

Post on functional programming in Java - maybe a tad verbose

[This is a good post](#) which tries to demonstrate the power of closures and functors.

It also helps illustrate how noisy things can be in Java. e.g.

```

import org.apache.commons.functor.*;
import org.apache.commons.functor.core.composite.*;
import org.apache.commons.functor.adapter.*;
import org.apache.commons.functor.UnaryFunction;
import org.apache.commons.functor.core.Constant;
import org.apache.commons.functor.core.IsEqual;
import org.apache.commons.functor.core.comparator.IsGreaterThanOrEqual;
import org.apache.commons.functor.core.comparator.Min;
import org.apache.commons.functor.core.Identity;

...

UnaryFunction getItemCat =
    new UnaryFunction()
    {
        public Object evaluate (Object obj)
        {
            return ((SETLItem)obj).getCategory();
        }
    };

UnaryFunction getItemPrice =
    new UnaryFunction()
    {
        public Object evaluate (Object obj)
        {
            return new Double(((SETLItem)obj).getPrice());
        }
    };

Constant catA = new Constant("A");
Constant usd200 = new Constant(new Double(200));

BinaryPredicateUnaryPredicate belongsToCatA = new
BinaryPredicateUnaryPredicate
    (new UnaryCompositeBinaryPredicate(new IsEqual(), getItemCat, catA));

BinaryPredicateUnaryPredicate moreThanUSD200 = new
BinaryPredicateUnaryPredicate
    (new UnaryCompositeBinaryPredicate(new IsGreaterThanOrEqual(), getItemPrice,
    usd200));

UnaryOr isEligibleForDiscount = new UnaryOr(new UnaryAnd(belongsToCatA,
moreThanUSD100),
    new UnaryAnd(belongsToCatB, moreThanUSD200));

if (isEligibleForDiscount.test(item1))
    System.out.println("Item #1 is eligible for discount!");
else
    System.out.println("Item #1 is not eligible for discount!");

```

whereas we could do something like this in Groovy

```
isEligibleForDiscount = { it.category == "A" && it.price > 200}
if (isEligibleForDiscount(item1) {
    println "Item #1 is eligible for discount!"
}
else {
    println "Item #1 is not eligible for discount!"
}
```

In IT you can usually use any tool to solve any problem. However sometimes switching tools makes things much easier & simpler.