Code Quality

What is Clean Code?

"You know you are working on clean code when each routine you read turns out to be pretty much what you expected."

- Ward Cunningham

"Clean code can be read, and enhanced by a developer other than its original author."

- Dave Thomas

"Clean code always looks like it was written by someone who cares. There is nothing obvious that you can do to make it better."

- Michael Feathers

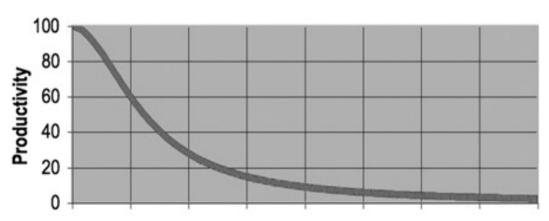
"Clean code is simple and direct. Clean code reads like well-written prose."

- Grady Booch

Technical Debt

"Doing things the quick and dirty way sets us up with a technical debt, which is similar to a financial debt. Like a financial debt, the technical debt incurs interest payments, which come in the form of the extra effort that we have to do in future development because of the quick and dirty design choice.

We can choose to continue paying the interest, or we can pay down the principal by refactoring the quick and dirty design into the better design. Although it costs to pay down the principal, we gain by reduced interest payments in the future."



http://martinfowler.com/bliki/TechnicalDebt.html Time
http://martinfowler.com/bliki/TechnicalDebtQuadrant.html
Clean Code ch1 / http://www.informit.com/articles/article.aspx?p=1235624
http://www.laputan.org/mud/

Refactoring

- "Refactoring is the process of changing a software system in such a way that it does not alter the external behavior of the code yet improves its internal structure."
 - -- Martin Fowler, Refactoring
- refactoring vs. Refactoring (a few seconds/minutes vs. days)
- DRY Don't Repeat Yourself
- Code Smells
- Keeping the code at top quality is required for writing code using TDD. Otherwise bad code will grind it to a halt. (see SOLID principles: http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod)
- A good regression test suite is needed. Run all tests after every change, to make sure that you broke nothing. Refactor only when all tests are green. Revert to last working state if you can't make the tests green – then try again with smaller steps.

Meaningful Names

- The name says what it is for, not what it is.
- Avoid encodings of any kind.
- Functions and variables have the same level of abstraction.
- Use pronounceable names.
- Shun names that disinform or confuse.
- Make context meaningful.
- Length of identifier matches scope.
- No lower-case L or upper-case o, ever.

```
int d; // elapsed time in days —
```

```
class DtaRcrd102 {
    private Date genymdhms;
    private Date modymdhms;
    private final String pszqint = "102";
```

```
int elapsedTimeInDays;
int daysSinceCreation;
int daysSinceModification;
int fileAgeInDays;

class Customer {
   private Date generationTimestamp;
```

private Date modificationTimestamp;

private final String recordId = "102";

Clean Code ch2 / http://tottinge.blogsome.com/meaningfulnames/ http://agileinaflash.blogspot.com/2009/02/meaningful-names.html http://butunclebob.com/ArticleS.TimOttinger.ApologizeIncode

Some Code Smells

- "In computer programming, code smell is any symptom in the source code of a program that possibly indicates a deeper problem."
 - Duplicated Code
 - Comments
 - Long Method
 - Large Class
 - Feature Envy
 - Divergent Change
 - Shotgun Surgery
 - Hard-to-Test Code
 - Fragile Test
 - Slow Tests etc.

Demo: (Big) Refactoring Example

Situation:

An application server used *java.math.BigInteger* as the ID for entity objects which are stored in the object database. This is the Primitive Obsession code smell. It will need to be replaced with a new class (*EntityId*), in order to make its intent clearer and future refactorings easier.

Codebase size:

4400 SLOC production code 7300 SLOC test code

Total time spent refactoring:

2 hours

(Demo: ~15 min, commit entityid-refactor~3 onwards)

1. Create target class

```
16
     public class ObjectId implements EntityId, Serializable {
17
          private static final long serialVersionUID = 1L;
18
19
          private final long id;
20
21
          public ObjectId(long id) {
22
              this.id = id;
23
24
25
          public boolean equals(Object obj) {
26
              if (!(obj instanceof ObjectId)) {
27
                  return false;
28
29
              ObjectId that = (ObjectId) obj;
              return this.id == that.id:
30
31
          }
32
33
          public int hashCode() {
34
              return (int) id:
35
          }
36
37
          public BigInteger toBigInteger() {
              return BigInteger.valueOf(id);
38
          }
39
40
          public String toString() {
41
              return getClass().getSimpleName() + "(" + id + ")";
42
43
44
45
```

