

Proyecto compiladores, lenguaje Code502.

Fase 1#.

Grupo 6.

Universidad Mariano Galvez.



Integrantes:

Grupo 6#:

Índice:

- Declaración de Variables.
- Palabras Claves.
- Operadores.
 - Asignación (=, :=).
 - Aritméticos (*, + ...).
 - Igualdad y relacionales.
- Sentencias de Control (if, switch...etc.).
- Ciclos (while do while, for.... etc.).
- Modificadores de acceso (public, private ...etc.).
- Clases y atributos.
- Métodos.
- Impresión de información a consola o a una ventana emergente.

Definiendo la clase principal:

sonido_estatico = "static";

secuencia_melodica = "function";

in_de_la_composicion = "final";

interpretar = "return";

TIPOS DE DATOS

cadena_de_sonido = "String";

nota_musical = "char";

tono_entero = "int";

tono_flotante = "float";

SENTENCIAS DE CONTROL

conjunto_musical = "class";

si_suena_la_nota = "if";

si_no_suena_la_nota = "else";

variacion_de = "extends"

cambio_de_tonalidad = "switch";

caso_de_acorde = "case";

por_defecto_sonido = "default";

detener = "break";

armonioso = "true";

armonioso = "true";

continuar_tocando = "continue";

importar_partitura = "import";

exportar_partitura = "export";

intentar_tocar = "try";

atrapar_error_musical = "catch";

CICLOS

por_cada_compas_en_la_partitura = "for";

mientras_suena_la_melodia = "while";

Operadores

Nombre	Declaración de Variables:
Descripción	Declaración de variable tipo tono
Ejemplo	nota_musical notaDo = 'C'; cadena_de_sonido nombreCancion = "Canción de Prueba";

Nombre	Tipos de Datos:
Descripción	Tipo de dato para representar un tono
Ejemplo	tono_entero tonoDo = 60; tono_flotante duracionCompas = 4.0;

Operadores de Asignación

Nombre	Operadores de Asignación
Descripción	Asignación de tono
Ejemplo	<code>nota_musical nota = 'A';</code> <code>nota = 'B';</code>

Operadores Aritméticos

Nombre	Operador Aritmetico
Descripción	El operador Aritmetico hace que las variables hagan operaciones aritmeticas entre ellas.(suma, resta, multiplicación, division)
Ejemplo	<code>tono_entero suma =</code> <code>tonoDo + 4;</code> <code>tono_entero resta = tonoDo</code> <code>- 2;</code> <code>tono_entero multiplicacion</code> <code>= tonoDo * 2;</code> <code>tono_flotante division =</code> <code>duracionCompas / 2;</code>

Nombre	Operador Relacionales
Descripción	Utilizan los símbolos de menor y mayor para referirse a otra variable
Ejemplo	armonioso esIgual = (tonoDo == 60); discordante esMayor = (tonoDo > 64); discordante esMenor = (tonoDo < 64);

Nombre	Operador logicos
Descripción	Utilizan los símbolos para dar condiciones de verdadero o falso como respuesta
Ejemplo	armonioso condicion = (esIgual && esMayor);

Sentencias de control:

Condicional: La primera de las sentencias de control es el condicional, que ejecuta una instrucción si una condición es verdadera, conocido como “if/else” en varios lenguajes, en code502 la palabra reservada para esto es “si_suena_la_notas”:

Ejemplo:

```
si_suena_la_notas (nota == 'C') {  
    // Hacer algo si la nota es do  
} si_no_suena_la_notas {  
    // Hacer otra cosa si la nota no es do  
}
```

Estructura de casos: La otra sentencia de control son las estructuras de casos, que manejan varias condiciones para varios escenarios o casos, y tienen una opción predeterminada, en otros lenguajes es conocido como “switch/case”, las palabras reservadas para estas instrucciones en nuestro lenguaje son “cambio_de_tonalidad”:

Ejemplo:

```
cambio_de_tonalidad (nota) {  
    caso_de_acorde 'C':  
        // Hacer algo si la nota es do  
        descansar();  
        romper;  
    por_defecto_sonido:  
        // Hacer algo si la nota no es do
```

Ciclos:

Ciclo para: Como el nombre lo dice, ejecuta una instrucción por cada iteración indicada, conocido como “for” en otros lenguajes, es un ciclo finito, su palabra reservada es “por_cada_compas_en_la_partitura”.

