Compiler Practical 2013

Inheritance (Part 1)

Berthold Hoffmann (B. Gersdorf, T. Röfer)
hof@informatik.uni-bremen.de
Cartesium 2.48

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

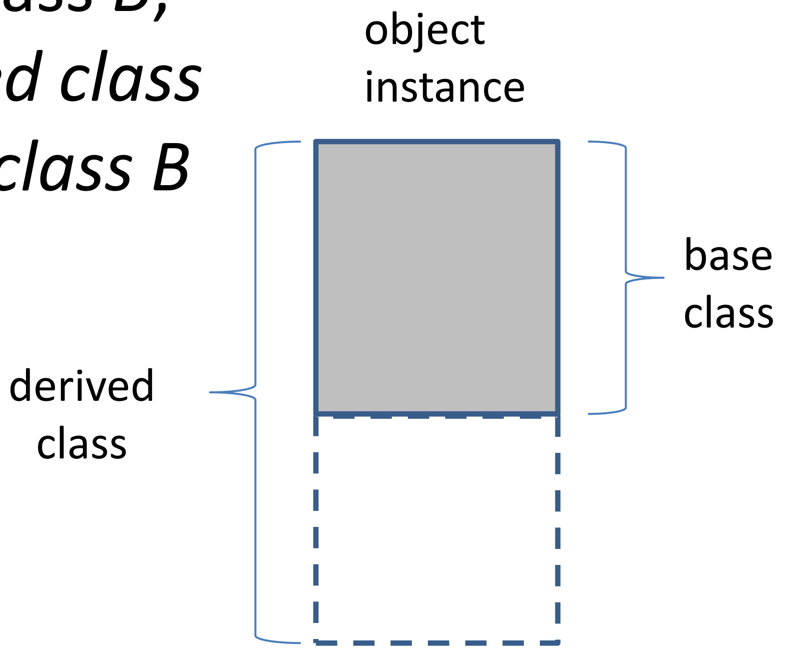
Universität Bremen

Structure

1. Inheritance and Derived Classes
2. Storage Organisation
3. Lexical and Syntax Analysis
4. Context Analysis
5. Bonus Task: Extension by Access Protection

Inheritance, Derivation

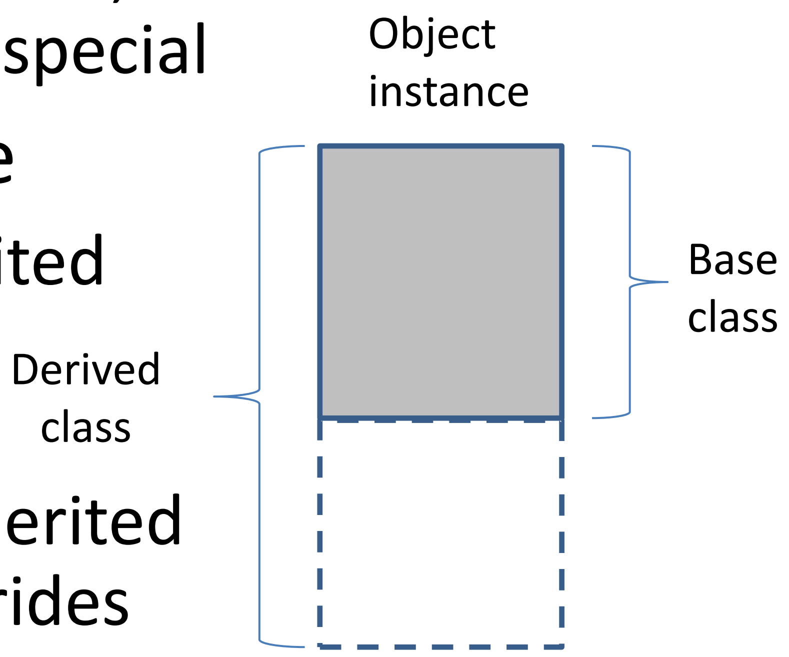
- Allows to model the *is-a* relation between Classes
  - If class *A* inherits from class *B*, *A* is the *subclass / derived class of the superclass / base class B*



- Goals
  - Reuse of code
  - Support of *polymorphism*

Virtual Methods (Polymorphism)

- Points of view
  - A derived class *extends* ist base class
  - A base class is more general, its derived class is more special
- Real /virtual inheritance
  - A method is *really* inherited if the derived class reuses the method
  - A method is *virtually* inherited if the derived class overrides the method.



