

“Design Patterns as Higher-Order Datatype-Generic Programs”

Pop Quiz

Terminology

Define the following terms:
Design pattern ?

Higher-order ?

Datatype-generic ?

Iterators

Given the following

```
interface Iterator<T> {
    boolean hasNext();    // Are there more elements to iterate over?
    T      next();        // Actually obtain the next element of type T.
```

Implement a method to process elements,

```
default void iterate(java.util.function.Consumer consumer)
{
```

```
}
```

Either

Given

```
public class Either<A,B>
{
    private A left;
    private B right;

    private static enum Location {LEFT, RIGHT};
    private Location tag;

    public<R> R Match(Function<A,R> f, Function<B,R> g)
    {
        if (this.tag == Location.LEFT)
            return f.apply(left);
        else
            return g.apply(right);
    }
}
```

Implement an iterator,

```
class EitherIterator implements Iterator{
```

```
    public boolean hasNext(){
```

```
}
```

```
    public Object  next(){
```

```
}
```

```
}
```

Hint the latter two can be defined as two-liners ;-)