

# SI-möte #4, Programkonstruktion och datastrukturer

Elias Castegren  
elca7381@student.uu.se

24 november 2010

## Begrepp

- i)* Förvillkor (Precondition)
- ii)* Eftervillkor (Postcondition)
- iii)* Rekursionsvariant
- iv)* Defensiv programmering
- v)* Felkod (Exception)
- vi)* Testfall

## Övningar

### 1.

Ge lämpliga för- och eftervillkor (du behöver inte skriva några funktioner) för:

- i)* Ett program som returnerar 1 eller -1 för positiva respektive negativa heltal
- ii)* Ett program som beräknar  $n! = 1 \cdot 2 \cdot \dots \cdot (n - 1) \cdot n$
- iii)* Funktionen `size` i SML
- iv)* Funktionen `String.sub` i SML
- v)* Ett program som tar en sträng med siffror och returnerar motsvarande heltal

Skriv också testfall för varje funktion (enligt kursens riktlinjer).

### 2.

Skriv följande SML-funktioner med fullständiga funktions-specifikationer (skriv specifikationerna innan koden!).

- i)* `third(s)` som returnerar det tredje tecknet i strängen `s`
- ii)* `bool2int(b)` som för `true` och `false` returnerar 1 respektive 0
- iii)* `xor3(a, b, c)` som returnerar `true` om *exakt ett* av argumenten är `true`, annars `false`.
- iv)* `isVowel(c)` som ger `true` om tecknet `c` är en vokal, annars `false` (den inbyggda funktionen `Char.toUpperCase(c)` som returnerar tecknet `c` verserat kan vara användbar här)
- v)* `isMiddle(s, c)` som returnerar `true` om `c` är det mittersta tecknet i `s`, annars `false`

**3.**

Skriv en funktion `rovar(s) : string -> string` som översätter en sträng till rövarspråket. I rövarspråket dubbleras man varje konsonant och lägger in ett o mellan dubbletterna. Vokalerna lämnas oförändrade. T.ex. blir "Program" till "Poprorogororamom". Du får använda funktionen `isVowel(c) : char -> bool` som returnerar true om c är en vokal. Fundera på hur man kan använda lokala deklARATIONER till sin fördel. Skriv funktionsspecifikationer innan du skriver koden.

**4.**

Skriv en funktion `binary(n) : int -> string` (med fullständig funktionsspecifikation) som översätter ett naturligt tal n till en sträng som är talet uttryckt binärt.

**5.**

Antag att du har en funktion `foo(n) : int -> int` som beräknar den berömda (och påhittade) *foo-transformen* av n och som ger felkoden `FooCrit` för värden på n som inte uppfyller (det lika påhittade) *foo-kriteriet*. Skriv en funktion `bar(x) : int -> real` med funktionsspecifikation som beräknar kvadratroten ur *foo-transformen* av x och som returnerar 0.0 om x inte uppfyller *foo-kriteriet*.

---

*Lycka till!*