



UNIVERSIDAD SIMÓN BOLÍVAR
Departamento de Computación y
Tecnología de la Información

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. Comentarios. Documentación del código. Especificaciones de entradas y salidas, precondition y postcondición.

1) Familiarización con ambiente de trabajo. Terminal y línea de comando (shell)

Los comandos básicos que se necesita usar en un terminal son los siguientes

COMANDO	DESCRIPCIÓN
pwd	Da la ruta del directorio actual
ls	Lista los elementos del directorio actual
cd [ruta/directorio]	Permite cambiar de directorio
cd ..	Permite retroceder al directorio padre
mkdir [nombre_del_directorio]	Crea un directorio en la ruta actual
rm -r [nombre_del_directorio]	Elimina un directorio
rm [nombre_del_archivo]	Elimina un archivo
clear (Ctrl+l)	Limpia la pantalla
fpc [archivo.pas]	Compila un programa en Pascal
./[archivo]	Ejecuta un programa en Pascal
tecla Tabulador	Permite autocompletar
teclas de flechas UP, DOWN	Visualiza los comandos realizados
Ctrl+C	Detiene el proceso actual

Usando estos comandos realice las siguientes acciones:

1. Directorios (creación, eliminación, cambio)
 - Crear un directorio con su nombre para sus programas
 - Cambiarse al directorio “demo” de fpc y luego al “text”
2. Editor de código fuente (gedit, notepad++ u otro)
 - Abra en algún editor algunos de texto los fuentes en el directorio “text”
3. Compilador (fpc)
 - Copie algún archivo de extensión “.pas” que se encuentre ubicado en el directorio “Laboratorio 1” del aula virtual a su propio directorio
 - Compilar y correr el programa ese programa desde su propio directorio
 - Abra tres ventanas distintas: una para editar, otra para compilar y otra para correr

2) Estructura general de un programa en PASCAL. 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.

1. Lea la guía “Introducción a Pascal” [4] que se encuentra en el directorio “Laboratorio 1” del aula virtual.
2. Investigue en el manual de Free Pascal (<http://www.freepascal.org/docs.var>) [1], cuáles son los tipos ordinales de Pascal.
3. Lea el código de los archivos ejercicio1.pas, ..., ejercicio8.pas y complete la DESCRIPCION (lo qué hace el programa) de cada uno. Para ello escriba una oración de máximo dos líneas.

3) 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  
* DESCRIPCION: Este programa imprime por pantalla el mensaje  
*           "hola mundo"  
* Autor: Pedro Pérez  
* Última modificación: 01/01/2005  
*)
```

1. Pregunta de Reflexión: ¿Fue fácil para usted, leer el código escrito por otros? ¿Por qué?

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.

2. ¿El código de los programas `ejercicio1.pas`, ..., `ejercicio8.pas` cumple con estas convenciones?

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

Las pre y post-condiciones son expresiones booleanas que deben ir entre llaves (`{}`), una expresión booleana está formada por expresiones matemáticas unidas mediante operadores lógicos, por ejemplo: `a+b>c /\ b+c>a /\ a+c>b`.

Los operadores permitidos son:

- *Operadores matemáticos:* `{=, !=, >, <, <=, >=, +, -, *, /, div, mod, max, min}` todos estos son operadores binarios, esto quiere decir que requieren de dos términos para su funcionamiento, por ejemplo: `a+b`, `x mod y`, `m>n`. Los operadores `div`, `mod`, `max` y `min` están definidos únicamente para los enteros (`int`).
- *Cuantificadores:* `%forall`, `%exist`, `%max`, `%min`, `%sigma`, `%pi` son usados para verificar condiciones en arreglos y matrices. Estos devuelven un valor booleano, en los casos de `%forall` y `%exist`, y un valor entero para `%max`, `%min`, `%sigma` y `%pi`. La sintaxis de los cuantificadores es la siguiente: `(<cuantificador> <variable>: <rango de la variable>: <expresión con arreglos>)`. Un uso de cuantificadores podrá ser el siguiente: `(%exist i : 0<=i<N : X min a[i])`. En este ejemplo se verifica si existe un elemento del arreglo que sea menor que un valor `X`.
- *Operadores lógicos:* `{ /\ (conjunción), \/ (disjunción), => (implicación), == (equivalencia), ! (negación)}`

- *Paréntesis*: { () } estos indican precedencia con respecto a la evaluación

1. ¿ Las precondiciones y postcondiciones de los archivos ejercicio1.pas, ..., ejercicio8.pas le ayudaron a entender el código de estos programas?

5) Ejercicios: 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).

Cree una carpeta dentro de su grupo titulada “Laboratorio 1”. Guarde los archivos creados con los nombres sugeridos a continuación.

1. Lab1Ejercicio1.pas: Dadas tres valores enteros a , b y c diferentes, determinar el valor de la división del cuadrado de a entre $(b - c)$ y almacenarlo en d .
2. Lab1Ejercicio2.pas: Dados tres valores a, b, c intercambie los valores de manera que cumplan $a \leq b \leq c$
3. Lab1Ejercicio3.pas: Dados dos números enteros m y n , devuelva m/n si $m=10$, $m*n$ si $m=5$, $m+n$ si $m=3$, m^n si $m=2$ y en cualquier otro devuelva m .
4. Lab1Ejercicio4.pas: Dada una fecha dd/mm/aa diga si está en el trimestre septiembre-diciembre del año 2013

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, *Guía para escribir programas en GaCeLa .Última actualización: Mayo 28, 2003, Disponible en la web:*
<http://ldc.usb.ve/~gpalma/cursos/ci2691/guiagacela.html>
4. Fernández Michelle, Aponte Carlos, *Introducción a Pascal, Universidad Simón Bolívar, 2013*