

TDP019 Projekt: Datorspråk

Språkspecifikation

Författare

Albin Alvarsson, albal273@student.liu.se
Ludwig Holmberg, ludho208@student.liu.se

1 Revisionshistorik

Ver.	Revisionsbeskrivning	Datum
0.1	Första utkast inför handledning 2	14/2-2020

2 Språk

Vårt programmeringsspråk ska vara ett språk med fokus på hanteringen och användningen av tid. Det ska vara möjligt att skapa loopar som kör i en viss tid, till exempel från nu och 2 timmar och 32 minuter framåt, och loopar som kör under ett visst tidsintervall, till exempel mellan klockslagen 11:53:20 och 13:15:00. Det ska dessutom gå att specificera hur ofta koden i loopar ska köras, om koden i looparna ska köras så ofta den kan (som i andra språk) eller om koden endast ska köras varje sekund, tvåhundra millisekund, en gång varje 1,5h eller kanske en gång varje 3 minuter 24 sekunder och 73 millisekunder.

Tidsangivelseformatet ska kunna variera beroende på situation men generellt ska tid vid exakta tidsangivelser skrivas på ett 24-timmars format. När något ska genomföras under en viss tid kan det skrivas antingen som Xh Ym Zs eller X:Y:Z. Om något bara ska genomföras under en timme kan Y minuter och Z sekunder utelämnas vilket betyder att en for-loop "for 5:" kommer köras i 5 timmar. (Funktionen time() kan ändras vid uppkomsten av en bättre lösning).

Språket ska vara indenteringskänsligt utan semikolon vid radbryt.

3 Loopar

3.1 For-loopar

```
for 8:45:30 do
  ...
end

for 10:30:30 each 0:0:5 do
  ...
end
```

3.2 While-loopar

"while now <Y do"

3.3 From-loopar

From-loopen är en speciellt framtagen loop gjord för att enkelt kunna loopa under en tidsperiod.

```
from 14 to 15 do
  ...
end

from now to 19 do
  ...
end

from 10 to 12:30 each 0:15 do
  ...
end
```

4 Styrstrukturer

4.1 If-satser

4.2 Switch

4.3 At

En specialisering av if-satser gjord för att köra ett kodstycke en gång vid ett specifikt tillfälle. At-satsen kan i kod bytas ut mot "if now is X:" vilket uppfyller samma funktion men med mindre läslighet.

```
at 4:46 do
  ...
end
```

5 Scope & Referens/Kopia av komplexa datatyper

Eftersom vårt språk inte ska ändra mycket på basfunktioner som andra språk har kommer vårt språks scope fungera liknande andra språks. En funktion kan bara nå variabler den får inskickade till sig i parameterlistan och variabler från utomstående scope om de är markerade som globala variabler.

6 Frågetecken

- Ska våra variabler inledas med vilken datatyp den ska innehålla? (dvs ska vi ha $x = 5$ eller $\text{int } x = 5$)
- Ska vi i våra if-satser kolla om ett klockslag är i ett interval genom att säga (if $X \rightarrow Y \Leftrightarrow Z$) eller mer läsligt genom (if X is between Y and Z)
- Hur ska vi markera en global variabel? ($\$a$, $@a$, global a, eller på något annat sätt)