

**UNIVERSIDAD MARIANO GALVEZ DE
GUATEMALA
SEDE PORTALES
FACULTAD DE INGENIERIA EN SISTEMAS DE
INFORMACION
CURSO COMPILADORES**



Code502

9989-20-273	Bryan Alexander Illescas Monroy
9989-20-4333	Josué Ezequiel Sanchez delgado
3190-14-17262	Anner Alester López López
9989-19-23686	Heber Jaziel Perla Rivas
9989-17-19285	Byron Estuardo Orozco Marroquín

CODE



502

Declaración de Variables

Para la declaración de variables en el lenguaje de Code502 se debe realizar de la siguiente forma: [NOTA_MUSICAL] = [NOMBRE] Se hace con base a las notas musicales de nuestro lenguaje

En este caso la (NOTA MUSICAL) hace referencia a las notas musicales y el [nombre] hace referencia a la variable que la persona que usa el programa hace a su elección, la declaración de una variable quedaría de la siguiente manera.

$$[NOTA_MUSICAL] + [NOMBRE] / \text{NotaNum numeroEntero} = 5;$$

Palabras clave

Las palabras reservadas en Code502 son palabras que tienen un significado especial para el compilador. No pueden usarse como nombres de variables, métodos, clases o cualquier otro identificador.

ALFABETO	
LETRAS	A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z A,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z
NUMEROS	0,1,2,3,4,5,6,7,8,9
CARACTERES	"" , ! , # , \$, % , / , (,) , = , + , * , [,] , { , }

DECLARACIÓN/DEFINICIÓN			
PALABRA CLAVE	CODE502	DEFINICIÓN	EJEMPLO
abstract	[NotAbstracta]	Se utilizan para implementar una abstracción	
class	[ClaseMusical]	Un tipo que define la implementación de un tipo particular de objeto	<pre>CancionPublicaClaseMusical MyClaseMusical { // Fields CancionPrivadaaint myField; // Constructor CancionPublicaMyClass(int field) { myField = field; } // Methods CancionPublicavoid myMethod() { System.out.println("Hello, World!"); } }</pre>

enum	[EnumerarNota]	Una palabra clave utilizada para declarar un tipo enumerado	<pre>EnumerarNota DayOfWeek { MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY }</pre> <pre>DayOfWeek today = DayOfWeek.MONDAY; System.out.println("Today is " + today);</pre>
extends	[ExtenderTempo]	Se utiliza en una declaración de clase para especificar la superclase, utilizado en una declaración de interfaz para especificar una o más superinterfaces	<pre>ClaseMusical Car ExtenderTempo Vehicle { int numberOfDoors; void accelerate() { System.out.println("The car is accelerating."); } }</pre>
final	[FinalCancion]	Defina una entidad una vez que no se puede cambiar ni derivar de más adelante.	
implements	[ImplementarRitmo]	Incluido en una declaración de clase para especificar una o más interfaces implementadas por la clase actual.	<pre>ClaseMusical MyClaseMusical ImplementarRitmo MyInterfaz { // ClaseMusical definition }</pre>
interface	[Interfaz]	Se utiliza para declarar un tipo especial de clase que solo contiene métodos abstractos o predeterminados	<pre>ClaseMusical MyClaseMusical ImplementarRitmo MyInterfaz { // ClaseMusical definition }</pre>
native	[RitmoNativo]	Se usa en declaraciones de métodos para especificar que el método no se implementa en el mismo archivo fuente, sino en otro idioma.	
strictfp	[EstrictoFP]	Una palabra clave utilizada para restringir la precisión y el redondeo de los cálculos de punto flotante para garantizar la portabilidad.	
synchronized	[SincronizarTempo]	Se utiliza en la declaración de un método o bloque de código para adquirir el bloqueo mutex para un objeto mientras el hilo actual ejecuta el código	<pre>SincronizarTempo (object) { // code to be executed in a SincronizarTempo manner }</pre>

CONTROL DE FLUJO

PALABRA CLAVE	CODE502	DEFINICIÓN	EJEMPLO
break	[PausaMusica]	Se utiliza para finalizar la ejecución en el cuerpo del bucle actual.	
case	[NotaCaso]	la palabra clave <code>NotaCaso</code> se utiliza como parte de la declaración <code>switch</code> . Ayuda a definir diferentes bloques de código que se ejecutarán en función del valor de una expresión determinada.	<pre> NotaCaso1: dayName = "Monday"; break; NotaCaso2: dayName = "Tuesday"; break; NotaCaso3: dayName = "Wednesday"; break; NotaCaso4: dayName = "Thursday"; break; NotaCaso5: dayName = "Friday"; break; default: dayName = "Invalid day"; } </pre>
ContinuarMusica	[ContinuarMusica]	Declaración de flujo de control que se utiliza dentro de los bucles para omitir la iteración actual y continuar con la siguiente iteración del bucle.	<pre> NotaPor(int i = 1; i <= 10; i++) { NotaSi(i % 2 != 0) { ContinuarMusica; // Skip odd numbers } System.out.println(i); } </pre>
default	[NotaDefecto]	<code>NotaDefecto</code> se utiliza para definir un método predeterminado en una interfaz.	<pre> CancionPublicaInterfaz MyInterfaz { NotaDefectovoid myMethod() { System.out.println("NotaDefectoimplementation of myMethod"); } } </pre>
do	[NotaDo]	La palabra clave <code>NotaDo</code> se utiliza como parte de la estructura del bucle <code>do-while</code> .	<pre> NotaDo{ // Code to be executed } MientrasSuena(condition); </pre>

