

Nov 24, 16 21:32

## hw10-CLASSES-output-class-env

Page 1/8

```

1 [dshin@acacia classes]$ classes /home/pl/hw10/tests/colorpoint
2 Welcome to MzScheme v370 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.
3 > % =====
4 % inheritance example: colorpoint. (value=((6 8) (20 40) 87))
5
6 class point extends object
7   field x
8   field y
9   method initialize(initx,inity)
10    begin
11      set x=initx;
12      set y=inity
13    end
14   method move(dx,dy)
15    begin
16      set x+=(x,dx);
17      set y+=(y,dy)
18    end
19   method get-location() list(x,y)
20
21 class colorpoint extends point
22   field color
23   method set-color(c) set color=c
24   method get-color() color
25
26 let p=new point(3,4)
27 cp=new colorpoint(10,20)
28 in begin
29   send p move(3,4);
30   send cp set-color(87);
31   send cp move(10,20);
32   list(send p get-location(),
33        send cp get-location(),
34        send cp get-color())
35 end
36
37 % =====
38 ((colorpoint
39   #(struct:a-class
40     point
41     (x y color)
42     ({set-color
43       #(struct:a-method
44         (c)
45         #(struct:assign-exp color #(struct:var-exp c))
46         point
47         (x y color)))
48     (get-color
49      #(struct:a-method () #(struct:var-exp color) point (x y color)))
50     (initialize
51      #(struct:a-method
52        (initx inity)
53        #(struct:begin-exp
54          #(struct:assign-exp x #(struct:var-exp initx))
55          ( #(struct:assign-exp y #(struct:var-exp inity)))
56          object
57          (x y)))
58      (move
59       #(struct:a-method
60         (dx dy)
61         #(struct:begin-exp
62           #(struct:assign-exp
63             x
64             #(struct:sum-exp #(struct:var-exp x) #(struct:var-exp dx)))
65           ( #(struct:assign-exp
66             y

```

Nov 24, 16 21:32

## hw10-CLASSES-output-class-env

Page 2/8

```

67   #(struct:sum-exp #(struct:var-exp y) #(struct:var-exp dy))))))
68   object
69   (x y))
70   (get-location
71    #(struct:a-method
72      ()
73      #(struct:list-exp ( #(struct:var-exp x) #(struct:var-exp y)))
74      object
75      (x y))))))
76 (point
77   #(struct:a-class
78     object
79     (x y)
80     ((initialize
81       #(struct:a-method
82         (initx inity)
83         #(struct:begin-exp
84           #(struct:assign-exp x #(struct:var-exp initx))
85           ( #(struct:assign-exp y #(struct:var-exp inity)))
86           object
87           (x y)))
88       (move
89        #(struct:a-method
90          (dx dy)
91          #(struct:begin-exp
92            #(struct:assign-exp
93              x
94              #(struct:sum-exp #(struct:var-exp x) #(struct:var-exp dx)))
95            ( #(struct:assign-exp
96              y
97              #(struct:sum-exp #(struct:var-exp y) #(struct:var-exp dy))))))
98       object
99       (x y))
100     (get-location
101      #(struct:a-method
102        ()
103        #(struct:list-exp ( #(struct:var-exp x) #(struct:var-exp y)))
104        object
105        (x y))))))
106 (object #(struct:a-class #f () ()))
107 ((6 8) (20 40) 87)
108 >
109 [dshin@acacia classes]$ classes /home/pl/hw10/tests/dynamic
110 Welcome to MzScheme v370 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.
111 > % =====
112 % dynamic method dispatching. (value=12)
113
114 class interior-node extends object
115   field left
116   field right
117   method initialize(l,r)
118   begin
119     set left=l;
120     set right=r
121   end
122   method sum() +(send left sum()),send right sum())
123
124 class leaf-node extends object
125   field value
126   method initialize(v) set value=v
127   method sum() value
128
129 let ol=new interior-node(
130   new interior-node(
131     new leaf-node(3),
132     new leaf-node(4)),