2. Create temporary adapter class

```
public class ObjectIdMigration {
13
14
15
          private final BigInteger bi;
16
17
          public ObjectIdMigration(int signum, byte[] magnitude) {
18
              bi = new BigInteger(signum, magnitude);
19
20
21
          public ObjectIdMigration(byte[] val) {
22
              bi = new BigInteger(val);
23
24
25
          public ObjectIdMigration(long val) {
26
              bi = BigInteger.valueOf(val);
27
28
29
          public static ObjectIdMigration valueOf(long val) {
30
              return new ObjectIdMigration(val);
31
32
33
          public byte[] toByteArray() {
34
              return bi.toByteArray();
35
36
37
          public ObjectIdMigration add(ObjectIdMigration val) {
38
              return new ObjectIdMigration(val.bi.longValue() + 1);
39
          }
40
41
          public int hashCode() {
42
              return bi.hashCode();
43
44
45
          public boolean equals(Object obj) {
46
              return bi.equals(obj);
47
48
49
          public String toString() {
50
              return bi.toString();
51
          }
52
53
```

3. Migrate* all usages of the original class to the adapter (hundreds of usages)

| 20.5.2009 21:47 - EntityReference.java(Read-only) | | | | 13.8.2009 14:22 - EntityReference.java(Read-only) | | | | |
|---|---|----|--|---|---|--|--|--|
| ▲// Copyright © 2008-2009, Esko Luontola. All Right | | | | 1 | // Copyright © 2008-2009, Esko Luontola. All Right | | | |
| | // This software is released under the MIT License | | | 2 | // This software is released under the MIT License | | | |
| | // The license may be viewed at http://dimdwarf.so | 3 | | 3 | // The license may be viewed at http://dimdwarf.so | | | |
| | | 4 | | 4 | | | | |
| | <pre>package net.orfjackal.dimdwarf.api.internal;</pre> | 5 | | <pre>5 package net.orfjackal.dimdwarf.api.internal;</pre> | <pre>package net.orfjackal.dimdwarf.api.internal;</pre> | | | |
| | | 6 | | 6 | | | | |
| | import java.math.BigInteger; | 7 | | 7 | <pre>import net.orfjackal.dimdwarf.api.internal.ObjectI</pre> | | | |
| | | 8 | | 8 | | | | |
| | /** | 9 | | 9 | /** | | | |
| | * Reference to an entity, equivalent to Darkstar' | 10 | | 10 | * Reference to an entity, equivalent to Darkstar' | | | |
| | * | 11 | | 11 | * | | | |
| | * <u>@author</u> Esko Luontola | 12 | | 12 | * <u>@author</u> Esko Luontola | | | |
| | * <u>@since</u> 15.8.2008 | 13 | | 13 | * <u>@since</u> 15.8.2008 | | | |
| | */ | 14 | | 14 | */ | | | |
| | <pre>public interface EntityReference<t> {</t></pre> | 15 | | 15 | <pre>public interface EntityReference<t> {</t></pre> | | | |
| | | 16 | | 16 | | | | |
| | T get(); | 17 | | 17 | T get(); | | | |
| | | 18 | | 18 | | | | |
| | BigInteger getEntityId(); | 19 | | 19 | ObjectIdMigration getEntityId(); | | | |
| | } | 20 | | 20 | } | | | |
| | | 21 | | 21 | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | I | | | | | | |

