

Definición del lenguaje

Jean Paul Yepes, Sebastian Ayala Suárez

Agosto 23, 2019

Índice

1. Descripción de la estructura de los programas	2
2. Tipos de datos	2
3. Variables y Constantes	2
4. Arreglos	3
5. Funciones.	3
6. Estructuras de control.	3
6.1. Si, sino y sinosi	3
6.2. Mientras	4
6.3. Para	4
7. Datos estructurados	4

1. Descripción de la estructura de los programas

El mínimo programa en este lenguaje de programación es:

```
entero main(){
    return 0;
}
```

2. Tipos de datos

Los tipos de datos que están disponibles en este lenguaje de programación son los siguientes:

- Entero: Se utiliza para guardar valores enteros, ya sea positivos o negativos en una variable que debe ser declarada, utilizando la palabra entero antes del nombre.
- Real: Se utiliza para guardar valores reales, o más bien, números de punto flotante en una variable que debe ser declarada, utilizando la palabra real antes del nombre.
- Caracter: Se utiliza para guardar un caracter en una variable que debe ser declarada, utilizando la palabra caracter antes del nombre.
- Cadena de caracteres: Se utiliza para guardar una cadena de caracteres en una variable que debe ser declarada, utilizando la palabra cadena antes del nombre.

Un ejemplo usando todos los tipos de variables que existen es:

```
entero main(){
    entero a = 0;
    real = 0.0;
    caracter = "a";
    cadena = "cadena";
}
```

3. Variables y Constantes

Todas las variables deben ser declaradas antes de poder utilizarse. Para declarar una variable se debe utilizar la siguiente estructura:

- Las variables deben empezar con una letra y debe estar compuesta por letras y numeros, además de las letras a...z, A...Z se puede usar guión bajo (_).
- Las variables deben ser nombradas
- Tipo de dato que tendrá la variable.
- Nombre de la variable.
- Igual “=”.
- Valor que tendrá la variable.

Ejemplo: `entero variable_1 = 10;`

Para declarar una constante se debe respetar la estructura que se tiene al declarar una variable, pero va precedida de la palabra constante.

Ejemplo: `constante real pi = 3.1416`

4. Arreglos

Para usar un arreglo en un programa, se debe poner el tipo de dato que se desea guardar en el arreglo seguido del tamaño entre corchetes “[]” y posteriormente el nombre del arreglo.

Para guardar un elemento en el arreglo, se debe poner el nombre del arreglo seguido de la posición deseada entre corchetes “[]” y finalmente la asignación y el valor que se desea guardar.

Ejemplo declaración: `entero[10] nombreArreglo;`

Ejemplo asignación: `nombreArreglo [0] = 10;`

5. Funciones.

Es un bloque de código con cero o más parametros de entrada, las cuales ejecuta una operación deseada y devuelve un valor o realiza una tarea. Para declarar una función se debe seguir la siguiente estructura:

```
tipoDeDato nombreFuncion (parametros) {  
    //acciones deseadas  
    return;  
}
```

Ejemplo:

```
entero suma (entero a, entero b) {  
    entero resultado = a + b;  
    return resultado;  
}
```

Nota: Las funciones que no devuelven ningún dato se les asigna como tipo de dato vacío.

```
vacio reiniciarArray (entero array[], entero tamaño ) {  
    para (entero i = 0; i < tamaño; i++) {  
        entero array[i] = 0;  
    }  
}
```

6. Estructuras de control.

6.1. Si, sino y sinosi

Esta es una estructura de decisión que evalúa una expresión lógica, utilizando la palabra si, con la expresión entre paréntesis y poniendo las acciones deseadas entre llaves “{ }”, y realiza unos pasos cuando la expresión es verdadera.

Si se desea realizar una acción en caso de que la expresión sea falsa, se debe usar la palabra sino y luego abrir llaves “{ }”.

Adicionalmente, se pueden evaluar otras expresiones utilizando la palabra sinosi en vez de sino, poniendo la nueva expresión a evaluar entre paréntesis y finalmente las acciones deseadas entre llaves “{ }”.

Ejemplo:

```

si (variable == 0) {
    //acciones deseadas.
}
sinosi (variable == 1){
    //acciones deseadas
}
sino{
    //acciones deseadas
}

```

6.2. Mientras

Es una estructura de repetición, que realiza una serie de acciones de manera repetitiva, siempre que, al evaluar una expresión lógica, esta resulte verdadera. Se debe usar la palabra *mientras*, seguida de la expresión lógica a evaluar puesta entre paréntesis y con las acciones deseadas entre llaves “{}”.

Ejemplo:

```

mientras (variable == 0) {
    //acciones deseadas
}

```

6.3. Para

Es una estructura que busca simplificar los ciclos. Utiliza un contador como base para repetir las acciones la cantidad de veces que se requiera. Se debe poner la palabra *para* seguida de la sintaxis (acción-inicial; condición-para-continuar-ciclo; acción-después-de-cada-iteración) y finalmente se ponen las acciones deseadas entre llaves “{}”.

Ejemplo:

```

para (int i = 0; i < 0; i = i + 1){
    //acciones deseadas
}

```

7. Datos estructurados

Las estructuras son una forma de agrupar datos y funcionalidades. Al crear una estructura, se creará un nuevo tipo de dato del tipo estructura, permitiendo crear nuevas instancias de ese tipo. Las estructuras llevan dentro los atributos que pertenecen a ella.

Para la declaración se debe anteponer la palabra *clase* al nombre de la misma y poner llaves “{}”.

Ejemplo:

```

estructura Humano {
    entero variable_uno;
    real variable_dos;
}

```

Para instanciar el objeto se debe instanciar con tipo de dato estructura, nombre de la estructura y el nombre de la variable.

Ejemplo:

estructura Humano humano1;