```

```

133     new leaf-node(5))
134 in send o1 sum()
135
136 % =====
137 ((leaf-node
138   #(struct:a-class
139     object
140     (value)
141     ((initialize
142       #(struct:a-method
143         (v)
144         #(struct:assign-exp value #(struct:var-exp v))
145         object
146         (value)))
147       (sum #(struct:a-method () #(struct:var-exp value) object (value))))))
148   (interior-node
149     #(struct:a-class
150       object
151       (left right)
152       ((initialize
153         #(struct:a-method
154           (l r)
155           #(struct:begin-exp
156             #(struct:assign-exp left #(struct:var-exp l))
157             (#(struct:assign-exp right #(struct:var-exp r)))
158             object
159             (left right)))
160         (sum
161           #(struct:a-method
162             ()
163             #(struct:sum-exp
164               #(struct:method-call-exp #(struct:var-exp left) sum ())
165               #(struct:method-call-exp #(struct:var-exp right) sum ())
166               object
167               (left right))))))
168   (object #(struct:a-class #f () ())))
169 12
170 >
171 [dshin@acacia classes]$ classes /home/pl/hw10/tests/override
172 Welcome to MzScheme v370 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.
173 > % =====
174 % method overriding and self example. (value=(11 22 22))
175
176 class c1 extends object
177   method initialize() 1
178   method m1() 11
179   method m2() send self m1()
180
181 class c2 extends c1
182   method m1() 22
183
184 let o1=new c1()
185     o2=new c2()
186 in list(send o1 m1(),
187         send o2 m1(),
188         send o2 m2())
189
190 % =====
191 ((c2
192   #(struct:a-class
193     c1
194     ()
195     ((m1 #(struct:a-method () #(struct:const-exp 22) c1 ()))
196       (initialize #(struct:a-method () #(struct:const-exp 1) object ()))
197       (m1 #(struct:a-method () #(struct:const-exp 11) object ()))
198       (m2

```

```

199     #(struct:a-method
200       ()
201       #(struct:method-call-exp #(struct:self-exp) m1 ()))
202     object
203     ())))))
204 (c1
205   #(struct:a-class
206     object
207     ()
208     ((initialize #(struct:a-method () #(struct:const-exp 1) object ()))
209       (m1 #(struct:a-method () #(struct:const-exp 11) object ()))
210       (m2
211         #(struct:a-method
212           ()
213           #(struct:method-call-exp #(struct:self-exp) m1 ()))
214         object
215         ())))))
216   (object #(struct:a-class #f () ())))
217 (11 22 22)
218 >
219 [dshin@acacia classes]$ classes /home/pl/hw10/tests/shadow
220 Welcome to MzScheme v370 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.
221 > % =====
222 % field shadowing example. (value=(101 102 101 999))
223
224 class c1 extends object
225   field x
226   field y
227   method initialize() 1
228   method setx1(v) set x=v
229   method sety1(v) set y=v
230   method getx1() x
231   method gety1() y
232
233 class c2 extends c1
234   field y
235   method sety2(v) set y=v
236   method getx2() x
237   method gety2() y
238
239 let o2=new c2()
240 in begin
241   send o2 setx1(101);
242   send o2 sety1(102);
243   send o2 sety2(999);
244   list(send o2 getx1(), send o2 gety1(),
245         send o2 getx2(), send o2 gety2())
246 end
247
248 % =====
249 ((c2
250   #(struct:a-class
251     c1
252     (x y%1 y)
253     ((sety2
254       #(struct:a-method
255         (v)
256         #(struct:assign-exp y #(struct:var-exp v))
257         c1
258         (x y%1 y)))
259       (getx2 #(struct:a-method () #(struct:var-exp x) c1 (x y%1 y)))
260       (gety2 #(struct:a-method () #(struct:var-exp y) c1 (x y%1 y)))
261       (initialize #(struct:a-method () #(struct:const-exp 1) object (x y)))
262       (setx1
263         #(struct:a-method
264           (v)

```

Nov 24, 16 21:32 **hw10-CLASSES-output-class-env** Page 5/8

```

265      # (struct:assign-exp x # (struct:var-exp v))
266      object
267      (x y)))
268      (set!
269      # (struct:a-method
270      (v)
271      # (struct:assign-exp y # (struct:var-exp v))
272      object
273      (x y)))
274      (get! # (struct:a-method () # (struct:var-exp x) object (x y)))
275      (get! # (struct:a-method () # (struct:var-exp y) object (x y))))))
276      (c1
277      # (struct:a-class
278      object
279      (x y)
280      ((initialize # (struct:a-method () # (struct:const-exp 1) object (x y)))
281      (set!
282      # (struct:a-method
283      (v)
284      # (struct:assign-exp x # (struct:var-exp v))
285      object
286      (x y)))
287      (set!
288      # (struct:a-method
289      (v)
290      # (struct:assign-exp y # (struct:var-exp v))
291      object
292      (x y)))
293      (get! # (struct:a-method () # (struct:var-exp x) object (x y)))
294      (get! # (struct:a-method () # (struct:var-exp y) object (x y))))))
295      (object # (struct:a-class #f () ())))
296      (101 102 101 999)
297      >
298      [dshin@acacia classes]$ classes /home/pl/hw10/tests/simple
299      Welcome to MzScheme v370 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.
300      > % =====
301      % a simple object-oriented program. (value=((3 -3) (5 -5)))
302
303      class c1 extends object
304      field i
305      field j
306      method initialize(x)
307      begin
308      set i=x;
309      set j=- (0,x)
310      end
311      method countup(d)
312      begin
313      set i+= (i,d);
314      set j=- (j,d)
315      end
316      method getstate() list(i,j)
317
318      let t1=0
319      t2=0
320      ol=new c1(3)
321      in begin
322      set t1=send ol getstate();
323      send ol countup(2);
324      set t2=send ol getstate();
325      list(t1,t2)
326      end
327
328      % =====
329      ((c1
330      # (struct:a-class

