

{ x :: Num, y :: Num }

String toString(Expr e) { ... }

```
String s = "abc";
```

```
s.length()
```

```
s.charAt(0)
```

```
JString sj = new JString();
```

```
s = sj;
```

```
class JString {  
    public int length() {  
        return 2;  
    }  
    public char charAt(int i) {  
        return 'a';  
    }  
    ...  
}
```

```
class C {
  int x;
}
```

```
class D {
  int x;
}
```

```
C c = new C();
D d = c;
```

Just class name?  
Any type?  
Expression?

Not just aliases

$C : \{x: \text{Int}\}$

$D :: \{x: \text{Int}\}$

$C @ \{f_1 :: \tau_1, \dots, f_n :: \tau_n\}$

$C :: C @ \{x :: \text{Int}\}$

new <expr> ()

Nominal Types  
Structural

type-of ( C ) = \_\_\_\_\_ → ClassInstance "C"  
 type-of (<expr>) = \_\_\_\_\_ → ClassInstance \_\_\_\_\_

class T-IPC:

...  
| t-class-instance(class-name::  
string)

end

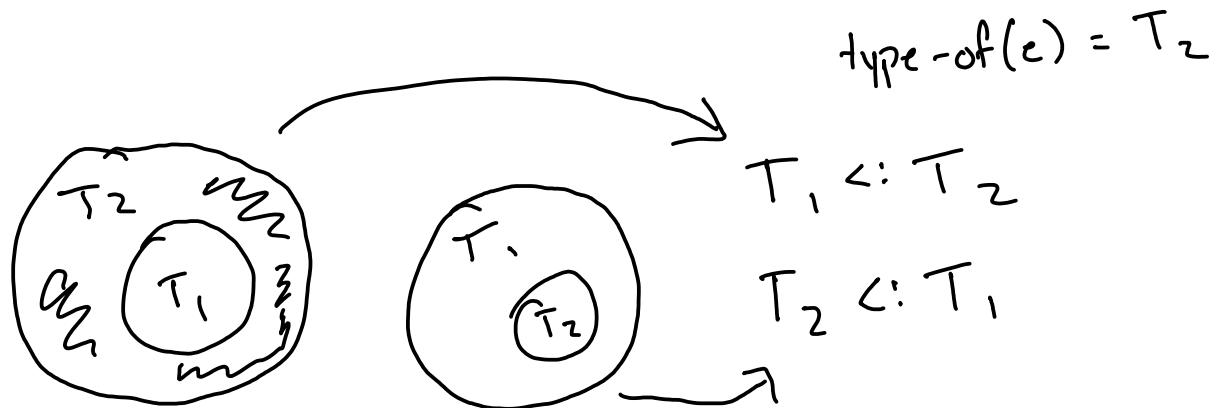
Class name      Class contents

C	{ x :: Int }
D	{ x :: Int }

E	{ }
---	-----

1. Subtyping?  $C \stackrel{?}{<} D$   $C \stackrel{?}{<} E$
2. Where to "put" inheritance
3. Interaction w/new construction

$T_1 \quad t = e$



class  $C_1$  extends  $C_2$ :

field  $f_1 :: T_{f_1}$   
⋮

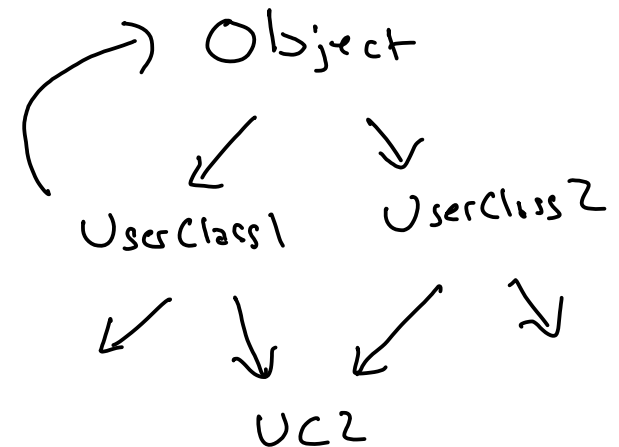
method  $m_1 :: (T_{a_1} \dots \rightarrow T_{r_1})$ : body end  
⋮

end

.data Program:

1 prog (classes :: List <Class>,  
main :: Expr)

end



1. Each " $C_1$  extends  $C_2$ " —  $C_2$  should exist

2. Each  $C$  is uniquely named

3. Is it a DAG? OK?

4. No cycles