^{*} IDEA 8: Refactor | Migrate

4. Add to the adapter class all methods and fields of the original class, until the project compiles and tests pass

```
import java.math.BigInteger;
                                                                                 public class ObjectIdMigration implements Serializable {
                                                                           15
                                                                                     public static final ObjectIdMigration ZERO = new ObjectIdMigration(0);
                                                                           16
* @author Esko Luontola
                                                                   10
                                                                           17
                                                                                     public static final ObjectIdMigration ONE = new ObjectIdMigration(1);
                                                                   11
* @since 13.8.2009
                                                                                     public static final ObjectIdMigration TEN = new ObjectIdMigration(10);
                                                                           18
                                                                   12
                                                                           19
                                                                   13
public class ObjectIdMigration {
                                                                                     public final BigInteger bi;
                                                                   14
                                                                           21
    private final BigInteger bi;
                                                                   15
                                                                           22
                                                                                     public ObjectIdMigration(int signum, byte[] magnitude) {
                                                                   16
                                                                           23
                                                                                         bi = new BigInteger(signum, magnitude);
    public ObjectIdMigration(int signum, byte[] magnitude) {
                                                                   17
                                                                           24
        bi = new BigInteger(signum, magnitude);
                                                                   18
                                                                           25
                                                                   19
                                                                           26
                                                                                     public ObjectIdMigration(byte[] val) {
                                                                   20
                                                                           27
                                                                                         bi = new BigInteger(val);
                                                                    21
    public ObjectIdMigration(byte[] val) {
                                                                           28
        bi = new BigInteger(val);
                                                                    22
                                                                           29
                                                                    23
                                                                           30
                                                                                     public ObjectIdMigration(long val) {
                                                                    24
                                                                           31
                                                                                         bi = BigInteger.valueOf(val);
   public ObjectIdMigration(long val) {
                                                                    25
                                                                           32
                                                                    26
        bi = BigInteger.valueOf(val);
                                                                           33
                                                                    27
                                                                           34
                                                                                     public static ObjectIdMigration valueOf(long val) {
                                                                    28
                                                                           35
                                                                                         return new ObjectIdMigration(val);
                                                                    29
    public static ObjectIdMigration valueOf(long val) {
                                                                           36
                                                                    30
        return new ObjectIdMigration(val);
                                                                           37
                                                                    31
                                                                           38
                                                                                     public byte[] toByteArray() {
                                                                    32
                                                                           39
                                                                                         return bi.toByteArray();
    public byte[] toByteArray() {
                                                                    33
                                                                           40
                                                                    34
        return bi.toByteArray();
                                                                           41
                                                                    35
                                                                           42
                                                                                     public ObjectIdMigration add(ObjectIdMigration that) {
                                                                    36
                                                                           43
                                                                                         return new ObjectIdMigration(this.bi.longValue() + that.bi.longValue());
    public ObjectIdMigration add(ObjectIdMigration val) {
                                                                    37
                                                                           44
        return new ObjectIdMigration(val.bi.longValue() + 1);
                                                                    38
                                                                           45
                                                                    39
                                                                           46
                                                                                     public int hashCode() {
                                                                    40
                                                                           47
                                                                                         return bi.hashCode():
    public int hashCode() {
                                                                    41
                                                                           48
        return bi.hashCode():
                                                                    42
                                                                           49
                                                                    43
                                                                           50
                                                                                     public boolean equals(Object obj) {
                                                                    44
                                                                           51
                                                                                         if (!(obj instanceof ObjectIdMigration)) {
   public boolean equals(Object obj) {
                                                                    45
                                                                           52
                                                                                              return false:
        return bi.equals(obj);
                                                                    46
                                                                           53
                                                                    47
                                                                           54
                                                                                         ObjectIdMigration that = (ObjectIdMigration) obj;
                                                                    48
                                                                           55
                                                                                         return bi.equals(that.bi);
    public String toString() {
                                                                    49
                                                                           56
                                                                    50
        return bi.toString();
                                                                           57
                                                                    51
                                                                           58
                                                                                     public String toString() {
                                                                    52
                                                                           59
                                                                                         return bi.toString();
                                                                           60
                                                                           61
                                                                           62
                                                                                     public int signum() {
                                                                           63
                                                                                         return bi.sianum():
                                                                           64
```