else	[NotaAlternativa]	NotaAlternativa se utiliza para indicar las ramas alternativas en una declaración if.	<pre> NotaSi(num > 0) { System.out.println("The number is positive."); } NotaAlternativa{ System.out.println("The number is non-positive."); } </pre>
for	[NotaPor]	NotaPor se utiliza para crear una estructura de bucle que permite la ejecución repetida de un bloque de código.	<pre> NotaPor(initialization; condition; iteration) { // code to be executed } </pre>
if	[NotaSi]	NotaSi es una declaración de flujo de control que permite la ejecución condicional de un bloque de código. Se utiliza para especificar una condición y ejecutar un bloque de código solo si esa condición se evalúa como verdadera.	<pre> NotaSi(condition) { // code to be executed } </pre>
if	[NotaSi]	RegresarNota en se usa principalmente dentro de los métodos para salir del método y proporcionar un valor de resultado.	<pre> [NotaSi] (20 > 18) { [GeneraNota]("20 es mayor a 18"); } [NotaAlternativa] { [GeneraNota]("20 es menor a 18"); } </pre>
switch	[CambioRitmo]	CambioRitmo se utiliza para controlar el flujo de ejecución en función del valor de una variable o expresión. Le permite comparar el valor de la variable con múltiples etiquetas case y ejecutar diferentes bloques de código según el caso coincidente.	<pre> CambioRitmo(variable) { NotaCasovalue1: // code to be executed break; NotaCasovalue2: // code to be executed break; // more cases... default: // code to be executed } </pre>

while	[MientrasSuenas]	MientrasSuenas se usa para crear un bucle que ejecuta repetidamente un bloque de código siempre que una condición determinada sea verdadera.	MientrasSuenas(condition) { // code to be executed }
--------------	------------------	--	--

MODIFICADORES DE ACCESO			
PALABRA CLAVE	CODE502	DEFINICIÓN	EJEMPLO
private	[CancionPrivada]	Modificador de acceso que restringe la visibilidad de un miembro de la clase (variable, método o clase anidada) dentro de la misma clase.	<div>CancionPublicaClaseMusical MyClaseMusical {</div> <div> // Fields</div> <div> CancionPrivada myField;</div> <div></div> <div> // Constructor</div> <div> CancionPublicaMyClass(int field) {</div> <div> myField = field;</div> <div> }</div> <div></div> <div> // Methods</div> <div> CancionPublica void myMethod() {</div> <div> System.out.println("Hello, World!");</div> <div> }</div> <div> }</div>

CLASES Y MÉTODOS

PALABRA CLAVE	CODE502	DEFINICIÓN	EJEMPLO
this	[EsteRitmo]	es una referencia a la instancia del objeto actual dentro de un método o constructor.	EsteRitmo.nombre = nombre;
super	[SuperposicionRitmo]	la palabra clave SuperposicionRitmo se utiliza para referirse a la superclase inmediata (clase principal) de una subclase. Se puede utilizar para acceder a los métodos, campos y constructores de la superclase.	

MANEJO DE EXCEPCIONES:

PALABRA CLAVE	CODE502	DEFINICIÓN	EJEMPLO
catch	[CapturaSonido]	CapturaSonido se utiliza como parte del manejo de excepciones para detectar y manejar tipos específicos de excepciones que ocurren durante la ejecución de un programa.	<pre> IntentarSonido{ // Code that may throw an exception int result = divide(10, 0); System.out.println("Result: " + result); } CapturaSonido(ArithmeticException e) { // Exception handling NotaPorArithmeticException System.out.println("Error: Division by zero"); } </pre>
finally	[SonidoFinal]	finally se usa junto con el bloque IntentarSonido–CapturaSonido para definir un bloque de código que siempre se ejecutará, independientemente de si se lanza una excepción o no.	<pre> IntentarSonido{ // Code that may throw exceptions // NotaPorexample, opening a file or performing a database operation } CapturaSonido(IOException e) { // Exception handling code specific to IOException } SonidoFinal{ // Code that will be executed regardless of whether an exception occurred or not // NotaPorexample, closing a file or releasing resources } </pre>
throw	[LanzarSonido]	LanzarSonido se utiliza para lanzar explícitamente una excepción. Le sigue un objeto de excepción o una expresión que se evalúa como un objeto de excepción.	
throws	[LanzamientosSonidos]	LanzamientosSonidos se utiliza en la firma de un método para declarar que el método puede	

		generar una o más excepciones. Le siguen los nombres de las excepciones que puede generar el método.	
try	[IntentarSonido]	IntentarSonido es un componente fundamental del manejo de excepciones en code502. Se utiliza para definir un bloque de código donde pueden ocurrir excepciones.	IntentarSonido{ // Code that may throw exceptions // NotaPorexample, opening a file or performing a database operation } CapturaSonido(IOException e) { // Exception handling code specific to IOException } SonidoFinal{ // Code that will be executed regardless of whether an exception occurred or not // NotaPorexample, closing a file or releasing resources }

