Este documento contiene una serie de pautas que debe seguir en la documentación de sus tareas programadas. El objetivo principal de estas pautas es organizar la solución de problemas. Para tener una idea más clara de estas pautas refiérase a http://www.di-mare.com/adolfo/p/convpas.htm.

Estilo

- Indente apropiadamente.
- Alinee verticalmente la llave de cierre '}' con el inicio del bloque de código correspondiente.
- Use un tamaño máximo de línea para que no sea necesario desplazarse demasiado en cada línea.
- No coloque más de una instrucción por línea.
- Use espacios para separar comentarios, no asteriscos ni otros símbolos.
- Use espacios para indicar líneas de código relacionadas.
- Estandarice el tamaño de tabulación.
- Use nombres de variables y métodos significativos.
- Mezcle mayúsculas y minúsculas para facilitar la lectura en identificadores.
- Variables booleanas deberían tener nombres que impliquen valores booleanos.
- Estandarice el uso de mayúsculas y minúsculas en palabras reservadas.
- Estandarice la declaración de variables.
- Indique qué cierra cada llave de cierre '}'.

Documentación Interna

- Mantenga documentación interna actualizada.
- La documentación interna debe ser clara, concisa, correcta y corta.
- Al inicio de cada archivo incluya nombre de autor(es) y una breve descripción de lo que contiene el archivo.
- Documente en dos pasos: definir especificaciones de cada método o ADT y luego al implementar cada uno.
- Para cada rutina o método haga comentarios estándar, repetitivos, que indiquen al menos EFECTO (lo que hace el método), REQUIERE (qué requiere para trabajar debidamente, cómo son los parámetros), MODIFICA (de qué manera modifica y qué cosas modifica).
- Evite comentarios al final de una línea de código, sólo los use para declaraciones de variable.
- Si usa comentarios al final de líneas colóquelos en la misma posición de tabulación.
- Elimine comentarios temporales o innecesarios.
- Estudie bien un método antes de documentarlo.
- Use frases completas para documentar.
- Comente mientras programe.
- Comente cualquier cosa que no sea obvia, no comente lo obvio.
- Se calificará ortografía y redacción en los comentarios.

Documentación Externa

- Incluya nombres de autor(es).
- Describa el problema a resolver.
- Plantee la solución. Use un alto nivel de abstracción para detallar la metodología de solución. Use cualquier instrumento necesario incluyendo diagramas, dibujos, tablas, etc.
- Describa todas las partes del programa.
- Describa el uso de archivos.
- Describa entradas y salidas.
- Incluya una lista de los requerimientos especiales del programa.
- Describa la manera y los datos que se pueden usar para comprobar su correcto funcionamiento.
- Se calificará ortografía y redacción.

Convenciones de Programación para Pascal
Adolfo Di Mare
Revisión 2.0

3. Documentación de programas

La documentación de programas tiene dos objetivos claramente definidos: primero lograr que un usario del programa pueda usarlo, y segundo lograr que entender el funcionamiento del programa sea fácil, de forma que el programa pueda ser cambiado o mejorado con un mínimo de esfuerzo.

A este fin, todos los buenos consejos de nuestros profesores de Redacción y Ortografía de secundaria son válidos: la exposición de ideas en la documentación debe ser clara, concisa, sencilla, válida, exacta. Además, debe usarse el lenguaje adecuadamente, evitando ambigüedades.

Escribir la documentación de un programa es un gran esfuerzo intelectual. Escribir no es fácil, y en general las personas lo hacen mal. Por eso lo usual es producir documentaciones pobres, mediocres o pésimas. Debemos cambiar nuestra actitud hacia la documentación, que en realidad es necesaria para que el producto de nuestro trabajo esté terminado: un programa sin documentación es un auto sin manubrio; un programa sin documentación no es un programa, es una adefesio.

Es en general cierto que los programadores derivan placer del programar, no del documentar. Además, parece que lo contrario es también cierto: a los programadores les duele escribir la documentación, hasta tal punto que la dejan para el final. Lo cierto es que la documentación es parte de la programación, independiente del estado anímico del programador. Cada programador debe aprender a escribir la especificación de su programa (o sea, la documentación), antes de escribir el programa.

Esta actitud negativa hacia el documentar se deriva de un estilo corrupto de programación, promovido por tratar de escribir el programa a la mayor celeridad. Se cree que lo único importante es que el programa funcione, y se relega a un segundo plano todo lo demás. Si al no documentar se logra en algunos casos terminar más rápidamente el programa, en realidad se hace mucho más difícil su mantenimiento posterior, pues la mayoría de los programas deben ser modificados muchas veces, antes de ser desechados.