5. Put the target class next to the original class

```
public static final ObjectidMigration ONE = new ObjectidMigration(1);
                                                                                           public final Biginteger bigid:
public static final ObjectIdMigration TEN = new ObjectIdMigration(10)
                                                                         18
                                                                                 21
                                                                                           public final ObjectId objId:
                                                                         19
                                                                                 22
                                                                         20
                                                                                 23
                                                                                           public ObjectIdMigration(int signum, byte[] magnitude) {
public final BigInteger bigId;
                                                                                 24
                                                                         21
                                                                                               bigId = new BigInteger(signum, magnitude);
                                                                         22
                                                                                 25
public ObjectIdMigration(int signum, byte[] magnitude) {
                                                                                               obild = new ObjectId(bigId.longValue()):
                                                                         23
    bigId = new BigInteger(signum, magnitude);
                                                                                 26
                                                                         24
                                                                                 27
                                                                         25
                                                                                 28
                                                                                           public ObjectIdMigration(byte[] val) {
                                                                         26
                                                                                 29
public ObjectIdMigration(byte[] val) {
                                                                                               bigId = new BigInteger(val):
    bigId = new BigInteger(val);
                                                                         27
                                                                                 30
                                                                                               objId = new ObjectId(bigId.longValue());
                                                                         28
                                                                                 31
                                                                         29
                                                                                 32
public ObjectIdMigration(long val) {
                                                                         30
                                                                                 33
                                                                                           public ObjectIdMigration(long val) {
                                                                         31
    bigId = BigInteger.valueOf(val):
                                                                                 34
                                                                                               bigId = BigInteger.valueOf(val):
                                                                                 35
                                                                                               objId = new ObjectId(bigId.longValue());
                                                                         33
                                                                                 36
                                                                         34
public static ObjectIdMigration valueOf(long val) {
                                                                                 37
                                                                         35
    return new ObjectIdMigration(val);
                                                                                 38
                                                                                           public static ObjectIdMigration valueOf(long val) {
                                                                         36
                                                                                 39
                                                                                               return new ObjectIdMigration(val);
                                                                         37
                                                                                 40
public byte[] toByteArray() {
                                                                         38
                                                                                 41
    return bigId.toByteArray();
                                                                         39
                                                                                 42
                                                                                           public byte[] toByteArray() {
                                                                                 43
                                                                         40
                                                                                               return bigId.toByteArray();
                                                                                 44
                                                                         41
                                                                                 45
public ObjectIdMigration add(ObjectIdMigration that) {
                                                                         42
    return new ObjectIdMigration(this.bigId.longValue() + that.bigId
                                                                         43
                                                                                 46
                                                                                           public ObjectIdMigration add(ObjectIdMigration that) {
                                                                                 47
                                                                         44
                                                                                               return new ObjectIdMigration(this.bigId.longValue() + that.bigId
                                                                         45
                                                                                 48
                                                                                 49
public int hashCode() {
    return bigId.hashCode():
                                                                                 50
                                                                                           public int hashCode() {
                                                                         48
                                                                                 51
                                                                                               return objId.hashCode();
                                                                         49
                                                                                 52
public boolean equals(Object obj) {
                                                                         50
                                                                                 53
    if (!(obj instanceof ObjectIdMigration)) {
                                                                         51
                                                                                 54
                                                                                           public boolean equals(Object obj) {
        return false:
                                                                         52
                                                                                 55
                                                                                               if (!(obj instanceof ObjectIdMigration)) {
                                                                         53
                                                                                 56
                                                                                                   return false:
                                                                         54
                                                                                 57
    ObjectIdMigration that = (ObjectIdMigration) obj;
                                                                         55
    return bigId.equals(that.bigId);
                                                                                 58
                                                                                               ObjectIdMigration that = (ObjectIdMigration) obj;
                                                                         56
                                                                                 59
                                                                                               return bigId.equals(that.bigId);
                                                                         57
                                                                                 60
                                                                                 61
public String toString() {
    return bigId.toString();
                                                                                 62
                                                                                           public String toString() {
                                                                         60
                                                                                               return objId.toString();
                                                                                 63
                                                                         61
                                                                                 64
public int signum() {
                                                                         62
                                                                                 65
    return bigId.signum();
                                                                         63
                                                                                           public int signum() {
                                                                                 66
                                                                         64
                                                                                 67
                                                                                               return bigId.signum();
                                                                                 68
                                                                                 69
                                                                                           public BigInteger toBigInteger() {
                                                                                 70
                                                                                 71
                                                                                               return objId.toBigInteger();
                                                                                 72
                                                                                 73
```