TIPOS DE DATOS PRIMITIVOS:			
PALABRA CLAVE	CODE502	DEFINICIÓN	EJEMPLO
boolean	[NotaBooleana]	Tipo de datos primitivo que se utiliza para declarar variables que pueden contener uno de dos valores posibles: true o false.	NotaBooleana esEstudiante = true;
byte	[NotaB]	Tipo de datos primitivo que se utiliza para declarar variables que pueden contener valores enteros que van desde -128 a 127. Representa un entero de 8 bits con signo.	NotaB diasMes = 31; //Aproximadamente
char	[NotaC]	se utiliza para declarar una variable que puede contener un único carácter Unicode. Representa un valor entero sin signo de 16 bits que oscila entre 0 y 65.535.	NotaC inicial = 'J';
double	[NotaDoble]	Ttipo de datos primitivo que representa un número de punto flotante de 64 bits.	NotaDoble e = 2.718281828459045235360;
float	[NotaFa]	Se utiliza para declarar una variable que puede contener un valor de punto flotante de 32 bits.	NotaFa pi = 3.1415926535f;
int	[NotaNum]	tipo de datos primitivo que se utiliza para declarar variables que almacenan valores enteros. Representa un entero en complemento a dos con signo de 32 bits, que va desde -	NotaNum edad = 30;

		2.147.483.648 a 2.147.483.647.	
long	[TempoLa]	utiliza para declarar un tipo de datos que puede contener un valor entero de 64 bits. Es un tipo de datos primitivo que representa un rango más amplio de valores en comparación con otros tipos de números enteros.	TempoLa añoLuz = velocidadLuz * 365;
short	[NotaCorta]	Se utiliza para declarar una variable que puede contener un valor entero con signo de 16 bits.	NotaCorta diasLustro = (12 * 31) * 5;
void	[NotaVacía]	Se utiliza para indicar que un método no devuelve un valor. Se utiliza como tipo de retorno al definir un método que realiza una tarea u operación, pero no produce ningún resultado.	

CONSTANTES			
PALABRA CLAVE	CODE502	DEFINICIÓN	EJEMPLO
const	letra	es un valor que se asigna a una variable y no cambia durante la ejecución del programa.	

CONTROL DE FLUJO EN ITERADORES			
PALABRA CLAVE	CODE502	DEFINICIÓN	EJEMPLO
assert	AfirmarTema	Describe un predicado (una declaración de verdadero / falso).	

CONTROL DE FLUJO EN BUCLES			
PALABRA CLAVE	CODE502	DEFINICIÓN	EJEMPLO
goto	IrTema		

CONTROL DE FLUJO EN CONDICIONALES			
PALABRA CLAVE	CODE502	DEFINICIÓN	EJEMPLO
enum	[EnumerarNota]	limitan la creación de objetos a los especificados explícitamente en la implementación de la clase.	<pre>EnumerarNota DayOfWeek { MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY }</pre> <pre>DayOfWeek today = DayOfWeek.MONDAY; System.out.println("Today is " + today);</pre>

MANEJO DE MEMORIA			
PALABRA CLAVE	CODE502	DEFINICIÓN	EJEMPLO
new	[NuevoTema]	Se utiliza para crear una instancia de una clase o para asignar memoria a un objeto. Se utiliza junto con un constructor para inicializar el objeto recién creado.	<pre>MyClaseMusical obj = NuevoTemaMyClass();</pre>

MANEJO DE PAQUETES			
PALABRA CLAVE	CODE502	DEFINICIÓN	EJEMPLO
import	[ImportarRitmo]	Se usa al comienzo de un archivo fuente para especificar clases o paquetes completos de Java para consultarlos más adelante sin incluir sus nombres de paquete en la referencia.	<pre>ImportarRitmo code502.util.Scanner;</pre>
package	[PaqueteRitmo]	Es un grupo de clases e interfaces similares. Los paquetes se declaran con la palabra clave del PaqueteRitmo.	<pre>ImportarRitmo PaqueteRitmo.ClaseMusical;</pre>

MANEJO DE PAQUETES			
PALABRA CLAVE	CODE502	DEFINICIÓN	EJEMPLO
Suma	+	Suma	
Resta	-	Resta	
Multiplicación	*	Multiplicación	
División	/	División	
Asignación	=	Asignación	

OTRAS PALABRAS RESERVADAS:

PALABRA CLAVE	CODE502	DEFINICIÓN	EJEMPLO
false	[Falso]	Un valor literal booleano.	
null	[Nulo]	Un valor literal booleano.	
true	[Verdadero]	Un valor literal booleano.	