

Introduktion till programmering

Programspråk och paradigmer

Vad är ett programspråk?

- "A **programming language** is a formal constructed **language** designed to communicate instructions to a machine, particularly a computer [...] to control the behavior of [the] machine or to express algorithms." – Google
- Syntax, semantik, implementation, bibliotek.

Exempel på icke-programspråk

- Skript-språk
 - Sh, Bash, Win32 batch, ...
- Query-språk
 - SQL, XQuery, XSLT, Datalog, ...
- Markup-språk
 - HTML, XML, JSON, ...

Hög-nivå vs låg-nivå

- Glidande skala: ju längre ifrån "processorn", som förstår binärkod, och minneshanteringen, desto högre nivå.
- Binärkod < Assembler (< C) < Hög-nivå-språk

Tolkat vs kompilerat

- En kompilator ("compiler") översätter ett program från en högre nivå till en lägre, ofta till assembler eller binärkod.
 - Kompilatorn körs en gång, det färdiga programmet kan köras många gånger.
 - Effektivt.
 - Behöver känna till datorarkitektur, processor mm. Ett program som kompilerats till en arkitektur kan inte flyttas till en annan.

Tolkat vs kompilerat

- En tolk ("interpreter") läser ett program på (typiskt) en hög nivå och exekverar dess kommandon "on the fly".
 - Tolken behöver tolka varje gång programmet körs.
 - Ineffektivt.
 - En tolk är i sig ett program, ofta skrivet i ett annat programspråk (ex Pythons tolk är skriven i C).

”Virtuell maskin” (VM)

- En virtuell maskin är i princip en tolk för ett lågnivå-språk.
 - Ex: Java kompileras till Java Bytecode, ett dedikerat lågnivå-språk (likt assembler).
 - En Java VM tolkar Java Bytecode.
 - Effektivare än att tolka Java direkt, mer portabelt än att kompilera till arkitektur-beroende assembler.

Statisk vs dynamisk typning

- Statisk typning: kontrollera att programmet använder rätt typer vid kompilering (eller inladdning)
 - Java, Haskell, C#, ML, ...
- Dynamisk typning: ge felmeddelande bara om programmet försöker använda data med fel typ
 - Python, JavaScript, Ruby

Statisk vs dynamisk *evaluering*

- Dynamiska språk (ej att förväxla med dynamiskt typade språk) tillåter skapande och exekverande av kod "on the fly".
 - Python, JavaScript, ...
- Utmärks typiskt av funktionen "eval"

Imperativt vs deklarativt

- Imperativa språk bygger på *kommandon* till datorn.
 - C, C++, C#, Java, ... (otaliga)
- Deklarativa språk deklarerar *betydelsen* av t ex funktioner, och låter tolk/kompilator hantera *hur* denna betydelse hanteras.
 - Funktionella eller logiska språk

Funktionella språk

- Funktioner är värden. De kan sparas i variabler och listor, användas som argument till funktioner, appliceras helt eller delvis.
 - Haskell, OCaml, SML, Scala, Clojure, ...
- Även andra språk stöder ofta funktioner som värden i någon utsträckning, men sällan bra.
 - Python, C#, Java 8, ...

Objekt-orientering

- Objekt är (bara) ett sätt att hantera data och dess beteende som en enhet, i en "klass".
 - Stöds av de flesta språk, t ex Python, Haskell...
- Objekt-orientering innebär ett sätt att tänka, där (i princip) allt i programmet modelleras med objekt.
- Objekt-orienterade *språk* är byggda så att allt är objekt från grunden. Kan vara imperativa (oftast) eller funktionella:
 - Java, C#, C++, Scala, Smalltalk, OCaml, ...

Prominenta programspråk I

- Java
 - Imperativt, objekt-orienterat, statiskt typat, hög-nivå. Dåligt stöd för funktionella mönster. Körs på VM.
- C#
 - I korthet, Microsofts version av Java. Lite bättre stöd för funktionella mönster. Körs på VM (.NET).
- C
 - Imperativt, "svagt" statiskt typat (går att kringgå), mid-nivå. Används typiskt vid system-nära programmering (t ex drivrutiner). Kompileras till binärkod. Bibliotek finns för allt.
- C++
 - Objekt-orienterad utökning av C. Används typiskt vid extremt performance-kritiska program (t ex grafikmotor till spel). Tusentals bibliotek.

Prominenta programspråk II

- Python
 - Imperativt, dynamiskt typat, har 'eval', hög-nivå. Tolkas (men finns även kompilatorer och VM). Tusentals bibliotek.
- Ruby
 - Likt Python, skillnader mest i syntax och implementation.
- JavaScript
 - Funktionellt(!), dynamiskt typat, har 'eval', hög-nivå. Tolkas (av browsers).
- PHP
 - Imperativt, dynamiskt typat, har 'eval', hög-nivå. Främst använt för hemsidor.

Prominenta programspråk III

- Haskell
 - Funktionellt, statiskt typat, kompileras eller tolkas. Det mest arketypiska funktionella språket. Separerar "pure" från "impure" kod.
- ML
 - Funktionellt, statiskt typat, kompileras eller tolkas. Separerar ej "pure" från "impure".
- OCaml
 - Objekt-orienterad variant av ML.
- Scala
 - Funktionellt, objekt-orienterat. Liknar Java, men med fullt utvecklat stöd för funktionell programmering. Körs på Java VM.