6. Delegate original method to target method

```
public ObjectIdMigration add(ObjectIdMigration that) {
                                                                                                           public ObjectIdMigration add(ObjectIdMigration that) {
    return new ObjectIdMigration(this.bigId.longValue() + that.bigId.longValue()):
                                                                                          47
                                                                                                  47
                                                                                                                return next():
                                                                                                  48
                                                                                          49
                                                                                                  49
                                                                                          50
                                                                                                  50
                                                                                                            private ObjectIdMigration next() {
public int hashCode() {
                                                                                                  51
    return objId.hashCode();
                                                                                          51
                                                                                                                return new ObjectIdMigration(this.bigId.longValue() + 1);
                                                                                                  52
                                                                                          52
                                                                                                  53
                                                                                          53
                                                                                          54
                                                                                                  54
                                                                                                            public int hashCode() {
public boolean equals(Object obj) {
                                                                                          55
                                                                                                  55
    if (!(obj instanceof ObjectIdMigration)) {
                                                                                                                return objId.hashCode();
        return false:
                                                                                          56
                                                                                                  56
                                                                                          57
                                                                                                  57
                                                                                          58
                                                                                                  58
    ObjectIdMigration that = (ObjectIdMigration) obj;
                                                                                                            public boolean equals(Object obj) {
    return bigId.equals(that.bigId);
                                                                                          59
                                                                                                  59
                                                                                          60
                                                                                                  60
                                                                                                                if (!(obj instanceof ObjectIdMigration)) {
                                                                                          61
                                                                                                  61
                                                                                                                    return false:
public String toString() {
                                                                                          62
                                                                                                  62
                                                                                          63
                                                                                                  63
                                                                                                                ObjectIdMigration that = (ObjectIdMigration) obj;
    return objId.toString();
                                                                                          64
                                                                                                  64
                                                                                                                return objId.equals(that.objId);
                                                                                          65
                                                                                                  65
public int signum() {
                                                                                          66
                                                                                                  66
    return bigId.signum();
                                                                                          67
                                                                                                  67
                                                                                                           public String toString() {
                                                                                          68
                                                                                                  68
                                                                                                                return obild.toString():
                                                                                          69
                                                                                                  69
public BigInteger toBigInteger() {
                                                                                          70
                                                                                                  70
    return objId.toBigInteger();
                                                                                          71
                                                                                                  71
                                                                                                           public BigInteger toBigInteger() {
                                                                                          72
                                                                                                  72
                                                                                                                return objId.toBigInteger();
                                                                                          73
                                                                                                  73
                                                                                          74
                                                                                                  74
```

7. Inline* original method

```
public ObjectIdMigration add(ObjectIdMigration that) {
                                                                     46
                                                                             42
                                                                                       public byte[] toByteArray() {
    return next():
                                                                     47
                                                                             43
                                                                                           return bigId.toByteArray();
                                                                             44
                                                                     48
                                                                             45
                                                                     49
                                                                     50
private ObjectIdMigration next() {
                                                                             46
                                                                                       public ObjectIdMigration hext() {
    return new ObjectIdMigration(this.bigId.longValue() + 1);
                                                                     51
                                                                             47
                                                                                           return new ObjectIdMigration(this.bigId.longValue() + 1);
                                                                     52
                                                                             48
                                                                     53
                                                                             49
                                                                     54
                                                                             50
public int hashCode() {
                                                                                       public int hashCode() {
                                                                     55
                                                                             51
    return objId.hashCode();
                                                                                           return objId.hashCode();
                                                                     56
                                                                             52
                                                                             53
```

8.1 Repeat until the adapter and target classes have the same methods

```
ObjectIdMigration implements Serializable {
                                                                     14
                                                                                   import java.math.BigInteger:
                                                                     15
tatic final ObjectIdMigration ZERO = new ObjectIdMigration(0);
                                                                     16
                                                                             10
tatic final ObjectIdMigration ONE = new ObjectIdMigration(1);
                                                                     17
                                                                             11
                                                                                    * @author Esko Luontola
tatic final ObjectIdMigration TEN = new ObjectIdMigration(10);
                                                                     18
                                                                             12
                                                                                    * @since 13.8.2009
                                                                     19
                                                                             13
inal BigInteger bigId;
                                                                     20
                                                                             14
                                                                                   public class ObjectIdMigration implements Serializable {
inal ObjectId objId;
                                                                     21
                                                                             15
                                                                     22
                                                                             16
                                                                                       public final ObjectId objId;
ojectIdMigration(int signum, byte[] magnitude) {
                                                                     23
                                                                             17
d = new BigInteger(signum, magnitude);
                                                                     24
                                                                             18
                                                                                       public ObjectIdMigration(long val) {
d = new ObjectId(bigId.longValue());
                                                                     25
                                                                             19
                                                                                           objId = new ObjectId(val);
                                                                     26
                                                                             20
                                                                     27
                                                                             21
ojectIdMigration(byte[] val) {
                                                                     28
                                                                             22
                                                                                       private ObjectIdMigration(ObjectId objId) {
d = new BigInteger(val);
                                                                     29
                                                                             23
                                                                                           this.objId = objId;
d = new ObjectId(bigId.longValue());
                                                                     30
                                                                             24
                                                                     31
                                                                             25
                                                                     32
                                                                             26
                                                                                       public ObjectIdMigration next() {
ojectIdMigration(long val) {
                                                                     33
                                                                             27
                                                                                           return new ObjectIdMigration(this.objId.next());
= BigInteger.valueOf(val);
                                                                     34
                                                                             28
= new ObjectId(bigId.longValue());
                                                                     35
                                                                             29
                                                                     36
                                                                             30
                                                                                       public int hashCode() {
                                                                     37
                                                                             31
                                                                                           return objId.hashCode();
tatic ObjectIdMigration valueOf(long val) {
                                                                     38
                                                                             32
rn new ObjectIdMigration(val);
                                                                     39
                                                                             33
                                                                     40
                                                                             34
                                                                                       public boolean equals(Object obj) {
                                                                     41
                                                                             35
                                                                                           if (!(obj instanceof ObjectIdMigration)) {
/te[] toByteArray() {
                                                                     42
                                                                             36
                                                                                               return false:
rn bigId.toByteArray();
                                                                     43
                                                                             37
                                                                     44
                                                                                           ObjectIdMigration that = (ObjectIdMigration) obj;
                                                                             38
                                                                     45
                                                                             39
                                                                                           return objId.equals(that.objId);
ojectIdMigration next() {
                                                                     46
                                                                             40
rn new ObjectIdMigration(this.bigId.longValue() + 1);
                                                                     47
                                                                             41
                                                                     48
                                                                             42
                                                                                       public String toString() {
                                                                     49
                                                                             43
                                                                                           return objId.toString();
nt hashCode() {
                                                                     50
                                                                             44
rn objId.hashCode();
                                                                     51
                                                                             45
                                                                     52
                                                                             46
                                                                                       public BigInteger toBigInteger() {
                                                                     53
                                                                             47
                                                                                           return objId.toBigInteger();
oolean equals(Object obj) {
                                                                     54
                                                                             48
                                                                     55
                                                                             49
!(obj instanceof ObjectIdMigration)) {
                                                                     56
                                                                             50
                                                                     57
return false:
```