```

Nov 24, 16 21:32 **hw10-CLASSES-output-class-env** Page 6/8

```

331      object
332      (i j)
333      ((initialize
334      # (struct:a-method
335      (x)
336      # (struct:begin-exp
337      # (struct:assign-exp i # (struct:var-exp x))
338      (# (struct:assign-exp
339      j
340      # (struct:diff-exp # (struct:const-exp 0) # (struct:var-exp x))))))
341      object
342      (i j)))
343      (countup
344      # (struct:a-method
345      (d)
346      # (struct:begin-exp
347      # (struct:assign-exp
348      i
349      # (struct:sum-exp # (struct:var-exp i) # (struct:var-exp d)))
350      (# (struct:assign-exp
351      j
352      # (struct:diff-exp # (struct:var-exp j) # (struct:var-exp d))))))
353      object
354      (i j)))
355      (getstate
356      # (struct:a-method
357      ()
358      # (struct:list-exp (# (struct:var-exp i) # (struct:var-exp j)))
359      object
360      (i j))))))
361      (object # (struct:a-class #f () ())))
362      ((3 -3) (5 -5))
363      >
364      [dshin@acacia classes]$ classes /home/pl/hw10/tests/super
365      Welcome to MzScheme v370 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.
366      > % =====
367      % super example. (value=172)
368
369      class point extends object
370      field x
371      field y
372      method initialize(initx,inity)
373      begin
374      set x=initx;
375      set y=inity
376      end
377      method move(dx,dy)
378      begin
379      set x+= (x,dx);
380      set y+= (y,dy)
381      end
382      method get-location() list(x,y)
383
384      class colorpoint extends point
385      field color
386      method initialize(initx,inity,initcolor)
387      begin
388      super initialize(initx,inity);
389      set color=initcolor
390      end
391      method set-color(c) set color=c
392      method get-color() color
393
394      let ol=new colorpoint(3,4,172)
395      in send ol get-color()
396

```

Nov 24, 16 21:32

## hw10-CLASSES-output-class-env

Page 7/8

```

397 % =====
398 (colorpoint
399   #(struct:a-class
400     point
401     (x y color)
402     ((initialize
403       #(struct:a-method
404         (initx inity initcolor)
405         #(struct:begin-exp
406           #(struct:super-call-exp
407             initialize
408             (#(struct:var-exp initx) #(struct:var-exp inity)))
409           (#(struct:assign-exp color #(struct:var-exp initcolor))))
410         point
411         (x y color)))
412     (set-color
413       #(struct:a-method
414         (c)
415         #(struct:assign-exp color #(struct:var-exp c))
416         point
417         (x y color)))
418     (get-color
419       #(struct:a-method () #(struct:var-exp color) point (x y color)))
420     (initialize
421       #(struct:a-method
422         (initx inity)
423         #(struct:begin-exp
424           #(struct:assign-exp x #(struct:var-exp initx))
425           (#(struct:assign-exp y #(struct:var-exp inity))))
426         object
427         (x y)))
428     (move
429       #(struct:a-method
430         (dx dy)
431         #(struct:begin-exp
432           #(struct:assign-exp
433             x
434             #(struct:sum-exp #(struct:var-exp x) #(struct:var-exp dx)))
435           (#(struct:assign-exp
436             y
437             #(struct:sum-exp #(struct:var-exp y) #(struct:var-exp dy))))
438         object
439         (x y)))
440     (get-location
441       #(struct:a-method
442         ()
443         #(struct:list-exp (#(struct:var-exp x) #(struct:var-exp y)))
444         object
445         (x y))))))
446 (point
447   #(struct:a-class
448     object
449     (x y)
450     ((initialize
451       #(struct:a-method
452         (initx inity)
453         #(struct:begin-exp
454           #(struct:assign-exp x #(struct:var-exp initx))
455           (#(struct:assign-exp y #(struct:var-exp inity))))
456         object
457         (x y)))
458     (move
459       #(struct:a-method
460         (dx dy)
461         #(struct:begin-exp
462           #(struct:assign-exp

```

Nov 24, 16 21:32

## hw10-CLASSES-output-class-env

Page 8/8

```

463       x
464       #(struct:sum-exp #(struct:var-exp x) #(struct:var-exp dx)))
465     (#(struct:assign-exp
466       y
467       #(struct:sum-exp #(struct:var-exp y) #(struct:var-exp dy))))))
468     object
469     (x y))
470   (get-location
471     #(struct:a-method
472       ()
473       #(struct:list-exp (#(struct:var-exp x) #(struct:var-exp y)))
474       object
475       (x y))))))
476 (object #(struct:a-class #f () ()))
477 172
478 >
479 [dshin@acacia classes]$

```