Speicherorganisation

```
CLASS A IS
   c, d : Integer;
END CLASS

CLASS B EXTENDS A IS
   d, e : Boolean;
   METHOD f IS
      a : A;
      b : B;
   BEGIN
      b := NEW B;
      a := b;
   END METHOD
END CLASS
```

Address	Stack
R3-2	SELF
R3-1	return address
R3	predeccesor frame
R3+1	a
R3+2	b

Address	Heap
n	A.c
n+1	A.d
n+2	B.d
n+3	B.e

Storage extract with control flow at this place

Lexical and Syntax Analysis

- Lexical analysis
  - *EXTENDS*
- Syntax analysis
  - Extend grammar
  - *ClassDeclaration* needs a *baseType attribute*
  - Without *EXTENDS, Object is the base class*

```
classdecl ::=    CLASS identifier [ EXTENDS identifier ] IS
                 { memberdecl }
                 END CLASS
```

Context Analysis

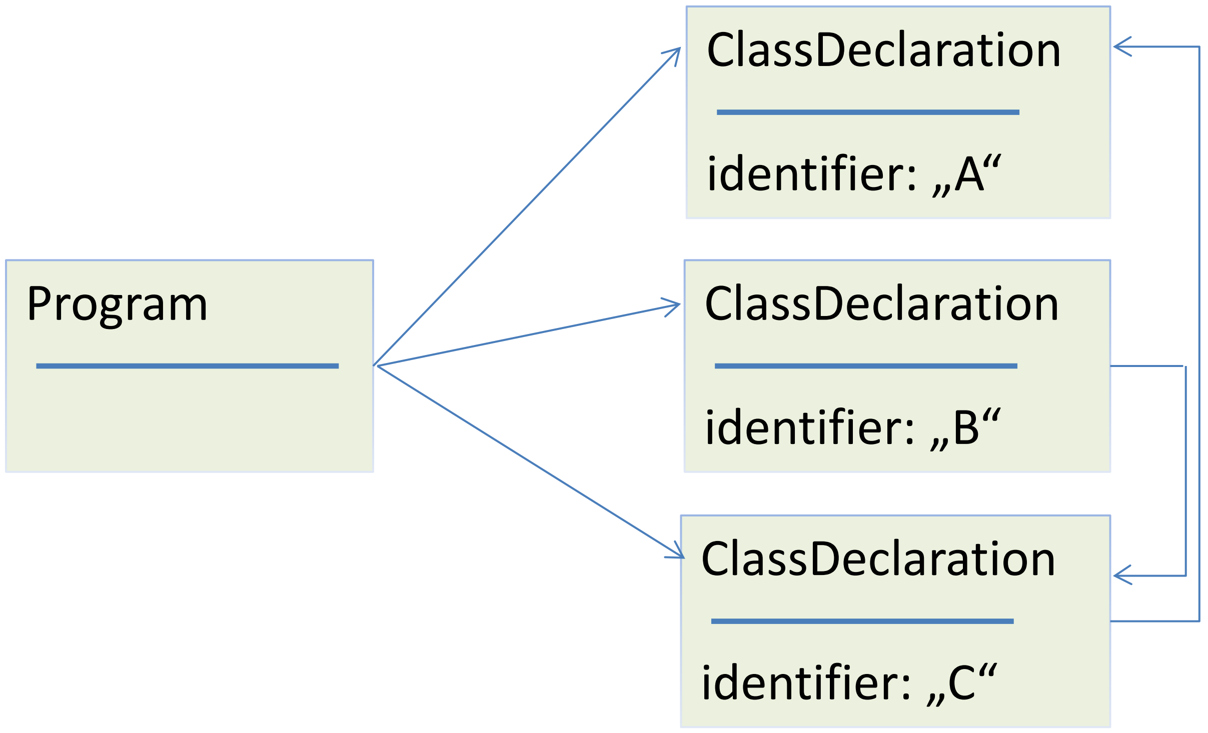
- Predefined Classes
  - New predefined class *Object*
  - The only class without a base class
  - Has neither attributes, nor methods (although it might have …)
  - *Integer* and *Boolean* inherit from *Object*
- Resolving base classes
  - Cycles are forbidden
  - Afterwards, *Program.classes* is an acyclic graph

Context Analysis, Subclasses

```
CLASS A IS
END CLASS

CLASS B
EXTENDS C IS
END CLASS

CLASS C
EXTENDS A IS
END CLASS
```



