
Agile Techniques: Code Refactoring

Néstor Cataño

Extra Points – 01

Due: 15/Feb/2015

Code Duplication (10 Marks)

How would you handle code duplication in the example below? The code below is correct: it does implement the functionality that it was intended for. However, the code below is messy and cluttered. Your work is to write a much neater version of the code below that implements exactly the same functionality that was originally intended for.

What to Hand In?

- A Props.java file that contains a refactored version of the Props class implementation.

```
public class Props {
    public void getTimes(Properties props) throws Exception {
        String valueString;
        int value;

        valueString = props.getProperty("interval");
        if (valueString == null) {
            throw new MissingPropertiesException("monitor interval");
        }
        value = Integer.parseInt(valueString);

        if (value <= 0) {
            throw new MissingPropertiesException("monitor interval > 0");
        }
        checkInterval = value;

        valueString = props.getProperty("duration");
        if (valueString == null) {
            throw new MissingPropertiesException("duration");
        }
        value = Integer.parseInt(valueString);
        if (value <= 0) {
            throw new MissingPropertiesException("duration > 0");
        }
        if ((value % checkInterval) != 0) {
            throw new MissingPropertiesException("duration%checkInterval");
        }
        monitorTime = value;
    }
}
```

```
valueString = props.getProperty("departure");
if (valueString == null) {
    throw new MissingPropertiesException("departure offset");
}
value = Integer.parseInt(valueString);
if (value <= 0) {
    throw new MissingPropertiesException("departure > 0");
}
if ((value % checkInterval) != 0) {
    throw new MissingPropertiesException("departure % checkInterval");
}
departureOffset = value;
}
}
```