Ejemplo:

```
por_cada_compas_en_la_partitura (entero i = 0; i < 10; i++) {  
    tocarNota(notas[i]);  
}
```

Ciclo mientras: También como el nombre lo dice, esto ejecuta una instrucción mientras que una condición esa verdadera, es un ciclo potencialmente infinito, en otros lenguajes se conoce como “while”, su palabra reservada es “mientras_suena_la_melodia”:

Ejemplo:

```
mientras_suena_la_melodia (contador < 10) {  
    tocarNota(do);  
    contador++;  
}
```


Modificadores de acceso

Public: Conocido en otros lenguajes como “public”, llamado “publico”, puede obtener acceso al tipo o miembro cualquier otro código del mismo ensamblado o de otro ensamblado que haga referencia a éste. El nivel de accesibilidad de los miembros públicos de un tipo se controla mediante el nivel de accesibilidad del propio tipo.

Ejemplo:

publico tono_flotante volumen = 0.5;

Private: En otros lenguajes se conoce como private, llamado “privado”, solamente el código de la misma clase o estructura puede acceder al tipo o miembro. **Ejemplo simple:**

privado tono_flotante tonoInicial = 60.5;

Clases y atributos:

Clases: Declaradas como “class” en otros lenguajes en code502 se declaran como “clase”, en pocas palabras las clases son plantillas para la creación de objetos.

Las clases se crean de la siguiente manera:

- colocando la palabra clase
- seguida de qué tipo de acceso tiene.
- y nombre de la clase

/ Definición de clase Instrumento

```
conjunto_musical Instrumento {
```

```
// Atributos
```

```
privado nota_musical nota;
```

Y los atributos son los que lo define

Lo cual da inicio a un constructor y para usar este constructor lo hacemos de la siguiente manera:

- colocando la palabra constructor
- seguida de un punto(.)
- seguida del nombre de la clase
- y dentro de paréntesis y separando por coma los datos.

```
Instrumento(nota_musical notaInicial) {  
    this.nota = notaInicial;  
}
```

Métodos:

Los métodos son bloques de código seguido de instrucciones para tareas específicas

Ejemplo

```
publico vacio tocar() {  
    tocarNota(nota);  
}  
}
```

Intancia:

Se va a referir a un objeto creado a partir de una clase y este mismo va a tener sus propias propiedades y que realiza sus propias acciones.

Ejemplo:

```
Instrumento guitarra = nuevo Instrumento('C');
```

La **clase** tiene un nombre que es “ejemplo”, seguido de un **estc** para declarar, métodos, propiedades, campos, etc. Para pertenecer a una clase y no ser una instancia, es decir no depender de otros.

espacio; esto indica cuando se llama el método, no va a esperar que devuelva ninguna información al código que lo llamó.

Acctotal: Esto es para iniciar el programa que entre en método de punto de entrada, es decir, el inicio de la aplicación.

cadena: Contiene los datos que van a representar, por ejemplo, la cadena va a ser el nombre.

argu: es una variable para almacenar argumentos entre las líneas de comandos que pasa a una aplicación cuando se ejecuta.

escribir: va a mostrar el mensaje de introduce tu nombre en la consola o vista.

leer; esto va a esperar que el usuario introduzca un nombre y será guardado.

Saludar(nombre): Esto va a llamar el método de saludar.

escribir(“Hola, ” + nombre): esto va mostrar en la consola el hola y el respectivo nombre que el usuario ingresó.

Ahora con otro ejemplo que suma 2 números:

```
clase sumaNumeros
{
    est espacio principal(cadena[] argu)
    {
        escribir("Introduce el primer número:");
        entero num1 = entero.convtn(Leer());

        escribir("Introduce el segundo número:");
        entero num2 = entero.convtn(Leer());

        entero resultado = Sumar(num1, num2);
        escribir("El resultado de la suma es: " + resultado);
    }

    Acctotal est entero Sumar(entero num1, entero num2)
    {
        entero resultado = num1 + num2;
        devolver resultado;
    }
}
```

entero: sería el número entero a evaluar.

entero.convtn: esto convierte la cadena de texto a número entero.

Vemos que en mostrar va pedir el primer y segundo número, que a la hora de ingresarlos, el entero.convtn y lo va guardar en Enseñar.

Luego en método Acctotal vamos a llamar el resultado, que vamos a sumarlos y va retornar el resultado.

Impresión de información a consola o a una ventana emergente:

Para mostrar en consola, se utilizará el **escribir** para enseñar los resultados en la consola, esto lo vimos en la sección de métodos por ejemplo que se uso el Mostrar

Ejemplo: **escribir("Hola mundo");**

Ventana emergente:

Para una ventana emergente se va a utilizar

ventana.mostrar("Hola, aqui esta el mensaje ")