8.2 Repeat until the adapter and target classes have the same methods

```
tityId, Serializable {
                                     16
                                             16
                                                   public class ObjectId implements EntityId, Serializable {
lVersionUID = 1L:
                                     17
                                             17
                                                       private static final long serialVersionUID = 1L;
                                     18
                                             18
                                     19
                                             19
                                                       private final long id;
                                     20
                                             20
                                     21
                                             21
                                                       public ObjectId(long id) {
                                      22
                                             22
                                                            this.id = id:
                                     23
                                             23
                                     24
                                             24
                                     25
                                             25
j) {
                                                       public boolean equals(Object obj) {
Id)) {
                                      26
                                             26
                                                            if (!(obj instanceof ObjectId)) {
                                     27
                                             27
                                                                return false:
                                      28
                                             28
                                      29
obi;
                                             29
                                                            ObjectId that = (ObjectId) obj;
                                     30
                                             30
                                                            return this.id == that.id;
                                     31
                                             31
                                     32
                                             32
                                     33
                                             33
                                                        public int hashCode() {
                                     34
                                             34
                                                            return (int) id:
                                     35
                                             35
                                     36
                                             36
                                     37
                                             37
                                                        public BigInteger toBigInteger() {
                                     38
                                             38
d);
                                                            return BigInteger.valueOf(id);
                                     39
                                             39
                                                       }
                                     40
                                             40
                                             41
                                                       public String toString() {
                                     41
Name() + "(" + id + ")";
                                     42
                                             42
                                                            return getClass().getSimpleName() + "(" + id + ")";
                                     43
                                             43
                                     44
                                             44
                                     45
                                             45
                                                       // TODO: remove this method, use an external ID generator
                                                       public ObjectId next() {
                                             46
                                                            return new ObjectId(id + 1);
                                             47
                                             48
                                             49
                                             50
```

