

Programmeringssprog

Datalogens Værktøj

Arinbjörn Brandsson - Arbr@di.ku.dk

Benjamin Rotendahl - Bero@di.ku.dk

Mathias Mortensen - Mamo@di.ku.dk

Datalogisk Institut



Hvad er Programmeringssprog?

Et Værktøj

- Programmeringssprog bruges som et værktøj til at løse en problemstilling
- Forskellige programmeringssprog bruges til forskellige formål

Et Regelsæt til Hvordan der Kommunikeres

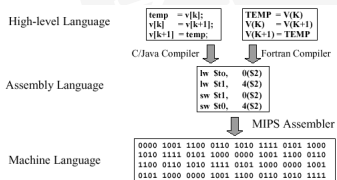
- Et programmeringssprog definerer hvordan man kan kommunikere med en datamat
- Uden programmeringssprog skulle vi kode i maskinssprog



Opbygning af et Programmeringssprog

Hvordan Kan Datamater Forstå Programmeringssprog?

- Programmeringssproget checkes for syntaksfejl
- Derefter køres den igennem dens sprog-specifikke oversætter, som oversætter koden til MIPS Assembly (Maskinsprog for mennesker)
- MIPS Assembly oversættes af datamaten om til binært som den kan forstå



Turing Tarpit

"Beware of the Turing tar-pit in which everything is possible but nothing of interest is easy."

-Alan Perlis, *Epigrams on Programming*

Turing Kompletthed

Et programmeringssprog er Turing komplet hvis alle (korrekte) algoritmer kan køre ved brug af programmeringssproget



Turing Tarpit Eksempel

Eksempel: "Hello, World!" i Brainfuck



Turing Tarpit Eksempel

Eksempel: "Hello, World!" i Brainfuck

```
+++++ +++           Set Cell #0 to 8
[
  >++++             Add 4 to Cell #1; this will always set Cell #1 to 4
  [               as the cell will be cleared by the loop
    >++           Add 2 to Cell #2
    >+++          Add 3 to Cell #3
    >+++          Add 3 to Cell #4
    >+            Add 1 to Cell #5
    <<<<<         Decrement the loop counter in Cell #1
  ]              Loop till Cell #1 is zero; number of iterations is 4
  >+             Add 1 to Cell #2
  >+             Add 1 to Cell #3
  >-             Subtract 1 from Cell #4
  >>+            Add 1 to Cell #6
  [<]            Move back to the first zero cell you find; this will
                  be Cell #1 which was cleared by the previous loop
  <-             Decrement the loop Counter in Cell #0
]               Loop till Cell #0 is zero; number of iterations is 8
>>.            Cell #2 has value 72 which is 'H'
>-..           Subtract 3 from Cell #3 to get 101 which is 'e'
+++++++...+++  Likewise for 'llo' from Cell #3
>>.            Cell #5 is 32 for the space
<-..           Subtract 1 from Cell #4 for 87 to give a 'W'
<-..           Cell #3 was set to 'o' from the end of 'Hello'
+++.....      Cell #3 for 'rl' and 'd'
>>+.           Add 1 to Cell #5 gives us an exclamation point
>++.           And finally a newline from Cell #6
```



Turing Tarpit - Fortsat

"Hello, World!" i C:



Turing Tarpit - Fortsat

"Hello, World!" i C:

```
#include<stdio.h>
```

```
main()
```

```
{
```

```
    printf("Hello, World");
```

```
}
```



Krav til et Godt Programmeringssprog

- Skal være læsbart og forståeligt
- Programmeringssprogets syntaks må ikke lede til tvetydigheder
 - For eksempel må der ikke være forvirring over hvad resultatet af " $2+4/2$ " er
- Skal have dokumentation over hvordan programmeringssproget virker og er struktureret



Programmeringssprogstyper (Imperative & Declarative)

Der Findes Mange Forskellige Typer af Programmeringssprog

- Object-Oriented programming
- Funktionel programming, og
- Logic programming

Og Disse Kan Underklassificeres om Hvordan Typer Håndteres

- Static/Dynamic typing
- Weak/Strong typing
- High-level/Low-level, og flere!

Disse beskrivelser af et programmeringssprog kaldes for paradigmer



Object-Oriented Programming



Functional Programming



Logic Programming

