

TDP019 Projekt: Datorspråk

Språkspecifikation

Albin Dahlén, `albda746@student.liu.se`
Filip Ingvarsson, `filin764@student.liu.se`

Innehåll

1	Introduktion	2
2	Grammatik	2
3	Typning	2
4	Styrstrukturer	2
5	Iteratorer	3
5.1	For-loop	3
5.2	While-loop	3
5.3	Foreach-loop	3
6	Funktioner	3
6.1	Parameteröverföring	3
7	Objektorienterat	3
7.1	Klasser	4
8	Scope	4
8.1	Variablers livslängd	4

Revisionshistorik

Ver.	Revisionsbeskrivning	Datum
1.1	Språkspecifikation 2	23-02-22

1 Introduktion

2 Grammatik

```
program ::= stmt

stmt ::= | var_declaration
      | expr

var_declaration ::= | var, const
                  | expr

expr ::= | assign_expr
        | logical_expr
        |
```

3 Typning

Typningen ska vara statisk, där variabler deklareraras genom att specificera vilken typ det är.

Exempel:

int a = 2

float b = 2

string c = "Hej"

4 Styrstrukturer

Exempel if-sats:

```
if (cond) {

}
```

Exempel if-else-sats:

```
if (cond) {

} else {

}
```

Exempel if-elsif-elsesats:

```
if (cond) {

} elsif cond {

} else {
```

```
}
```

5 Iteratorer

5.1 For-loop

5.2 While-loop

5.3 Foreach-loop

6 Funktioner

Funktioner defineras genom att skriva nyckelordet *func* följt av vilket returtyp funktionen ska ha. Efter returtypen följer funktionsnamnet och en komma separerad parameterlista omgiven av paranteser där varje parameters typ anges innan parameterns namn. Funktionskroppen omslutes av en start- och slutmåsvinge. I slutet av funktionskroppen måste ett nyckelordet *return* finnas om det inte är en void funktion där nyckelordet *void* har skrivit i funktions deklarationen.

```
Exempel publik funktion:  
func string name(string param1, int param2...) {  
    statements  
    return "This is a string"  
}
```

6.1 Parameteröverföring

Alla parametrar skickas alltid som referenser och vill man ha en kopia får man anropa funktionen *copy* på objektet.

```
Exempel funktionsanrop med parameteröverföring:  
func_name(param1, param2)
```

```
Exempel funktionsanrop med parameteröverföring som kopior:  
object.func_name(param1.copy(), param2.copy())
```

7 Objektorienterat

Språket ska vara objektorienterat med arv där funktioner kan specificeras som public, protected och private.

```
Exempel publik funktion:  
func string name(string param1, int param2...) {  
    statements  
    return "This is a string"  
}
```

```
Exempel publik funktion:  
public func int name(int param1, int param2...) {  
    statements  
    return 45  
}
```

Exempel protected funktion:

```
protected func string name(string param1, int param2...) {  
    statements  
    return "This is a string"  
}
```

Exempel privat funktion:

```
private func string name(string param1, int param2...) {  
    statements  
    return "This is a string"  
}
```

7.1 Klasser

Exempel klass deklaration:

```
Class Name {  
  
}
```

Exempel klass deklaration med arv:

```
Class Name -> Parent {  
  
}
```

8 Scope

8.1 Variablers livslängd

En variabel där i samband med sitt scope.