

Re: How to name variables in a program?

Source: <http://coding.derkeiler.com/Archive/General/comp.programming/2005-05/msg00698.html>

- *From:* spinoza1111@xxxxxxxxxx
 - *Date:* 28 May 2005 10:01:25 -0700
-

Philip wrote:

> Martijn wrote:

>

>>> Having a strong convention for naming your variables in your program

>>> is an important issue often overlooked.

>>> Can anyone point me to a good source on the web dealing with this

>>> issue?

>>

>> Ever since I picked up the book Windows programming book by Petzold, I

> have

>> been using something called "Hungarian notation":

>> http://en.wikipedia.org/wiki/Hungarian_notation

>

>> Although it is by no means a guide on picking good names for your

> variables,

>> this convention works with prefixes to indicate variable type.

>

> There's a long list of various reasons why that's wrong. Use Google and

> Google Groups for "Hungarian notation" for the bloggage of many an engineer

> coming to realize this. Unfortunately, here comes another one, because below

> here you point to one of the biggest and best examples why HN is wrong.

>

>> – every structure I define, I trail with a comment indicating the

>> hungarian code

>

> An early MS editor, Programmer's WorkBench, could not browse code. You could

> not click on an indentifier and hit "Go to Definition". So, to enable a kind

> of browsing, you could drag-select the wart, such as msgFoo for MSG, and

> invoke Help. It would take you to the online help for MSG.

>

> This minor editor feature is a major reason all the early MS code used HN.

> Modern MS code is drifting away from it.

>

>> – prefix global variables with g_

>

> No prob. I forgot "prefix members with m_". But these indicate storage, not

> type, so they are not HN.

Re: How to name variables in a program?

- >
- >> – prefix pointers with lp (this stands for "long pointer" and is still
- >> reminiscent of the 16 bit Windows days)
- >
- > That is a primary example why Hungarian Notation wrong. VC++ supports
- > libraries that use the unpronouncable name LPCTSTR, meaning Long Pointer to
- > a Typed Constant STRing.
- >
- > The biggest and most commonly cited reason HN is wrong is this: When you
- > refactor, if a type changes, you must either hunt down and change every
- > instance of the HN and change it, or you must live a lie, and forsake HN's
- > slim capacity to self–document.

This is a problem only if the name is so global that it exists in multiple files: but very few names should be so global that they exist in more than one file. Within one file, using a modern editor it is straightforward to change `intIndex1` to `lngIndex1` by a global change that is toggled to change complete words only.

- >
- > That problem has bitten Microsoft. My version of VC++ can't produce 16–bit
- > programs. If it produced a 64–bit programs, the L would in fact be a Short,
- > because 64–bit chips support various 32– and 48–bit memory models, where
- > 32–bit short pointers can peacefully coexist with various 48–bit segmented
- > pointers, and with 64–bit super pointers.
- >
- > So the L in LPCTSTR lives a lie. At refactor time, MS cannot search and
- > replace every instance of LPCTSTR in the world. MS header files are infested
- > with this nonsense. The beloved DWORD is no longer a Double WORD; on a
- > modern CPU it is a WORD.

>
Global identifiers for this reason should not use HN. Where HN remains useful is in local and class level identifiers.

- > And POINT_TYPED_STRING would have been pronouncable.

If types in a program are so unstable that one is changin `intIndex` to `strIndex` then one needs to step outside for a smoke and figure out why the design is unstable in the first place.

If the code which uses `intIndex1` is targeted for both a 16 bit Int environment and a 32 bit Int environment then common sense would interpret the `int` to mean "integer in relation to the platform" and make the type a preprocessor symbol if C is in use.

- >
- > --
- > Philip
- > <http://www.c2.com/cgi/wiki?ZeekLand>

.

- *Follow-Ups:*
 - ◆ *Re: How to name variables in a program?*
 - ◇ *From:* Arafangion

- *References:*
 - ◆ *How to name variables in a program?*
 - ◇ *From:* SerGioGio
 - ◆ *Re: How to name variables in a program?*
 - ◇ *From:* Martijn
 - ◆ *Re: How to name variables in a program?*
 - ◇ *From:* Philip

- Prev by Date: *A very wierd sorting algorithm*
- Next by Date: *Re: Creationist software engineering? A frightening possibility!*
- Previous by thread: *Re: How to name variables in a program?*
- Next by thread: *Re: How to name variables in a program?*
- Index(es):
 - ◆ *Date*
 - ◆ *Thread*