NEW! STRONGER THAN DIRT

ENTER.
tech with a heartbeat

{Clean Code}

CAUTION: EVEN BAD CODE CAN FUNCTION
40 FL OZ (1.25 QT) 1.18L

INTRODUCTION

- Comments are used to compensate for our failure to express ourselves in code.
- They are always failure.
- Only the code can tell the truth of what it is doing.

ENTER.
tech with a heartbeat

COMMENTS DO NOT MAKE UP FOR BAD CODE

- Bad code is a good motivation for writing comments.
- Clear and expressive code with a feel comments is far superior than complex and commented code.

GOOD COMMENTS

- Legal comments

```
// Copyright (C) 2003,2004,2005 by Object Mentor, Inc. All rights reserved.
// Released under the terms of the GNU General Public License version 2.
```

ENTER.
tech with a heartbeat

GOOD COMMENTS

- Informative comments

```
// format matched kk:mm:ss EEE, MMM dd, yyyy
Pattern timeMatcher = Pattern.compile("\\d*:\\d*:\\d* \\w*, \\w* \\d*, \\d*");
```

GOOD COMMENTS

- Explanation of intent

```
//This is our best attempt to get a race condition
//by creating large number of threads.
for (int i = 0; i < 25000; i++) {
    WidgetBuilderThread widgetBuilderThread =
    new WidgetBuilderThread(widgetBuilder, text, parent, failFlag);
    Thread thread = new Thread(widgetBuilderThread);
    thread.start();
}
```

GOOD COMMENTS

- Warning consequences

```
// Don't run unless you
// have some time to kill.
public void _testWithReallyBigFile() {
    writeLinesToFile(10000000);
    response.setBody(testFile);
    response.readyToSend(this);
    String responseString = output.toString();
    assertSubString("Content-Length: 1000000000", responseString);
    assertTrue(bytesSent > 1000000000);
}
```

GOOD COMMENTS

- TODO comments

```
//TODO-MdM these are not needed
// We expect this to go away when we do the
// checkout model protected

VersionInfo makeVersion() throws Exception
{
        return null;
}
```

GOOD COMMENTS

- Amplification

```
String listItemContent = match.group(3).trim();
// the trim is real important. It removes the starting
// spaces that could cause the item to be recognized
// as another list.
new ListItemWidget(this, listItemContent, this.level + 1);
return buildList(text.substring(match.end()));
```

BAD COMMENTS

- Mumbling

```java
public void loadProperties() {
    try {
        String propertiesPath = propertiesLocation + "/" + PROPERTIES_FILE;
        FileInputStream propertiesStream = new FileInputStream(propertiesPath);
        loadedProperties.load(propertiesStream);
    }
    catch(IOException e) {
        // No properties files means all defaults are loaded
    }
}
```

BAD COMMENTS

- Redundant comments

```java
// Utility method that returns when this.closed is true. Throws an exception
// if the timeout is reached.
public void waitForClose(final long timeoutMillis) throws Exception {
    if(!closed) {
        wait(timeoutMillis);
        if(!closed)
        throw new Exception("MockResponseSender could not be closed");
    }
}
```

ENTER. tech with a heartbeat

BAD COMMENTS

- Mandated comments

```
/** *
* @param title The title of the CD
* @param author The author of the CD
* @param tracks The number of tracks on the CD
* @param durationInMinutes The duration of the CD in minutes */
public void addCD(String title, String author, int tracks, int durationInMinutes) {
        CD cd = new CD();
        cd.title = title;
        cd.author = author;
        cd.tracks = tracks;
        cd.duration = duration;
        cdList.add(cd);
}
```

ENTER. tech with a heartbeat

BAD COMMENTS

- Position marker comments

```
// Actions ///////////////////////////////////
...

//------------------------
...
//========================
```

BAD COMMENTS

- Closing brace comments

```
public void loadProperties() {
    try {
        for(int i=0; i< entry.size; i++){
        {
                . . .
        }
    } // try
    catch(IOException e) {
        . . .
    } // catch
}
```

