

| | | | |
|--|--|--|---|
| Meaningful Distinction What's source? What is destination? <pre>public static void copyChars(char a1[], char a2[]){ ... }</pre> | No Member Prefixes //Historic => Old Code or old Coder <pre>private String m_dsc;</pre> | Use Solution Domain Names AccountVisitor VisitorPattern | Avoid Mental Mapping Clean: loop Counter only: i, j, k Dirty: k and l because One (1) ⚡ => too much |
| Use Pronouncable Names Write names out! Longer names trumps short Names Searchable names trumps a constant in Code Avoid Encodings Make names pronouncable Dirty: <pre>class DtaRcrd102 { private Date genymdhms; private Date modymdhms; private final String pszqint = "102";} ⚡</pre> | Interface & Implementation Dirty Clean interface class IShapeFactory ShapeFactoryImpl ⚡ interface Shapefactory | Don't be cute Dirty Clean holyHandGrenade() ⚡ deleteItems() eatMyShorts() ⚡ abort() | cont = continuation |
| One Level of Abstraction per Function inter- String pagePathName = mediate PathParser.render(pagePath); Level low .append("\n"). Level | Don't Repeat Yourself Refactor Clone Copy & Paste | Add Meaningful Context addressFirstName addressLastName addressState | Flag Arguments Passing a boolean into a function is a truly terrible practise render(boolean isSuite) ⚡ |
| Verbs And Keywords verb/noun Pair: (to)writeField(name); assertExpectedEqualsActual(expected, actual); | Use Descriptive Names The Sequence should tell a Story: includeSetup-AndTeardownPages(); includeSuiteSetupPage(); includeSetupPages(); | Functions | Structured Programming Use * occasional multiple return, * break and * continue statement does no harm |
| Hungarian Notation // Name not changed when type changed PhoneNumber phone -String; | Use Problem Domain Names If is there no Programming technique use the name from the Problem Domain DomainDrivenDesign D3 not D3.js | Sideeffects If you must have a temporal coupling, you should make it clear in the name of the function checkPasswordAndInitializeSession() | Don't USE never, ever, any goto statements ⚡ but only in special languages. Be careful, you can destroy or control flow |
| Method Names Use verbs, phrases like PostPayment, deletePage or SavePage Constr- Static factory uctor Methods, with Overload names, that use describe the arguments | | | Function Arguments Useless: <i>Niladic(0) best Solution:</i> includeSetupPage() <i>Polyadic (More then three Arguments):</i> |



Function Arguments (cont)

```
ResponseEntity<Map>
responseEntity1 =
restTemplate.exchange
(REST_SERVICE_URI+"/contract/"+ID,
HttpMethod.PUT, entity, Map.class)
```

Usable

Monadic:

```
boolean fileExists("MyFile")
```

Dyadic:

```
assertExpectedEqualsActual(expected,
actual);
```

Triadic:

```
assertEquals(1.0, amount, .001);
```

TO paragraph (cont)

To say this differently, we want to be able to read the program as though it were a set of TO paragraphs, each of which is describing the current level of abstraction and referencing subsequent TO paragraphs at the next level down.

MethodHeader:

To include the setups,

Code:

we include the suite setup if this is a suite, then we include the regular setup.

Avoid Mental Mapping

loop Counter i, j, k

only:

Never I because
One(1)

One Thing

Functions should do **one thing**. They should do it well. They should do it only

* Aids the Reader

* Promote Reuse

*Eases Naming & Testing

*Avoid Sideeffects

Don't add Gratuitous Context

You need:

* descriptive Skills

* shared cultural background

* Don't be afraid of Renaming

* Use Refactoring Tools

-> **IntelliJ**

Don't Pun

Use:

```
insert(), append()
```

Instead of

```
add()
```

Command Query Seperation

Either your function should change the state of an object, or it should return some information about that object:

```
public boolean set(String
attribute, String value);
```

Clean:

```
if (attributeExists("username"))
{setAttribute("username",
"unclebob"); ... }
```

Dirty:

```
if (set("username", "uncle-
bob")){...} ⚡
```

Blocks And Indenting - Arrow Code

The indent level of a function should not be greater then

one or two

TO paragraph

We want every function to be followed by those at the next level of abstraction so that we can read the program, descending one level of abstraction at a time as we read down the list of functions.



By **ChaosJD**

cheatography.com/chaosjd/

Not published yet.

Last updated 10th November, 2019.

Page 2 of 2.

Sponsored by **ApolloPad.com**

Everyone has a novel in them. Finish Yours!

<https://apollopadd.com>