CS 4110

Programming Languages & Logics

Lecture 30
Featherweight Java

12 November 2014

Object-Oriented Features

So far we've been mostly studying functional languages...

Object-Oriented Features

So far we've been mostly studying functional languages...

Today we'll study a core calculus developed by Igarashi, Pierce, and Wadler called *Featherweight Java*

Object-Oriented Features

So far we've been mostly studying functional languages...

Today we'll study a core calculus developed by Igarashi, Pierce, and Wadler called *Featherweight Java*

Featherweight Java is small by design: it has essential features including classes, inheritance, constructors, fields, methods, and casts, but omits everything else

Object-Oriented Features

So far we've been mostly studying functional languages...

Today we'll study a core calculus developed by Igarashi, Pierce, and Wadler called *Featherweight Java*

Featherweight Java is small by design: it has essential features including classes, inheritance, constructors, fields, methods, and casts, but omits everything else

Because the language is simple, its proof of type soundness is short and it is easy to extend

Question

Question

What *is* Object-Oriented Programming?

Syntax

$$
\begin{array}{rcll}
P & ::= & \overline{CL}\ e & \textit{programs} \\
CL & ::= & \mathtt{class}\ C\ \mathtt{extends}\ C\ \{\overline{C\,f};\ K\,\overline{M}\} & \textit{classes} \\
K & ::= & C(\overline{C\,f})\ \{\mathtt{super}(\bar{f});\ \overline{\mathtt{this}.f} = \bar{f};\ \} & \textit{constructors} \\
M & ::= & C\ m(\overline{C\,x})\{\mathtt{return}\ e\} & \textit{methods} \\
e & ::= & x & \textit{expressions} \\
 & | & e.f & \\
 & | & e.m(\overline{e}) & \\
 & | & \mathtt{new}\ C(\overline{e}) & \\
 & | & (C)\ e & \\
v & ::= & \mathtt{new}\ C(\overline{v}) & \textit{values} \\
E & ::= & [\cdot] & \textit{evaluation contexts} \\
 & | & E.f & \\
 & | & E.m(\overline{e}) & \\
 & | & v.m(\overline{v}, E, \overline{e}) & \\
 & | & \mathtt{new}\ C(\overline{v}, E, \overline{e}) & \\
 & | & (C)\ E &
\end{array}
$$

Example

```
class A extends Object { A() { super(); } }
class B extends Object { A() { super(); } }
class Pair extends Object {
  Object fst;
  Object snd;
  Pair(Object fst, Object snd) {
    super();
    this.fst = fst;
    this.snd = snd;
  }
  Pair swap Object() {
    return new Pair(this.snd, this.fst);
  }
}
new Pair(new A(), new B()).swap()
```

Subtyping

$$\frac{}{C \leq C}\ \text{S-Refl}$$

Subtyping

$$\frac{}{C \leq C}\ \text{S-Refl}$$

$$\frac{C \leq D \quad D \leq E}{C \leq E}\ \text{S-Trans}$$

Subtyping

$$\frac{}{C \leq C}\ \text{S-Refl}$$

$$\frac{C \leq D \quad D \leq E}{C \leq E}\ \text{S-Trans}$$

$$\frac{P(C) = \texttt{class}\ C\ \texttt{extends}\ D\ \{\overline{C\, f};\ K\, \overline{M}\}}{C \leq D}\ \text{S-Class}$$

Field Lookup

$$\frac{}{\mathit{fields}(\texttt{Object}) = []}\ \text{F-Object}$$

Field Lookup

$$\frac{}{\mathit{fields}(\texttt{Object}) = []}\ \text{F-Object}$$

$$\frac{P(C) = \texttt{class}\ C\ \texttt{extends}\ D\ \{\overline{C\,f};\ K\,\overline{M}\} \qquad \mathit{fields}(D) = \overline{D\,g}}{\mathit{fields}(C) = \overline{D\,g}\ \texttt{@}\ \overline{C\,f}}\ \text{F-Class}$$