9. Migrate* all usages of the adapter class to the target (hundreds of usages)

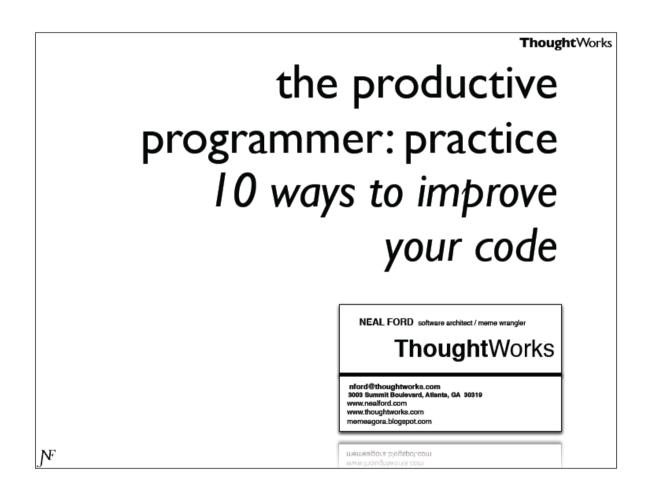
| 13. | 8.2009 14:22 - EntityReference.java(Read-only) | | 13.8.2009 15:02 - EntityReference.java(Read-only) | | |
|-----|--|----|---|---|--|
| | ▲// Copyright © 2008-2009, Esko Luontola. All Rights Reser | 1 | 1 | // Copyright © 2008-2009, Esko Luontola. All Rights Reser | |
| | // This software is released under the MIT License. | 2 | 2 | // This software is released under the MIT License. | |
| | // The license may be viewed at http://dimdwarf.sourcefor | 3 | 3 | // The license may be viewed at http://dimdwarf.sourcefor | |
| | | 4 | 4 | | |
| | <pre>package net.orfjackal.dimdwarf.api.internal;</pre> | 5 | 5 | <pre>package net.orfjackal.dimdwarf.api.internal;</pre> | |
| | | 6 | 6 | | |
| | <pre>import net.orfjackal.dimdwarf.api.internal.ObjectIdMigrat</pre> | 7 | 7 | <pre>import net.orfjackal.dimdwarf.api.internal.ObjectId;</pre> | |
| | | 8 | 8 | | |
| | /** | 9 | 9 | /** | |
| | * Reference to an entity, equivalent to Darkstar's Manag | 10 | 10 | * Reference to an entity, equivalent to Darkstar's Manage | |
| | * | 11 | 11 | * | |
| | * @author Esko Luontola | 12 | 12 | * @author Esko Luontola | |
| | * @since 15.8.2008 | 13 | 13 | * @since 15.8.2008 | |
| | */ | 14 | 14 | */ | |
| | <pre>public interface EntityReference<t> {</t></pre> | 15 | 15 | <pre>public interface EntityReference<t> {</t></pre> | |
| | | 16 | 16 | | |
| | T get(); | 17 | 17 | T get(); | |
| | , gott., | 18 | 18 | , gett | |
| | ObjectIdMigration getEntityId(); | 19 | 19 | ObjectId getEntityId(); | |
| - | } | 20 | 20 | 1 | |
| | | 21 | 21 | | |
| | | 21 | 21 | | |
| | | | | · | |

^{*} IDEA 8: Refactor | Migrate

10. Delete the adapter class and do final cleanups

```
package net.orfjackal.dimdwarf.api.internal;
                                                                         12
                                                                               * Unique ID for an entity of type {@link EntityObject}. Lat
import net.orfiackal.dimdwarf.api.EntityId;
                                                                         14
                                                                                * more entities, of which some are stored in the database :
                                                                         15
                                                                               * the entity objects of application code. Examples of such
import java.io.Serializable:
                                                                 10
import java.math.BigInteger;
                                                                         16
                                                                                * These will probably be stored in their own database, sepa
                                                                 11
                                                                        17
                                                                                * <0/>
                                                                 12
                                                                         18
                                                                                * TODO: When that happens, there will be need to be more ca
                                                                 13
  @author Esko Luontola
                                                                         19
                                                                                * It will be necessary to make a distinction between differ
 * @since 13.8.2009
                                                                 14
                                                                         20
                                                                                * implementation level, but to the application programmer
                                                                 15
                                                                         21
public class ObjectId implements EntityId. Serializable {
                                                                 16
                                                                         22
                                                                                * @author Esko Luontola
    private static final long serialVersionUID = 1L;
                                                                 17
                                                                         23
                                                                                * @since 13.8.2009
                                                                        24
                                                                 18
    private final long id;
                                                                 19
                                                                         25
                                                                              public class EntityObjectId implements EntityId, Serializabl
                                                                 20
                                                                                   private static final long serialVersionUID = 1L;
                                                                         26
    public ObjectId(long id) {
                                                                 21
                                                                         27
        this.id = id:
                                                                 22
                                                                         28
                                                                                   private final long id;
                                                                 23
                                                                         29
                                                                 24
                                                                         30
                                                                                   public EntityObjectId(long id) {
                                                                 25
    public boolean equals(Object obj) {
                                                                         31
                                                                                       this.id = id:
                                                                 26
        if (!(obj instanceof ObjectId)) {
                                                                         32
                                                                 27
            return false:
                                                                         33
                                                                 28
                                                                         34
                                                                                   public boolean equals(Object obj) {
        ObjectId that = (ObjectId) obj;
                                                                 29
                                                                         35
                                                                                       if (!(obj instanceof EntityObjectId)) {
        return this.id == that.id;
                                                                 30
                                                                         36
                                                                                           return false:
                                                                 31
                                                                         37
                                                                 32
                                                                                       EntityObjectId that = (EntityObjectId) obj;
                                                                 33
    public int hashCode() {
                                                                         39
                                                                                       return this.id == that.id:
                                                                 34
        return (int) id:
                                                                         40
                                                                 35
                                                                         41
                                                                 36
                                                                         42
                                                                                   public int hashCode() {
    public BigInteger toBigInteger() {
                                                                 37
                                                                         43
                                                                                       return (int) id:
        return BigInteger.valueOf(id);
                                                                 38
                                                                         44
                                                                 39
                                                                         45
                                                                 40
                                                                         46
                                                                                   public BigInteger toBigInteger() {
    public String toString() {
                                                                 41
                                                                         47
                                                                                       return BigInteger.valueOf(id);
        return getClass().getSimpleName() + "(" + id + ")";
                                                                 42
                                                                         48
                                                                 43
                                                                         49
                                                                 44
                                                                         50
                                                                                   public String toString() {
    // TODO: remove this method, use an external ID generate
                                                                 45
                                                                                       return getClass().getSimpleName() + "(" + id + ")";
                                                                         51
                                                                         52
    public ObjectId next() {
                                                                 46
        return new ObjectId(id + 1);
                                                                 47
                                                                         53
                                                                 48
```

