

Imperativ programmierung

1DL126

3p

Imperativ programmering

Jesper Wilhelmsson

ICQ: 20328079

Yahoo: amigajoppe

MSN / ePost: jesperw@it.uu.se

Rum: 1335

Tel: 471 1046

Imperativ programmering

Vilka programmeringsspråk kan du?

Imperativ programmering

"Haha.. Gammal hederlig imperativ programmering!"

Schema

	M	T	O	T	
■ v24	F		F	L	
■ v25	F		F	L	
■ v26	F	S	F	L	
■ v27	F		F	L	
■ v28	F	S	F	L	Föreläsningar i 1311
■ v29	F	S			Labbar i 1549D Seminarier i 1213

Kursupplägg

- Föreläsningar
 - 11 st. All viktig *information* som behövs för att klara kursen.
- Labbar
 - 5 st. Tillfällen att fråga. Labbtiden räcker inte. Kom förberedda!!
- Seminarier
 - 3 st. Examination genom diskussion.

38 timmar schemalagt av 120.

Kursmaterial

- "Programming languages", Ravi Sethi
- <http://www.it.uu.se/edu/course/homepage/impprog/st07>
- Wikipedia (engelsk!)
- Kompendium om C
- Tutorials och andra hjälpsidor på nätet
- Egna förslag

Redovisningsuppgifter

Uppgift 1

- Fem program skrivna i C, C++, Java, Ada och Pascal. Översätt alla programmen till de fem olika språken.
- Undersök språkkonstruktioner. Välj ett imperativt språk och ta reda på vad som är och inte är tillåtet. Hitta egenskaper hos språket som skiljer det från andra språk.
- Redovisas på seminarie 1.

Redovisningsuppgifter

Uppgift 2

- Egen valfri programmeringsuppgift. Några förslag på uppgift finns men ni är välkomna att föreslå egna uppgifter.
- Sammanfattning och stilredovisning på seminarie 1.
- Slutredovisning på seminarie 2.

Redovisningsuppgifter

Uppgift 3

- Ta ett projekt från SourceForge.net och studera källkoden.
 - Vilken paradig följer koden?
 - Går den att läsa?
 - Vad är bra, vad är mindre bra?
 - Vad skulle du göra annorlunda?
- Slutdiskussion på seminarie 3.

Paradigm

Programmeringsstil - olika sätt att se på programmet och dess syfte.

- *"Data som programmet arbetar med ska förvaras tillsammans med koden som arbetar på datat."*
- *"Ett anrop till en funktion ska vara oberoende av vad som sker runt omkring - samma indata ska alltid ge samma svar."*
- *"Programmet är en serie av kommandon som förändrar tillståndet i datorn."*
- *"Programmet består endast av relationer mellan olika objekt."*

Det var en gång...

... en vision (Leibniz, 1674)

1. Skapa ett universellt språk i vilket alla tänkbara problem kan formuleras.
2. Hitta en metod för att lösa alla problem som kan formuleras i det universella språket.

Det var en gång...

- Olöst i 250 år
- Oavgörbarhetsproblemet (Hilbert & Ackermann, 1928)

"Skapa ett datorprogram som tar som indata en definition av ett formellt språk och ett matematiskt uttryck beskrivet i språket. Programmet ska returnera sant eller falskt beroende på om uttrycket evaluerar till sant eller falskt."

Det var en gång...

- Allan Turing - Turing-maskinen
- Alonzo Church - λ -kalkylen

"Ett universellt språk..." - Nej, tyvärr!

Två olika sätt att se på problemet

Två olika sätt att tänka

Paradigmer

Religiösa krig...

Det finns ingen silverkula!

Ingen paradigm är bäst!
Inget programmeringsspråk är bäst!

Inte ens för ett enskilt problem!

Paradigmer och språk

Tre huvudgrupper

- Imperativa språk
 - Algol, Fortran, C, Pascal, Ada
- Funktionella språk
 - Haskell, Scheme, ML, Erlang, Excel
- Logiska språk
 - Prolog

Deklarativa språk

Funktionellt synsätt

"Värdet av ett uttryck beror endast av värdet av dess deluttryck."

- Förbjuder sidoeffekter - Samma indata ska alltid ge samma resultat.
- Funktioner är data.
- Rekursiv programmering.

Läs mer i Sethi, Programming Languages.

Logiskt synsätt

"Att programmera med relationer är mer flexibelt än att programmera med funktioner."

- Ingen skillnad på argument och resultat.
 - $5 + 7 = X$ Ja, om $X = 12$
 - $5 + X = 12$ Ja, om $X = 7$

Läs mer i Sethi, Programming Languages.

Paradigmer

- Funktionell programmering
 - Meddelandeprogrammering
 - Dataflödesprogrammering
- Logisk programmering
 - Constraint-programmering

Imperativt synsätt

- En serie instruktioner som förändrar omgivningens tillstånd.
- Iterativ programmering.
- Ett recept i en kokbok - Steg-för-steg-instruktioner som påverkar tillståndet i grytan. Den globala grytan innehåller efter att receptet är slutfört ett resultat som kan betraktas av andra.

Imperativa paradigmer

- Ostrukturerad programmering
- Strukturerad programmering
 - Procedurell programmering
 - Objektorienterad programmering
 - Klassbaserad programmering
 - Prototypbaserad programmering
 - Konceptorienterad programmering
 - Aspektorienterad programmering
 - Attributorienterad programmering
 - ... och så vidare ...

Paradigmer

Riktiga program är för komplexa.

T.ex. kompilator:

- Parsning - Imperativt
- Optimering - Funktionellt
- Registerallokering - Logikprogrammering
- Kodgenerering - Funktionellt

En paradigm räcker inte!

C++ stödjer många tanke sätt

- Procedurell programmering
- Objektbaserad programmering
- Objektorienterad programmering
- Agentorienterad programmering
- Generisk programmering

Common Lisp, Python, Ruby, Oz.

Imperativ hårdvara

- Hårdvaran styrs av maskinkod som skrivs i en imperativ stil.
- Datorns tillstånd - minnet - förändras av en serie instruktioner.
- Abstraktionslager ovanpå hårdvaran bygger mer och mer avancerade språk.

Historisk tillbakablick

I begynnelsen fanns det maskinkod
- den var imperativ...

- FORTRAN 1954
- B-0 (COBOL) 1957
- ALGOL 1958
- LISP 1958
- SNOBOL 1962
- CPL 1963
- BASIC 1964
- Simula I 1964
- Smalltalk 1969
- Pascal 1970
- Prolog 1970
- ML 1973
- Scheme 1975

Mot högre nivåer

Java	Erlang	Prolog
Ada	ML	
C	Pascal	BASIC
Assembler		
Hårdvara		

Obligatorisk uppgift

Skriv ner det viktigaste / det du minns bäst
från dagens föreläsning