Si un programa se construye siguiendo las sanas prácticas de diseño modular y programación estructurada, es entonces necesario definir primero qué debe hacer cada parte del programa, mediante una especificación, para luego escribir cómo se logra. Esto es prueba de que el engorro que representa al programador el escribir documentación se deriva de su ignorancia de las técnicas de programación usadas por los buenos profesionales, y no de que sea desagradable escribirla: antes de construir un programa es necesario definir cuál debe ser su comportamiento. Es triste que muchos programadores no diseñen sus programas: el resultado es un sistema "feo".

Además, conforme se codifica cada algoritmo, el programador puede hacer pequeñas anotaciones en el programa, que aumentan su legibilidad. No es necesario hacer más. Quiere esto decir que la documentación (interna) del programa se escribe de manera natural con el programa, en dos pasos: primero al definir las especificaciones de cada procedimiento o tipo abstracto de datos, y luego al concretar cada uno de los algoritmos. Hacer más que esto es innecesario, e irrelevante.

Desde una perspectiva de alto nivel, al no documentar inicialmente un programa adecuadamente, lo que se hace es traspasar el costo de la documentación al futuro, cuando deben hacerse modificaciones. Pero si no existe documentación, para cada modificación debe estudiarse profundamente el funcionamiento del programa, de hecho redescubriendo todo lo no documentado, con la desventaja adicional de generalmente quien modifica el programa no es el mismo que lo escribió en primera instancia. Y cada nueva modificación implica de nuevo un esfuerzo intelectual, que pudo evitarse al documentar bien el programa desde el principio.

O sea, siempre es necesario documentar, y lo mejor (y más barato) es hacerlo cuando se está escribiendo el programa. No después, cuando ya se han olvidado los detalles relevantes, o las decisiones de impacto que se tomaron al concretar la implementación.

Si un programa tiene más de 250 líneas (que es lo usual), el programador deberá trabajar en él varias horas, sino varios días. El documentar su programa mientras lo escribe le evita recordar todos los detalles de su implementación, y le obliga a definir y aclarar sus ideas. Si una persona no puede hablar (escribir) concretamente sobre algo, menos podrá programarlo. La prueba de fuego es lograr que otro entienda lo que uno ha producido: ¿se atreve usted ha tomarla en cada uno de sus programas?

Además, como todo buen escritor, un programador debe revisar el código que ha escrito una y otra vez. En cada nueva revisión encontrará nuevas maneras de documentar su programa, y podrá corregir las imprecisiones en que haya incurrido. La programación es una proceso iterativo, y es necio tratar de lograr un resultado correcto al primer intento.

Pasemos ahora a examinar en detalle lo que debemos incluir en nuestra documentación. La documentación del programa en general se divide en dos partes: interna y externa. La interna es la que acompaña al código fuente, y todo lo demás es la externa. Aunque es posible prescindir de la documentación externa por completo, en general la costumbre es mantener la documentación interna simple y concisa. Cuando es necesario hacer grandes aclaraciones en prosa o por medio de dibujos, es mejor incluirlas en la documentación externa.

Un teórico de la programación podría argumentar que la diferencia entre documentación interna y externa no existe en realidad, pues en la práctica lo que debe hacerse es escribir una jerarquía de documentos que describen un producto de logical. En la práctica cada programa o sistema tiene requerimientos específicos, y no es fácil usar siempre el mismo patrón de documentación.

Por eso, el sagaz programador deberá incluir tantos niveles descriptivos, o de abstracción, como sean necesarios para lograr una buena documentación. No debe poner más de lo suficiente, ni menos de lo necesario. Sobre todo, debe producir un conjunto de documentos que le satisfagan.

Como corolario de lo anterior se desprende que un buen método para evaluar la documentación producida, es contestar la pregunta: ¿Si me entregaran a mí esta documentación para que yo modificara el programa, sería suficiente? Obviamente un mejor método para evaluar la documentación es lograr que un segundo la analice, a la luz de esta misma pregunta. Todo aquello que él no pueda resolver fácilmente a partir de lo documentado merece ser revisado y mejorado, hasta lograr un documento que, valga la redundancia, si documente.

Es bueno escribir los programas para el mundo, no para "yo". Tal vez el sentirnos expuestos a la crítica de los demás nos ayude a producir programas de más calidad.