ENTER.
tech with a heartbeat

BAD COMMENTS

- Commented out code

```
this.bytePos = writeBytes(pngIdBytes, 0);
//hdrPos = bytePos;
writeHeader();
writeResolution();
//dataPos = bytePos;
if (writeImageData()) {
        writeEnd();
        this.pngBytes = resizeByteArray(this.pngBytes, this.maxPos);
}
```

BAD COMMENTS

- Too much information

```
/*
RFC 2045 - Multipurpose Internet Mail Extensions (MIME)
Part One: Format of Internet Message Bodies
section 6.8. Base64 Content-Transfer-Encoding
The encoding process represents 24-bit groups of input bits as output strings of 4
encoded characters. Proceeding from left to right, a 24-bit input group is formed by
concatenating 3 8-bit input groups. These 24 bits are then treated as 4 concatenated 6-
bit groups, each of which is translated into a single digit in the base64 alphabet. When
encoding a bit stream via the base64 encoding, the bit stream must be presumed to be
ordered with the most-significant-bit first. That is, the first bit in the stream will be
the high-order bit in the first 8-bit byte, and the eighth bit will be the low-order bit
in the first 8-bit byte, and so on.
*/
```

BAD COMMENTS

- Not obvious connection

```
/*
start with an array that is big enough to hold all the
pixels * (plus filter bytes), and an extra 200 bytes for header info
*/
this.pngBytes = new byte[((this.width + 1) * this.height * 3) + 200];
```

INTRODUCTION

- We should take care that our code is nicely formatted.

- By choosing a simple set of rules that govern the format of the code.

VERTICAL FORMATING

- How big should a source file be?
    - No clear answer to this question, but small files are usually easier to understand than larger files. **So keep your files small**.

THE NEWSPAPER METAPHOR

- The further you read more detail you get: We want our source files to be read like a newspaper article
- The top most part of the code should provide the high-level concepts and algorithms
- Detail should increase as we move downwards

VERTICAL OPENNESS BETWEEN CONCEPTS

- Each line represents an expression or a clause, and each group of lines represents a complete thought.
- Each thought should be separated from each other with blank lines. It helps to separate concepts.
- Vertical density implies close association.

VERTICAL DISTANCE

- Concepts that are closely related should be kept vertically close to each other;

- Closely related concepts should not be separated into multiple files unless you have a very good reason for…

VARIABLE DECLARATION

- Should be declared as close to their usage as possible;

- Control variables in loops should be declared within the loop statement.

INSTANCE VARIABLES

- Should be declared at the top of the class.
- They will not violate the vertical distance in a well designed class because those variables are used in many if not all methods of the class.

HORIZONTAL FORMATTING

- How wide should be our code lines.
- Keep your lines short, within a limit of 80 characters, but it is okay between 100 and 120.
- You should never be able to scroll to the right.

BREAKING INDENTATION

- Never make 1 line if statements or function implementations

```
if (file.isreadable) { return true; }
```

- Dummy scopes
  - Add the dummy body is indented and surrounded by braces. Make the semicolon visible;

```
while (dis.read(buf, 0, readBufferSize) != -1)
;
```

DATA ABSTRACTION

- Hiding the implementation is all about abstraction.
- We don't want to expose the details of our data; We want to express our data in abstract terms.
- And this is not accomplished by using interfaces and/or getters and setters.
- **The worst option is blindly add getters and setters**.

DATA/OBJECT ANTI-SYMMETRY

- Objects hide their data behind abstractions and expose functions that operate on that data
- Data structures expose their data and have no meaningful functions
- *They are virtually opposites!*
- And have their advantages in different situations.

LAW OF DEMETER

- A module should not know about the innards of the objects it manipulates;
- It means that an object should not expose its internal structure through accessor because to do so it exposes its internal structure
- A method should not invoke methods on objects that are returned by any allowed functions.
- *Talk to friends not to strangers*.