GO FIRST Source Code Conventions

Max Veit

Last Updated: June 8, 2012

Contents

1 References

- Sun Code Conventons for Java: `http://www.oracle.com/technetwork/java/codeconventions-150003.pdf`
  This is a good, but not perfect, introduction to basic style for Java source code. We will be using it with a few modifications (detailed below) for all of our Java, C and C++ code because of the syntactic similarities between the languages. Please read this thoroughly if you have no experience with C syntax (come on, it's only 18 pages. A TL;DR response will disqualify you from our programming team). If you have programmed in the C-syntax languages before, at least skim the guide to see what specific conventions well be using.
- Style Guide for Python Code: `http://www.python.org/dev/peps/pep-0008/`
  A very good guide for Python code with general advice that applies to other programming languages as well. Please at least read the introduction, first section, and section on comments. If we do any coding in Python, the style should follow this guide. Comments in other languages should follow the advice in this guide; for all other areas the language's style guide naturally has precedence.
- Pasta Code: `http://www.gnu.org/fun/jokes/pasta.code.html`
  If youre doing object-oriented programming, please try for ravioli code. Canderli code and Ristto code should be avoided because of poor maintainability. Polenta code is nice, too, if it never needs to be modified (which is generally never the case, so watch out!)
- Good Code: `http://xkcd.com/844/`
  Please try to stay in the "Code Well" loop. Yes, requirements do change, and I've had to throw out a few projects because of this, but trust me: this is not always the case!

2 Technical Notes

- Encoding: Please use UTF-8 for general symbol support and inter-system compatibility. Line-endings usually aren't a problem if your editor is not Notepad (if it is, shame on you. See A), but if we have problems with this we'll have to go with Windows line endings (CR LF).
- General Rules: We will presumably have programmers on all different systems, so if you have a choice between convenience and compatibility you should almost always choose compatibility.

3 Whitespace

Please read the Python style guide, sections "Code lay-out" and "Whitespace…" (you can ignore the "Imports" subsection since it is Python-specific). These rules apply well both to Python and C-syntax code, so please observe them.

A Editor and IDE recommendations

A.1 Text editors by platform

Windows As mentioned above, you want to move away from Notepad. It's not even good for editing configuration files. One alternative that I know of is Notepad++ (`http://notepad-plus-plus.org/`).

Mac I don't know of any good Mac-specific text editors (the built-in one is usable, but unsuited to programming), although quite a few cross-platform editors integrate well with OSX. You could also try TextWrangler (`http://www.barebones.com/products/TextWrangler/`), although I have personally never used it.

GNU/Linux The built-in editors GEdit (GNOME) and Kate (KDE) are both good, extensible, full-featured, and usable for programming in many languages. However, it might be worth it to take a look at some of the cross-platform editors as well.

Cross-platform There are two classic editors for Unix-like systems that have been extensively ported:

- Vim, the "improved version" of Vi (`http://www.vim.org`). My editor of choice; it's by far the most efficient editor I've ever used. However, Vim is notorious for its learning curve. If you have a free weekend and feel like learning to use the editor efficiently, check out the program "vimtutor" which should be installed along with the editor (try typing `:help tutor` after starting Vim). Features include everything you'd expect from a programmer's text editor (syntax highlighting, search/replace with regexes, code folding, and diff to name a few). Built-in support for most known programming languages, often with extensions available for specific languages. Originally designed for use in a terminal, although good graphical versions are available on the download page.
- (x)Emacs (`http://www.xemacs.org`), perhaps the more full-featured (according to some, bloated) of the two classic editors. I personally find it unusable, although if you're a boss with chained keyboard shortcuts in Word or whatever you might like this editor. Lots of programmer's features and extensions available. It even has a built-in psychologist (try `M-x doctor` or something like that). The graphical version is xEmacs.

In addition, jEdit (`http://www.jedit.org/`) is a full-featured, cross-platform text editor with support for many programming languages. I've used it for editing Java code and found it to work well for this purpose.

A.2 Cross-platform IDEs by language

TODO