Method Body Lookup

$$\frac{P(C) = \texttt{class}\ C\ \texttt{extends}\ D\ \{\overline{C\,f};\ K\,\overline{M}\} \qquad B\ m\ (\overline{B\,x})\ \{\texttt{return}\ e\} \in \overline{M}}{mbody(m, C) = (\bar{x}, e)}\ \text{MB-Class}$$

Method Body Lookup

$$\frac{P(C) = \texttt{class}\ C\ \texttt{extends}\ D\ \{\overline{C\,f};\ K\,\overline{M}\} \qquad B\ m\ (\overline{B\,x})\ \{\texttt{return}\ e\} \in \overline{M}}{mbody(m, C) = (\bar{x}, e)}\ \text{MB-Class}$$

$$\frac{P(C) = \texttt{class}\ C\ \texttt{extends}\ D\ \{\overline{C\,f};\ K\,\overline{M}\} \qquad B\ m\ (\overline{B\,x})\ \{\texttt{return}\ e\} \notin \overline{M}}{mbody(m, C) = mbody(m, D)}\ \text{MB-Super}$$

Operational Semantics

$$E ::= [\cdot] \mid E.f \mid E.m(\overline{e}) \mid v.m(\overline{v}, E, \overline{e}) \mid \texttt{new}\, C(\overline{v}, E, \overline{e}) \mid (C)\, E$$

Operational Semantics

$$E ::= [\cdot] \mid E.f \mid E.m(\overline{e}) \mid v.m(\overline{v}, E, \overline{e}) \mid \texttt{new}\ C(\overline{v}, E, \overline{e}) \mid (C)\ E$$

$$\frac{e \rightarrow e'}{E[e] \rightarrow E[e']}\ \text{E-Context}$$

Operational Semantics

$$E ::= [\cdot] \mid E.f \mid E.m(\overline{e}) \mid v.m(\overline{v}, E, \overline{e}) \mid \texttt{new}\ C(\overline{v}, E, \overline{e}) \mid (C)\ E$$

$$\frac{e \rightarrow e'}{E[e] \rightarrow E[e']}\ \text{E-Context}$$

$$\frac{\mathit{fields}(C) = \overline{C\ f}}{\texttt{new}\ C(\overline{v}).f_i \rightarrow v_i}\ \text{E-Proj}$$

Operational Semantics

$$E ::= [\cdot] \mid E.f \mid E.m(\overline{e}) \mid v.m(\overline{v}, E, \overline{e}) \mid \texttt{new}\ C(\overline{v}, E, \overline{e}) \mid (C)\ E$$

$$\frac{e \rightarrow e'}{E[e] \rightarrow E[e']}\ \text{E-Context}$$

$$\frac{\mathit{fields}(C) = \overline{C\ f}}{\texttt{new}\ C(\overline{v}).f_i \rightarrow v_i}\ \text{E-Proj}$$

$$\frac{\mathit{mbody}(m, C) = (\bar{x}, e)}{\texttt{new}\ C(\overline{v}).m(\overline{u}) \rightarrow [\overline{x} \mapsto \overline{u}, \texttt{this} \mapsto \texttt{new}\ C(\overline{v})]e}\ \text{E-Invk}$$

Operational Semantics

$$E ::= [\cdot] \mid E.f \mid E.m(\overline{e}) \mid v.m(\overline{v}, E, \overline{e}) \mid \texttt{new}\ C(\overline{v}, E, \overline{e}) \mid (C)\ E$$

$$\frac{e \rightarrow e'}{E[e] \rightarrow E[e']}\ \text{E-Context}$$

$$\frac{\mathit{fields}(C) = \overline{C\,f}}{\texttt{new}\ C(\overline{v}).f_i \rightarrow v_i}\ \text{E-Proj}$$

$$\frac{\mathit{mbody}(m, C) = (\bar{x}, e)}{\texttt{new}\ C(\overline{v}).m(\overline{u}) \rightarrow [\bar{x} \mapsto \overline{u}, \texttt{this} \mapsto \texttt{new}\ C(\overline{v})]e}\ \text{E-Invk}$$

$$\frac{C \leq D}{(D)\ \texttt{new}\ C(\overline{v}) \rightarrow \texttt{new}\ C(\overline{v})}\ \text{E-Cast}$$

