



Introduction to CLU

A Prototype for Object-Oriented Programming

Developed by Barbara Liskov and her team

Nakarin Srijumrat

HS20 – Seminar: Turing Award Winners and Their Contributions

Overview

- Declaration and Assignment
- Collections
- Functions
- Classes (Clusters)

Declaration and Assignment

```
1    % declare a to be an integer variable
2    a: int
3
4    % declare first and last to be string variables
5    first, last: string
```

```
1    int: a := 20
2    real: ratio := (4/a)
3    revenue: int, string: company, ceo := "Apple", "Tim Cook"
```

```
1    a[i], a[j] := a[j], a[i] % swap array elements
2    real: a, b, c, d := 2x2matrix$getElements(A)
```

Collections

```
1    % create empty int array with size 20
2    array[int]: storage := array[int]$create(20)
3    storage[12] := 36
4
```

```
5    % sequences are immutable arrays
6    sequence[int]: order := sequence$new(4)
7
```

```
8    % records are collections of named variables
9    record[name, profession: string, age: int]: person
10   person.name := "Michael"
```

Functions

H

Body

```
1  convert_to_complex_number = proc (a, b: int)
2                                returns (complex)
3                                signals (no_imaginary_part)
4
5      if b = 0
6        then signal no_imaginary_part(a)
7        end
8
9      number.real := a
10     number.imag := b
11     return(number)
12     end set_complex_number
```

Classes(Clusters)

H

Body

```
1  complex = cluster is create , add , [...]  
2      number = struct [x, y: real]  
3  
4      create = proc(x,y: real) returns(cvt)  
5          return(number${x:x, y:y})  
6          end create  
7  
8      add = proc(z1 , z2: cvt) returns(cvt)  
9          return(number${x: z1.x + z2.x, y: z1.y + z2.y})  
10         end add  
11         [...]  
12     end complex
```

```
1 complex: z := complex$create(2, 3.5)  
2 complex: y := complex$create(1.2, 8)  
3 z := z$add(y)
```

Thank You!