-ER

Alternatives, Clients, MVC

Yegor Bugayenko

Lecture #5 out of 8
80 minutes

The slidedeck was presented by the author in this YouTube Video

Examples and Alternatives

-Client Suffix

What About Performance?

Model-View-Controller (MVC)

Rultor + Takes

"When you need a manager, it's often a sign that the managed are just plain old data structures and that the manager is the smart procedure doing the real work."

— Carlo Pescio. Your Coding Conventions Are Hurting You. http://jttu.net/pescio2011your, 4 2011. [Online; accessed 25-09-2024]

Chapter #1:

Examples and Alternatives

Parser

```
class Parser {
  static int parseInt(String t) {
    // Parse String into Integer
  }
  static float parseFloat(String t) {
    // Parse String into Float
  }
  // And many more methods...
}

int x = Parser.parseInt("42");
```

```
class StringAsInt implements Number {
  private final String txt;
  StringAsInt(String t) { this.txt = t; }
  @Override int intValue() {
    // Parse String into Integer
    // and return the value
  }
}

Number n = new StringAsInt("42");
int x = n.intValue();
```

Reader

```
class Reader {
  static char readChar(InputStream i) {
    // Read the next char from the
    // stream and return it, or NULL
    // if the stream is at the EOF
  }
}

InputStream i = new FileInputStream(..);
char c = Reader.readChar(i);
```

```
class Chars
  private final InputStream is;
  Chars(InputStream i)
    this.is = i;
  char next()
    // Read the next char from the
    // stream and throw exception
    // if !exists()
  bool exists()
    // Return TRUE if not EOF
InputStream i = new FileInputStream(..);
Chars chars = new Chars(i);
char c = chars.next();
```

Controller

```
class SimpleController {
  @GET
  @Path("/index")
  HttpResponse index(HttpRequest e) {
    // Build an index page and return
  }
  @POST
  @Path("/update")
  HttpResponse update(HttpRequest e) {
    // Save new user information
    // and return HTTP 303
  }
}
```

```
class IndexPage implements HttpPage
  HttpResponse process(HttpRequest e) {
    // Build an index page and return
  }
class UpdatePage implements HttpPage
  HttpResponse process(HttpRequest e) {
    // Save new user information
    // and return HTTP 303
  }
new AllPages(
  new IndexPage(),
  new UpdatePage()
);
```

Validator

```
class Validator {
  bool isValid(int age) {
    return age >= 18;
  }
}
int a = 23;
Validator v = new Validator();
if (!v.isValid(a)) {
  throw new Exception(
    "Age is not valid"
  );
}
```

```
interface Age
  int value();
class DefaultAge implements Age
  private final int a;
  DefaultAge(int a)
    this.a = a;
  @Override int value()
    return this.a;
class ValidAge implements Age {
  private final Age origin;
  ValidAge(Age age)
    this.origin = age;
  @Override int value()
    int v = this.origin.value();
    if (v < 18)
      throw new Exception("Age is not valid");
    return v;
Age a = new ValidAge(new DefaultAge(23));
```

Encoder

```
package java.net;

class URLEncoder {
  static String encode(String s, String enc) {
    // Encode the string "s" using
    // the "enc" encoding and return
    // the encoded string
  }
}
String e = URLEncoder.encode("@foo");
e.equals("%40foo");
```

You may want to read more about this in my blog [Bugayenko, 2015, 2017].

```
class Encoded implements String {
  private final String origin;
  private final String encoding;
  Encoded(String s, String enc) {
    this.origin = s;
    this.enc = encoding;
  }
  @Override String value() {
    // Encode the string "origin" using
    // the "encoding" and return
    // the encoded string
  }
}
String e = new Encoded("@foo");
e.value().equals("%40foo");
```

The right snippet won't work in Java, since `String` is a final class, not an interface, unfortunately.

Proof: Classes with -ER Suffixes Are More Complex

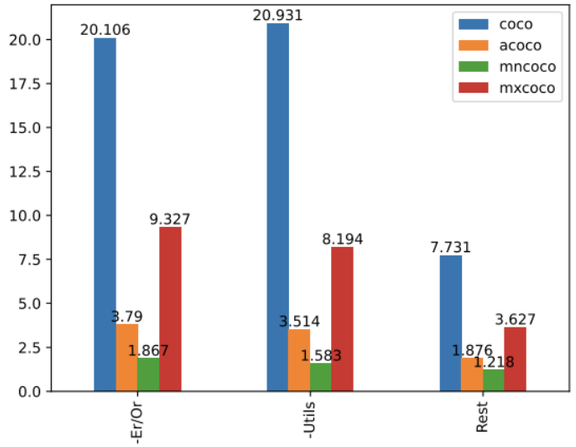


Figure 3. The average values of total CoCo for three groups of Java classes: the classes with "`-Er/-Or`" and "`-Utils`" suffixes are at least 2.5 times more complex.