Method Type Lookup

$$\frac{P(C) = \texttt{class}\ C\ \texttt{extends}\ D\ \{\overline{C\,f};\ K\,\overline{M}\} \qquad B\ m\ (\overline{B\,x})\ \{\texttt{return}\ e\} \in \overline{M}}{mtype(m, C) = \overline{B} \to B}\ \text{MT-Class}$$

Method Type Lookup

$$\frac{P(C) = \texttt{class}\; C\; \texttt{extends}\; D\; \{\overline{C\, f};\; K\, \overline{M}\} \qquad B\; m\; (\overline{B\, x})\; \{\texttt{return}\; e\} \in \overline{M}}{mtype(m, C) = \overline{B} \rightarrow B}\ \text{MT-Class}$$

$$\frac{P(C) = \texttt{class}\; C\; \texttt{extends}\; D\; \{\overline{C\, f};\; K\, \overline{M}\} \qquad B\; m\; (\overline{B\, x})\; \{\texttt{return}\; e\} \notin \overline{M}}{mtype(m, C) = mtype(m, D)}\ \text{MT-Super}$$

Typing Rules

$$\frac{\Gamma(x) = C}{\Gamma \vdash x : C}\ \text{T-Var}$$

Typing Rules

$$\frac{\Gamma(x) = C}{\Gamma \vdash x : C} \text{ T-Var}$$

$$\frac{\Gamma \vdash e : C \qquad \mathit{fields}(C) = \overline{C\, f}}{\Gamma \vdash e.f_i : C_i} \text{ T-Field}$$

Typing Rules

$$\frac{\Gamma(x) = C}{\Gamma \vdash x : C}\ \text{T-Var}$$

$$\frac{\Gamma \vdash e : C \qquad \mathit{fields}(C) = \overline{C\, f}}{\Gamma \vdash e.f_i : C_i}\ \text{T-Field}$$

$$\frac{\begin{array}{c}\Gamma \vdash e : C \qquad \mathit{mtype}(m, C) = \overline{B} \to B \\ \Gamma \vdash \overline{e} : \overline{A} \qquad \overline{A} \leq \overline{B}\end{array}}{\Gamma \vdash e.m(\overline{e}) : B}\ \text{T-Invk}$$

Typing Rules

$$\frac{\Gamma(x) = C}{\Gamma \vdash x : C} \text{ T-Var}$$

$$\frac{\Gamma \vdash e : C \qquad \mathit{fields}(C) = \overline{C\,f}}{\Gamma \vdash e.f_i : C_i} \text{ T-Field}$$

$$\frac{\begin{array}{c}\Gamma \vdash e : C \qquad \mathit{mtype}(m, C) = \overline{B} \to B \\ \Gamma \vdash \overline{e} : \overline{A} \qquad \overline{A} \leq \overline{B}\end{array}}{\Gamma \vdash e.m(\overline{e}) : B} \text{ T-Invk}$$

$$\frac{\mathit{fields}(C) = \overline{C\,f} \qquad \Gamma \vdash \overline{e} : \overline{B} \qquad \overline{B} \leq \overline{C}}{\Gamma \vdash \texttt{new}\ C(\overline{e}) : C} \text{ T-New}$$

Typing Rules

$$\frac{\Gamma(x) = C}{\Gamma \vdash x : C} \text{ T-Var}$$

$$\frac{\Gamma \vdash e : C \qquad \mathit{fields}(C) = \overline{C\,f}}{\Gamma \vdash e.f_i : C_i} \text{ T-Field}$$

$$\frac{\begin{array}{c}\Gamma \vdash e : C \qquad \mathit{mtype}(m, C) = \overline{B} \to B \\ \Gamma \vdash \overline{e} : \overline{A} \qquad \overline{A} \leq \overline{B}\end{array}}{\Gamma \vdash e.m(\overline{e}) : B} \text{ T-Invk}$$

$$\frac{\mathit{fields}(C) = \overline{C\,f} \qquad \Gamma \vdash \overline{e} : \overline{B} \qquad \overline{B} \leq \overline{C}}{\Gamma \vdash \texttt{new}\; C(\overline{e}) : C} \text{ T-New}$$

