p));
```

Svaret på fråga a) är tex (det finns fler möjligheter) Specifikation: static double random()

System.out.println(random()); // Returnerar en double som uppfyller 0.0 <= x < 1.0

java.lang.Character:

I den här klassen finns det ett flertal nyttiga metoder. Se tex på digit, isDigit, isLowerCase, isUpperCase, toLowerCase, toUpperCase.

e) Vad gör isLowerCase?

Double, Integer:

Här är framförallt compare, parseDouble, parseInteger och toString användbara.

f) Vilken typ har den statiska toString på sin parameter och vilken resultattyp har den?

g) Vad gör compare?

String:

Här finns ett många användbara metoder. På föreläsnings-OH finns en del förklarat.