"We took 13,861 Java classes from 212 open source repositories, divided them into three groups ('-Er' classes, '-Utils', and others), and evaluated their complexity. Because average CC and CoCO in the first two groups were almost 3x larger than in the third group, we concluded that functor classes may be considered bad design."

Source: Anna Sukhova, Alexey Akhundov, Efim Verzakov, and Yegor Bugayenko. Java Classes With "-Er" and "-Utils" Suffixes Have Higher Complexity, 2024

Chapter #2:

-Client Suffix

[AWS]

AWS Java Client

```
class AmazonS3Client {
  createBucket(String name);
  deleteBucket(String name);
  doesBucketExist(String name);
  getBucketAcl(String name)
  getBucketPolicy(String name);
  listBuckets();
  // 160+ more methods here
}
client = new AmazonS3Client("us-1");
client.createBucket("foo");
client.putObject("foo", "a.txt");
client.writeObject("foo", "a.txt", "data");
```

```
region = new S3Region("us-1");
bucket = region.createBucket("foo");
object = bucket.putObject("a.txt");
object.write("data");
```

It's here: jcabi/jcabi-s3.

The left snippet is: 1) procedural, 2) hard to test, 3) resembles a utility class, and 4) is hard to extend. The right one is object-oriented.

Chapter #3:

What About Performance?

Sticky Parseable Object

```
class StringAsInt implements Number {
  private final String txt;
  StringAsInt(String t) { this.txt = t; }
  @Override int intValue() {
    // Parse String into Integer
    // and return the value
  }
}

Number n = new StringAsInt("42");
int x = n.intValue();
```

```
class StickyInt implements Number {
  private final Number origin;
  private int cache = 0;
  private bool cached = false;
  StickyInt(Number n) { origin = n; }
  @Override int intValue() {
    if (!cached) {
      cache = origin.intValue();
      cached = true;
    }
    return cache; } }
```

Is it thread-safe though?

Thread-safe Sticky Parseable Object

```
class StickyInt implements Number {
  private final Number origin;
  private int cache = 0;
  private bool cached = false;
  StickyInt(Number n) { origin = n; }
  @Override int intValue() {
    if (!cached) {
      cache = origin.intValue();
    }
    return cache;
  }
}
```

```
class StickyInt implements Number {
  private final Number origin;
  private final AtomicReference<Integer> cache =
    new AtomicReference<Integer>(null);
  StickyInt(Number n) { origin = n; }
  @Override int intValue() {
    return cache.updateAndGet(
      x -> {
        if (x == null) {
          return origin.intValue();
        }
        return x;
      }
    );
  }
}
```

[Sticky Safe]

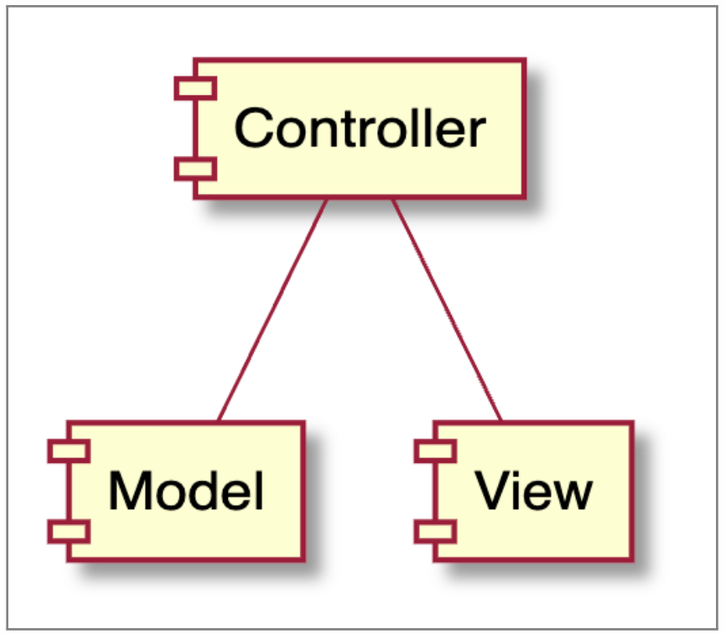
The left snippet is not thread-safety, while the right one is.

