



CI2691: Laboratorio de Algoritmos y Estructuras I

TALLER 01: Familiarización con ambiente de trabajo. Estructura general de un programa en PASCAL. Declaración de constantes y variables. Tipos Básicos (entero, caracter, real, boolean). Instrucciones de entrada y salida. Especificaciones de entradas y salidas, precondition y postcondición. Comentarios. Documentación del código.

Familiarización con ambiente de trabajo

- Terminal y línea de comando (shell)

- Directorios (creación, eliminación, cambio)

 - Crear un directorio con su nombre para sus programas

 - Cambiarse al directorio “demo” de fpc y luego al “text”

- Editor de código fuente (gedit, notepad++ u otro)

 - Visualizar en el editor algunos de los fuentes en el directorio “text”

- Compilador (fpc)

 - Copiar el fuente “hello” del directorio “text” a su propio directorio

 - Compilar y correr el programa “hello” en su propio directorio

 - Usar tres ventanas distintas editar, compilar y correr

Estructura general de un programa en PASCAL [1].

- Sección “16.1 Programs”, salvo “uses clause”

- Sección “16.4 Blocks”, en la “declaration part” sólo “constant declaration part” y “variable declaration part”

- Sección “13.2.1 Compound statements”

Declaración de constantes y variables [1].

- Sección “1.4 Identifiers”

- Sección “1.3 Reserved words”

- Sección “2.1 Ordinary constants”

- Sección “4.1 Definition”

- Sección “4.2 Declaration”, sólo usaremos variables regulares (primera forma)

Tipos Básicos (entero, caracter, real, boolean) [1].

- Sección “3.1.1 Ordinal types”, sólo tipo “integer” y tipo “boolean”

Sección “3.1.2 Real types”, s’lo tipo “real”

Sección “3.2.1 Char”

Instrucciones de entrada y salida.

Procedimientos predefinidos: read readln write writeln

Ejercicio: Escriba un programa en Pascal para

Imprimir los números del 1 al 4 en una misma línea

Imprimir los números del 1 al 4 en distintas líneas

Leer los valores de 3 variables una entera, otra real y otra caractr, todas en una misma línea e imprimirlas en la línea siguiente

Leer los valores de 3 variables enteras en líneas distintas e imprimirlas en orden inverso al orden en que se leyeron

Comentarios [1].

Sección “1.2 Comments”

Documentación del código [2]

Es importante seguir las convenciones de estilo de codificación de cualquier lenguaje de programación por las siguientes razones:

- Gran parte del tiempo de desarrollo de software se invierte en modificarlo, lo cual requiere repetidas lecturas y re-lecturas del código.
- Por lo general, los programas no son escritos sólo por una persona. Además, con frecuencia se requiere realizar mantenimiento a software escrito por otros programadores.
- Las convenciones de estilo incrementan la legibilidad de los programas, permitiendo que cuando otras personas lean el código puedan entender fácilmente su funcionamiento.
- Para que las convenciones funcionen, es necesario que todo el que escriba programas las acepte y haga uso de ellas.

Todo código debería comenzar con un pequeño resumen en donde se liste el nombre y propósito del programa, el autor y la última fecha de modificación. Siguiendo el ejemplo:

```
(*  
* HolaMundo  
* Este programa imprime por pantalla el mensaje "hola mundo"  
* Autor: Pedro Peréz  
* Última modificación: 01/01/2005  
*)
```

Las convenciones de nombres hacen a los programas más entendibles haciéndolos más fáciles de leer. Además pueden dar información acerca de la función del identificador

Variables: Las palabras internas comienzan con mayúsculas. No deben comenzar con _ o \$. Los nombres de variables deben ser cortos pero significativos. Deberían ser mnemónicos - o sea,

diseñados para indicar al observador casual del código la intención de su uso. Nombres de variables de un carácter deberían ser evitados, excepto por variables temporales "desechables". Nombres comunes para variables enteras temporales son i, j, k, m y n.

Constantes: Los nombres de variables declaradas como constantes del programa deberían estar todo en mayúsculas, con las palabras separadas por underscore ("_").

Comentarios de cola: Son comentarios muy cortos que pueden aparecer en la misma línea que el código que está describiendo, pero debe ser separado lo suficiente del código. Si aparece más de un comentario corto en un trozo de código, los mismos deben ser indentados a la misma altura.

Las constantes y variables deben declararse cada una en una línea diferente u colocarle un comentario de cola a cada una.

Ejercicio: Documente los programas que hizo anteriormente en este taller

Especificaciones de entradas y salidas, precondition y postcondición. [3]

Sección "Pre y Post-Condiciones", salvo ejemplo de "operador de concatenación" que no son admitidos en Pascal

Ejercicio: Hacer las especificaciones de los problemas siguientes, crear los respectivos archivos de código fuente en Pascal, incluya las declaraciones de variables respectivas, coloque sólo las instrucciones de entrada y salida, mas no las de cálculo (esto será tema del próximo taller)

- Dado un número natural n, calcular el valor absoluto de n.
- Dadas dos variables enteras a y b, calcular (en una variable booleana), si el valor de a es mayor al de b, y además b es par.
- Dadas tres variables enteras, a, b y c, calcular (en una variable booleana) si se cumple la ecuación $a/b = c$
- Dado un numero natural n, calcular (en una variable booleana) si dicho número es un cuadrado perfecto.

Referencias

1. *Michaël Van Canney Free Pascal : Reference guide. Reference guide for Free Pascal, version 2.6.0, Document version 2.6, December 2011, Disponible en la web:*
<http://www.freepascal.org/docs.var>
2. *Gabriela Montoya, Jorge Guerra, Víctor Fuentes, Wendi Urribarrí, Yolifé Arvelo, y Jesús Ravelo Guía de Estilo Para Programas en GaCeLa, Disponible en la web:*
<http://wiki.lal.labf.usb.ve/GacelaWiki/Wiki.jsp?page=GuiaDeEstilo1-4>
3. *Damaris Candanedo y Jorge Guerra, Guia para escribir programas en GaCeLa .Última actualización: Mayo 28, 2003, Disponible en la web:*
<http://ldc.usb.ve/~gpalma/cursos/ci2691/guiagacela.html>