Typing Rules

$$\frac{\Gamma(x) = C}{\Gamma \vdash x : C} \text{ T-Var}$$

$$\frac{\Gamma \vdash e : C \qquad \mathit{fields}(C) = \overline{C\, f}}{\Gamma \vdash e.f_i : C_i} \text{ T-Field}$$

$$\frac{\begin{array}{c}\Gamma \vdash e : C \qquad \mathit{mtype}(m, C) = \overline{B} \to B \\ \Gamma \vdash \overline{e} : \overline{A} \qquad \overline{A} \leq \overline{B}\end{array}}{\Gamma \vdash e.m(\overline{e}) : B} \text{ T-Invk}$$

$$\frac{\mathit{fields}(C) = \overline{C\, f} \qquad \Gamma \vdash \overline{e} : \overline{B} \qquad \overline{B} \leq \overline{C}}{\Gamma \vdash \texttt{new}\, C(\overline{e}) : C} \text{ T-New}$$

Typing Rules

$$\frac{\Gamma(x) = C}{\Gamma \vdash x : C} \text{ T-Var}$$

$$\frac{\Gamma \vdash e : C \qquad \mathit{fields}(C) = \overline{C\,f}}{\Gamma \vdash e.f_i : C_i} \text{ T-Field}$$

$$\frac{\begin{array}{c}\Gamma \vdash e : C \qquad \mathit{mtype}(m, C) = \overline{B} \to B \\ \Gamma \vdash \overline{e} : \overline{A} \qquad \overline{A} \leq \overline{B}\end{array}}{\Gamma \vdash e.m(\overline{e}) : B} \text{ T-Invk}$$

$$\frac{\mathit{fields}(C) = \overline{C\,f} \qquad \Gamma \vdash \overline{e} : \overline{B} \qquad \overline{B} \leq \overline{C}}{\Gamma \vdash \texttt{new}\; C(\overline{e}) : C} \text{ T-New}$$

Method Typing

$$\frac{\mathit{mtype}(m, D) = \overline{A} \to A \text{ } \mathit{implies} \text{ } \overline{A} = \overline{B} \text{ and } A = B}{\mathit{override}(m, D, \overline{B} \to B)} \text{ Override}$$

Method Typing

$$\frac{\mathit{mtype}(m, D) = \overline{A} \to A \;\mathit{implies}\; \overline{A} = \overline{B} \text{ and } A = B}{\mathit{override}(m, D, \overline{B} \to B)}\ \text{Override}$$

$$\frac{\begin{array}{c}\overline{x : B}, \texttt{this} : C \vdash e : A \qquad A \leq B \\ P(C) = \texttt{class}\ C\ \texttt{extends}\ D\ \{\overline{C\,f};\ K\,\overline{M}\} \\ \mathit{override}(m, D, \overline{B} \to B)\end{array}}{B\ m(\overline{B\,x})\{\texttt{return}\ e\}\ \mathit{OK\ in}\ C}\ \text{Method-OK}$$

Class Typing

$$\dfrac{\begin{array}{c} K = C(\overline{D\,g}, \overline{C\,f})\ \{\texttt{super}(\overline{g});\ \overline{\texttt{this}.f = f};\ \} \\ \mathit{fields}(D) = \overline{D\,g} \qquad\qquad \overline{M}\ \mathit{OK\ in}\ C \end{array}}{\texttt{class}\ C\ \texttt{extends}\ D\ \{\overline{C\,f};\ K\,\overline{M}\}\ \mathit{OK}}\ \text{Class-OK}$$