

Object-Oriented Programming

Method Naming Standards

Michael L. Collard, Ph.D.

Department of Computer Science, The University of Akron

Naming Standards

Alsuhaibani, R., Newman, C., Decker, M., Collard, M.L., Maletic, J.I., [On the Naming of Methods: A Survey of Professional Developers](#), in the Proceedings of the 43rd International Conference on Software Engineering (ICSE), Madrid Spain, May 25 - 28, 2021, 13 pages.

Alsuhaibani, R., Newman, C., Decker, M., Collard, M.L., Maletic, J.I., [A Survey on Method Naming Standards: Questions and Responses Artifact](#), in the Proceedings of the 43rd International Conference on Software Engineering: Companion (ICSE-Companion) Artifact Track, Madrid Spain, May 25 - 28, 2021, 2 pages.

Alsuhaibani, R., Newman, C., Decker, M., Collard, M.L., Maletic, J.I., [An Approach to Automatically Assess Method Names](#), in the Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension (ICPC), Pittsburgh, PA, USA, May 16-17, 2022, 12 pages.

- Survey of 1100+ developers
- Result: *Wide support for these standards*
- Result: *Similar across multiple programming languages*
- Result: *Very slight differences based on experience*
- Paper refers to *methods*, but results also apply to *free functions*

List of Naming Standards

- Naming Style
- Grammatical Structure
- Verb Phrase
- Dictionary Terms
- Full Words
- Idioms and Slang
- Abbreviations & Acronyms
- Prefix/Suffix
- Length

Naming Style

```
getFullName();
```

```
getScriptState();
```

```
call_with_default();
```

```
garbage_collection();
```

```
check_static_allocation_size();
```

```
getfullName();
```

```
getscriptstate();
```

- Popular styles: camelCase and under_score
- Little difference in cognitive load during comprehension
- Pretty equivalent, except that camelCase appears to have a slight advantage

Grammatical Structure

```
registerManagedResource();  
managedResourceRegister();  
  
// verb phrase  
drawContentBorder();  
  
// verb phrase with a  
prepositional phrase  
performTestsFromZipFile();  
  
// noun phrase  
nextArea();
```

- Function names with multiple words need to have a grammatically correct sentence structure
- *verb phrase*
- *verb phrase with a prepositional phrase*
- *noun phrase*

Verb Phrase

manage_caching_sizes();

computeProductBlockingSizes();

get_cached_node();

x_cached_node();

- Not only grammatically correct but contain a verb or a verb phrase

- Functions are actions

Dictionary Terms

`findLength();`

`abcdefg();`

`cccc();`

`aa2020();`

- Words used in the function name are actual dictionary terms
- Meaningful natural-language terms
- Not a non-dictionary word, e.g., "2" instead of "to"
- Use of non-dictionary words makes an otherwise high-quality identifier hard to understand

Full Words

```
startConnection();  
  
/* for the same function as above  
*/  
c();
```

- Use full words as opposed to single-letter identifier names
- Research clearly shows improved comprehension with full-word identifiers instead of single-letter names

Idioms and Slang

```
// personal name  
fido();  
  
// idiom  
cutting_corners();  
  
// slang  
CurveBall();
```

- The name should not contain personal expressions, idioms, or slang
- Special case of *Dictionary Terms and Full Word* standards
- Using slang and idioms referred to as a *cute* practice
- Choose clarity over entertainment value

Abbreviations & Acronyms

`getStr(); pyConnection();`

`get_algo(); close_db_connection();`

`open_GUI_interface(); get_URL();`

`get_FIFO(); traverse_DOM_tree();`

`repr(); getProtoNameNode();`

`get_QWE(); sendAAAA();`

- If a name contains words that are abbreviations or acronyms, they should be well-known or standard

- *evil abbreviations* are unfamiliar abbreviations that are very hard for programmers to understand

- Standard ones used by the organization or domain

Prefix/Suffix

```
gimpItemGetPath();
```

```
swift_stdlib_u_char();
```

```
m_count();
```

- Method names should not contain a prefix or suffix that is a term from the system
- Developers learn to ignore prefix/suffix, existence becomes a marker of old code. Names should not contain scope information.
- Does not apply to the C language or groups of related free functions

Length

```
return false if no setter was found and if report no setter found is false();
```

Length

- Long debate in the research literature and developer discussion [Cunningham](#)
- Longer, more descriptive names have a positive impact on code comprehension [Knuth](#)
- Fixed maximum number of words in the name from 1 to 5.
- There is some disagreement in the results, as some developers stated that there should be no maximum length in all cases. However, almost all agreed to around 5 in nearly all cases.
- One exception: Names of unit-test (test-case) methods

Summary

- Presented standards are widely accepted in the developer community
- Quality of identifiers has a substantial impact on code comprehension
- Although the survey was for methods, almost all the standards apply to free functions