Context Analysis

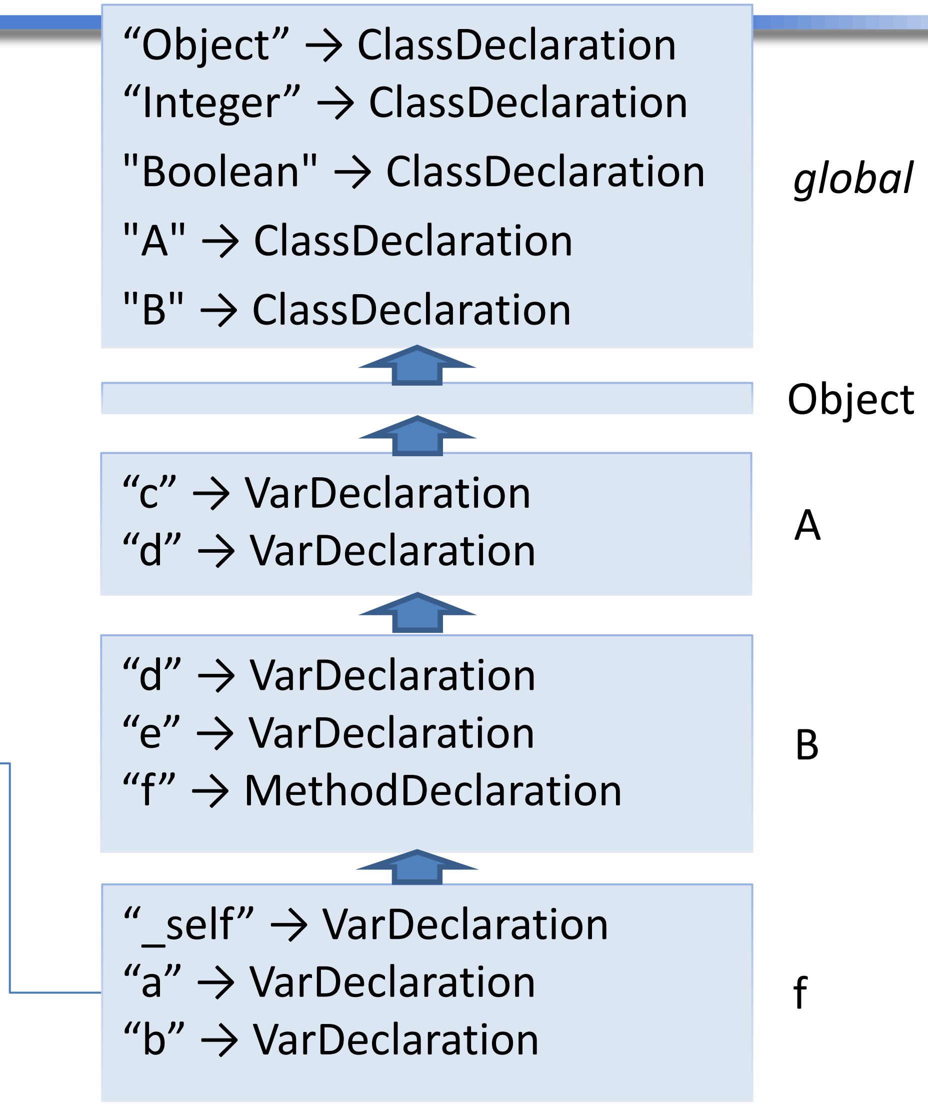
- Check base class before actual class
- Actual class inherits *Declarations* of the base class and extends them
  - *Object* „inherits“ visibility of class names
- Storage offset for attributes starts after the last offest for the base class
  - With *Object* , it starts after *HEADERSIZE*
- Extensions of *ClassDeclaration.isA(…)*

```
CLASS A IS
    c, d : Integer;
END CLASS
CLASS B EXTENDS A IS
    d, e : Boolean;
    METHOD f IS
        a : A;
        b : B;
    BEGIN
        b := NEW B;
        a := b;
    END METHOD
END CLASS
```

Management of Declarations

```
CLASS A IS
  c, d : Integer;
END CLASS

CLASS B EXTENDS A IS
  d, e : Boolean;
  METHOD f IS
    a : A; b: B;
  BEGIN
    b := NEW B;
    a := b;
  END METHOD
END CLASS
```



ClassDeclaration.isA(...)

- *a isA b*, if
  - *a* = *b,* or else
  - *a* = *nullType* AND *b isA objectClass*, or else
  - *a* # *objectClass* AND *a.baseType isA b*

```
CLASS B IS
END CLASS

CLASS C EXTENDS B IS
    METHOD f IS
        b : B;
    BEGIN
        b := NEW B;
        b := NULL;
        IF NULL THEN | Fehler
        END IF
        b := NEW C;
        b := NEW Integer; | Fehler
    END METHOD
END CLASS
```

Bonus Task: Access Protection

- Access protection
  - *PRIVATE*: access only within the class
  - *PROTECTED*: *PRIVATE* + access from derived classes
  - *PUBLIC*: access from everywhere (default)

```
CLASS Example IS
    PRIVATE internal : Integer;
    PUBLIC METHOD readonly: Integer IS
    BEGIN
        RETURN internal;
    END METHOD
END CLASS
```

- Class *Declarations*
  - Storing access rights with identifiers
  - *resolve(…)* needs class of access
  - Overriding must not restrict the access to a method

Bonus Task: Syntax

memberdecl ::= **[PRIVATE | PROTECTED | PUBLIC]**
(*vardecl* ';'
| METHOD *identifier* ['(' *vardecl* { ';' *vardecl* } ')']
[':' *identifier*] IS *methodbody* **)**

5%