Chapter #4:

Model-View-Controller (MVC)

The Controller

```
1  class Controller {
2    @GET
3    @Path("/b{id}")
4    String index(int id) {
5      Book book = em.findById(id);
6      View v = new HtmlView("book.html");
7      v.set("title", book.getTitle());
8      return v.renderHtml();
9    }
10 }
```



This is bad OOP [Bugayenko, 2016].

[Controller HTML]

Book as HTML

```
class Controller {
  @GET
  @Path("/b{id}")
  String index(int id) {
    Book book = em.findById(id);
    View v = new HtmlView("book.html");
    v.set("title", book.getTitle());
    return v.renderHtml();
  }
}
```

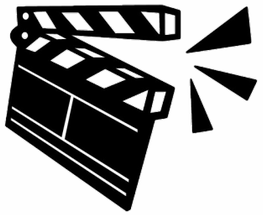
```
interface Book
  String title();
class PgBook implements Book
  String title() // loads from PostgreSQL
interface Page
  String html();
class HtmlBook implements Book, Page
  String html() // renders in HTML
  String title() // returns origin.title()
class PageOnPath implements Page
  private final String path;
  private final Page origin;
  String html() // renders if path matches
```

Check yegor256/jpages and yegor256/takes.

Chapter #5:

Rultor + Takes

rultor.com



takes.org

```
return new TkWithHeaders(
    new TkVersioned(
        new TkGzip(
            new TkMeasured(
                new TkFlash(
                    new TkAppFallback(
                        new TkAppAuth(
                            new TkForward(
                                TkApp.regex(talks, pulse, toggles)
                            )
                        )
                    )
                )
            )
        )
    ),
    String.format("X-Rultor-Revision: %s", TkApp.REV),
    "Vary: Cookie"
);
```

```
private static Take regex(final Talks talks,
    final Pulse pulse, final Toggles toggles) {
    return new TkFork(
        new FkRegex("/robots.txt", ""),
        new FkRegex("/ticks", new TkTicks(pulse)),
        new FkRegex("/status", new TkStatus(pulse)),
        new FkRegex("/s/.*", new TkRedirect()),
        new FkRegex("/sitemap", new TkSitemap(talks)),
        new FkRegex(
            "/xsl/.*",
            new TkWithType(new TkClasspath(), "text/xsl")
        ),
        new FkRegex(
            "/js/.*",
            new TkWithType(new TkClasspath(), "text/javascript")
        ),
        new FkRegex(
            "/css/.*",
            new TkWithType(new TkClasspath(), "text/css")
        ),
        new FkRegex("/", new TkHome(talks, toggles)),
        new FkRegex("/b/([/a-zA-Z0-9_\\-\\.]+)", new TkButton()),
        new FkRegex("/t/([0-9]+)-([a-f0-9]+)", new TkDaemon(talks)),
        new FkRegex("/p/([/a-zA-Z0-9_\\-\\.]+)", new TkSiblings(talks)),
        new FkAdminOnly(
            new TkFork(
                new FkRegex("/t/([0-9]+)", new TkTalk(talks)),
                new FkRegex("/t/([0-9]+)/kill", new TkTalkKill(talks)),
                new FkRegex("/t/([0-9]+)/delete", new TkTalkDelete(talks)),
                new FkRegex("/toggles/read-only", new TkToggles(toggles))
            )
        )
    );
}
```

https://github.com/yegor256/rultor/blob/master/src/main/java/com/rultor/web/TkApp.java

Bibliography

Yegor Bugayenko. Don't Create Objects That End With -ER. https://www.yegor256.com/150309.html, 3 2015.

Yegor Bugayenko. MVC vs. OOP. https://www.yegor256.com/161213.html, 12 2016. [Online; accessed 22-09-2024].

Yegor Bugayenko. Yet Another Evil Suffix for Object Names: Client. https://www.yegor256.com/170912.html, 9 2017. [Online; accessed 22-09-2024].

Carlo Pescio. Your Coding Conventions Are Hurting You. http://jttu.net/pescio2011your, 4 2011. [Online; accessed 25-09-2024].

Anna Sukhova, Alexey Akhundov, Efim Verzakov, and Yegor Bugayenko. Java Classes With "-Er" and "-Utils" Suffixes Have Higher Complexity, 2024.