



Языки программирования. Семантика и система типов
Теоретическое задание. Тема 10

Бронников Егор

Задание 1. Реализуйте классы, соответствующие следующим императивным объектам на Welterweight Java. Можете использовать следующие определения `Unit` и `Nat`.

```
class Unit extends Object { }  
class Nat extends Object { }  
class Zero extends Nat { }  
class Succ extends Nat { Nat n; }
```

(a) `SetCounter`:

```
SetCounter =  
  { get : Unit -> Nat  
    , set : Nat -> Unit  
    , inc : Unit -> Unit  
  }
```

(b) `InstrCounter` (реализация метода `inc` не должна быть переопределена, вместо этого необходимо полагаться на открытую рекурсию):

```
InstrCounter =  
  { get : Unit -> Nat  
    , set : Nat -> Unit  
    , inc : Unit -> Unit  
    , accesses : Unit -> Nat  
  }
```

Решение. Задание (a).

```
class Unit extends Object { }
class Nat extends Object { }
class Zero extends Object { }

class Succ extends Nat {
    Nat n;

    Succ(Nat n) {
        this.n = n;
    }
}

class SetCounter {
    Nat value;

    SetCounter() {
        this.value = new Zero();
    }

    Nat get(Unit u) {
        return this.value;
    }

    Unit set(Nat n) {
        this.value = n;
        return new Unit();
    }

    Unit inc(Unit u) {
        this.value = new Succ(this.value);
        return new Unit();
    }
}
```

Решение. Задание (b).

```
class InstrCounter extends SetCounter {
    SetCounter accessCount;

    InstrCounter() {
        super();
        this.accessCount = new SetCounter();
    }

    Nat get(Unit u) {
        accessCount.inc(u);
        return super.get(u);
    }

    Unit set(Nat n) {
        accessCount.inc(new Unit());
        return super.set(n);
    }

    Unit inc(Unit u) {
        accessCount.inc(u);
        return super.inc(unit);
    }

    Nat accesses(Unit u) {
        return this.accessCount.get(u);
    }
}
```

Задание 2. Для следующего примера программы на Welterweight Java, выпишите следующие моменты динамической семантики:

- (а) начальную конфигурацию;
- (б) последовательность правил динамической семантики (только названия правил в корректном порядке), которые будут применены при вычислении;
- (с) конечную конфигурацию.

```
1  class Thunk extends Object { Object eval() { return this; } }
2  class Bool extends Thunk {
3      Object if(Object t, Object f) {
4          Object o; Bool b;
5          o = this.eval();
6          b = (Bool)o;
7          o = b.if(t, f);
8          return o;
9      }
10 }
11 class True extends Bool { Object if(Object t, Object f) { return t; } }
12 class False extends Bool { Object if(Object t, Object f) { return f; } }
13 class And extends Bool {
14     Bool l; Bool r;
15     Object eval() {
16         Object o; Bool b;
17         b = this.l;
18         f = new False();
19         r = this.r;
20         o = b.if(r, f);
21         return o;
22     }
23 }
24 class Main extends Object {
25     Object main() {
26         Bool t; Bool f; Bool x; Bool y; Object o;
27         t = new True();
28         f = new False();
29         x = new And();
30         x.l = t;
31         x.r = f;
32         y = new And();
33         y.l = x;
34         y.r = x;
35         o = y.eval();
36         return o;
37     }
38 }
```

Решение.

Задание (a).

Ответ: $H \mid < F, s > p_0$

Задание (b).

Ответ:

```
+ D-NEW - // t = new True()
+ D-NEW - // f = new False()
+ D-NEW - // x = new And()
+ D-UPDATE - // x.l = t
+ D-UPDATE - // x.r = f
+ D-NEW - // y = new And()
+ D-UPDATE - // y.l = x
+ D-UPDATE - // y.r = x
+ D-CALL - // o = y.eval()
  + D-UPDATE - // b = this.l
  + D-NEW - // f = new False()
  + D-UPDATE - // r = this.r
  + D-CALL - // o = b.if(r, f)
    + D-CALL - // o = this.eval()
      + D-UPDATE - // b = this.l
      + D-NEW - // f = new False()
      + D-UPDATE - // r = this.r
      + D-CALL - // o = b.if(r, f)
        + D-RETURN - // return t
      + D-RETURN - // reutrnr o
    + D-CAST - // b = (Bool)o
    + D-CALL - // o = b.if(t, f)
      + D-RETURN - // return f
    + D-RETURN // return o
  + D-RETURN - // return o
+ D-RETURN - // return o
+ D-FINISHED
```

Задание (c).

Ответ: *False*