

SI-pass 5

Johan Brook och Jesper Persson

2 september 2012

1. Swing och UI – begrepp och skillnader

Vi börjar med lite begrepp inom Swing-ramverket och GUI-programmering i Java, vilket är bra att ha koll på. Se till så att alla medlemmar i gruppen förstår!

Förklara för varandra vad följande begrepp är/innebär:

- JFrame
- JPanel
- AWT
- Component
- Container
- Event
- Listener

Vad är skillnaden mellan

- Swing och AWT?
- Programstyrning och händelsestyrning?
- FlowLayout och GridLayout?
- JFrame och JPanel?

2. Swing it med GridLayout!

Efter att ni är färdiga med att håna den oerhört lamska titeln till denna uppgift ska ni börja programmera ett simpelt gränssnitt med syftet att träna mer på Swing-ramverket och händelsehantering.

Att göra: Skriv en grafisk klass som har ett rutnät med dimensionen 3x3 av knappar med titlarna "Button 1" .. "Button 9". De ska alla ha olika bakgrundsfärger. När man klickar på en av knapparna ska en dialogruta öppnas och färgen ska skrivas ut som text i rutan.

Programmet **ska**

- ha ett fönster som är 300x300 pixlar
- ha ett fönster på positionen (600, 300)
- Ha fönstertiteln "Apple piskar"
- avslutas när man trycker på kryss-knappen i fönstret.

Saker att komma ihåg:

- Anropa metoderna `setVisible(true)` för att visa en `JFrame` och ev. `pack()`.
- För att få tillgång till en `JFrames` underliggande rityta att lägga element på kan man använda metoden `getContentPane()` som returnerar en `Container`.

Tips: För ett `ActionEvent`, använd metoden `getSource()` för att hämta upp den komponent som triggade eventet. `getSource()` returnerar ett objekt av typen `Object`.

3. Strängar, tal, och rekursion, rekursion, rekursion

Man kan ha otroligt mycket kul med strängar, tal, och rekursiva funktioner (faktiskt).

För varje uppgift, tänk på arbetsgången för att skriva rekursiva funktioner:

1. Hitta basfall!
2. Dela upp problemet i små delar och lös dem var för sig

Skriv kod för följande uppgifter:

a) En funktion som rekursivt vänder på ordningen av alla tecken i en given sträng.

Tips: Använd metoderna `length()`, `charAt(int index)`, och `substring(int start, int end)` för `String`-klassen.

b) En funktion som returnerar summan av ett tal, dvs. 10 för talet 4 (= 4+3+2+1 = 10).

c) En funktion som returnerar antal tecken i en given sträng, utan att använda `String.length()`.

Tips: Metoden `isEmpty()` i `String`-klassen returnerar `true` om strängen är tom.

d) Skriv Euklides algoritmen rekursivt.

Euklides algoritmen är en samling steg för att hitta den största gemensamma nämnaren av två givna tal. Algoritmen ("gcd") är beskriven så här:

- | | |
|--|----------------|
| 1. $\text{gcd}(a, 0) = a$ | <i>Basfall</i> |
| 2. $\text{gcd}(a, b) = \text{gcd}(b, a \bmod b)$ | $a, b > 0$ |

e) Skriv en funktion som beräknar det n:te Fibonacci-talet för ett givet heltal n.

1, 1, 2, 3, 5, 8, 13, 21 ...

Ett tal i Fibonacci-talföljden kan beräknas ganska snyggt med rekursion. Algoritmen ser ut så här:

1. $\text{fib}(0) = 0$
2. $\text{fib}(1) = 1$
3. $\text{fib}(n) = \text{fib}(n-1) + \text{fib}(n-2)$

Till exempel ska ett anrop `fib(3)` returnera 2.

Att diskutera:

1. Kan alla rekursiva funktioner skrivas iterativt (med loopar)?
2. Varför vill man använda rekursiva funktioner?
3. Välj en av deluppgifterna ovan och stega igenom tillsammans i gruppen för att få en förståelse vad som egentligen händer.
4. Fibonacci-uppgiften ovan (och säkert många andra) är ett dåligt exempel på användning av rekursion för att lösa ett problem. Varför? (*Tips: tänk på prestandan*).

4. Gammal tentauppgift – Addition med rekursion

Uppgiften nedan är tagen rakt av ur tentan från 2011 - klarar ni denna har ni 6 poäng rakt av! Wohoo!

Antag att din dator klarar av rekursion men inte vanlig addition. Den kan bara addera/subtrahera ett till ett annat (valfritt) tal. Om man skall addera $9+3$ måste man alltså låta datorn addera "1" tre gånger, dvs $((9+1)+1)+1$ eller $(9+3) = (10+2) = (11+1) = (12+0) = 12$.

Att göra: Skriv en rekursiv funktion som givet två heltal ≥ 0 returnerar deras summa. Om något av talen är negativt skall ett `IllegalArgumentException` kastas. Antalet rekursiva anrop skall minimeras, alltså skall $3+9$ och $9+3$ beräknas på samma sätt nämligen som $((9+1)+1)+1$ och inte

som

((((((((3+1)+1)+1)+1)+1)+1)+1)+1). Och absolut inte som $(9+(1+(1+1)))$.

5. Extra: skriv Fibonacci-funktionen från uppgift 3.e iterativt