10 Ways to Improve Your Code



http://www.infoq.com/presentations/10-Ways-to-Better-Code-Neal-Ford (59 min) http://qconsf.com/sf2008/file?path=/qcon-sanfran-2008/slides//NealFord_10_Ways_to_Improve_Your_Code.pdf

1. Composed Method

 "Divide your program into methods that perform one identifiable task. Keep all of the operations in a method at the same level of abstraction. This will naturally result in programs with many small methods, each a few lines long."

Benefits:

- shorter methods easier to test
- method names become documentation
- large number of very cohesive methods
- discover reusable assets that you didn't know were there

2. Test-Driven Design

- Benefits:
 - first consumer
 - think about how the rest of the world uses this class
 - creates consumption awareness
 - forces mocking of dependent objects
 - naturally creates composed method
 - cleaner metrics (e.g. cyclometric complexity)

3. Static Analysis

- 4. Good Citizenship
 - "How classes react to one another in a civilized society."
 - Example: static methods
 - good: stateless utility methods
 - bad: mixing state + static (e.g. the evil singleton pattern)

5. YAGNI – You Ain't Gonna Need It

- Discourages gold plating:
 - build the simplest thing that we need right now
 - don't indulge in speculative development
 - increases software entropy
 - only saves time if you can guarantee you won't have to change it later
 - leads to (the evil version of) frameworks

6. Question Authority

- angry monkeys: test names (camel case hard to read)
- non-intuitive: pair programming (15% slower, 15% fewer defects; real-time code review)

7. SLAP - Single Level of Abstraction Principle

- "Keep all lines of code in a method at the same level of abstraction."
 - jumping abstraction layers makes code hard to understand
 - composed method → SLAP
 - refactor to slap, even if it means single-line methods

8. Polyglot Programming

 "Leveraging existing platforms with languages targeted at specific problems and applications."

9. Every Nuance

 "If you are going to spend time primarily in a language, it makes sense to learn all the nuances of that language – all the little back alleys and strange little places where you hardly ever go."

10. Anti-Objects

- "The metaphor of objects can go too far by making us try to create objects that are too much inspired by the real world."
- "Take something that appears to be the opposite of what you think you should be writing code about, and see if that yields a simpler solution to your problem."
 - "Pacman smell": intelligence not in the ghost, but in the maze

Course Material

- Clean Code, chapter 1: Clean Code
 http://www.informit.com/articles/article.aspx?p=1235624
- Clean Code, chapter 2: Meaningful Names
 ≈ http://tottinge.blogsome.com/meaningfulnames/
 ≈ http://agileinaflash.blogspot.com/2009/02/meaningful-names.html
- http://martinfowler.com/bliki/TechnicalDebt.html
- http://c2.com/cgi/wiki?DontRepeatYourself
- http://butunclebob.com/ArticleS.TimOttinger.ApologizeIncode
- http://www.codinghorror.com/blog/archives/000589.html
- http://xunitpatterns.com/Test%20Smells.html
- http://www.infoq.com/presentations/10-Ways-to-Better-Code-Neal-Ford