

CONSIDERACIONES METODOLÓGICAS PARA LA ENSEÑANZA DE LA PROGRAMACIÓN.

Autores: MSc. Marcos Antonio León Fonseca. (mleonf@udg.co.cu)
MSc. Juan Antonio Fonseca Hernández (jfonsecah@udg.co.cu)
MSc. Noralys Muñiz Maldonado. (nmunizm@udg.co.cu)
MSc. Yosvanys Casí Ladrón de Guevara. (ycasig@udg.co.cu)

Resumen:

En este artículo los autores proponen algunas consideraciones metodológicas para contribuir a que los estudiantes aprendan a programar.

Palabras claves: Programación, Recursividad.

Introducción:

El éxito de todo aprendizaje radica en lograr que el alumno penetre en la esencia de los fenómenos objeto de enseñanza, sin embargo, no siempre los métodos y medios que se utilizan en la enseñanza de un lenguaje de programación conducen a esto, y por ende, los niveles de conocimientos alcanzados no posibilitan a nuestros estudiantes enfrentar la solución de problemas de manera independiente.

El objetivo de este artículo es examinar qué se puede hacer para contribuir a que los alumnos aprendan a programar.

Desarrollo:

La enseñanza de la programación debe realizarse a través de la resolución de problemas donde el aprendizaje de cada nueva instrucción del lenguaje constituya una necesidad en la solución del problema a resolver, por lo que el tratamiento de los contenidos debe realizarse mediante la implementación predominante de los enfoques problémico y del problema base en combinación con el resto de los enfoques metodológicos.

El trabajo en el entorno de desarrollo del lenguaje de programación, debe estar precedido por un análisis consciente del problema a resolver basado en el conocimiento de métodos de trabajo y técnicas de programación tales como:

- *Enfoque de Divide y Vencerás:* Reflexionar acerca del problema a resolver y división en sus partes componentes. Esto permitirá ir buscando pequeñas soluciones que progresivamente integrarán la solución general. Para ello se deben elaborar procedimientos, que se convertirán a su vez en nuevas funciones que podrán ser utilizadas en el desarrollo de futuros programas (Reutilización del software).
- *Técnica de Abstracción:* Crear el programa que da solución al problema a partir de las funciones estandarizadas.
- Actualmente los lenguajes de programación no imponen restricciones en cuanto al número de caracteres a utilizar para denominar variables y procedimientos, por lo que es factible que su nomenclatura resulte una expresión del contenido de los valores que se guarden o de las acciones que se realizarán. Este estilo de trabajo debe constituir una exigencia permanente ya que facilita la socialización de los programas obtenidos.
- *Uso consecuente de la recursividad:* Permite reducir problemas con soluciones complejas a módulos sencillos relativamente fáciles de concebir y con una gran potencia desde el punto de vista de la programación.

Ejemplo:

1. *Planteamiento de la Situación Problémica donde el aprendizaje del nuevo contenido constituye una necesidad en la solución del problema a resolver.*

En el trabajo con las estructuras repetitivas, se resolvieron diferentes problemas en que se procesaban conjuntos de datos. Sin embargo, hasta el momento, solo nos interesó el valor final alcanzado por ciertas variables, que resumían y totalizaban a estos conjuntos de datos.

Es el caso, por ejemplo, del procedimiento elaborado para el cálculo del promedio de la cantidad de países que han votado, en la Asamblea General de las Naciones Unidas, a favor de una resolución que plantea la necesidad de poner fin al bloqueo económico impuesto al pueblo cubano por el gobierno de los Estados Unidos.

VOTACIONES EN LA ASAMBLEA GENERAL DE NACIONES UNIDAS CONTRA EL BLOQUEO			
AÑOS	A FAVOR	EN CONTRA	ABSTENCIÓN
1992	59	3	71
1993	88	4	57
1994	101	2	48
1995	117	3	38
1996	137	3	25
1997	143	3	17
1998	157	2	12
1999	155	2	8
2000	167	3	4
2001	167	3	3
2002	173	3	4
2003	179	3	2
2004	179	4	7
2005	182	4	1
2006	183	4	1
2007	184	4	1
2008	185	3	2
2009	187	3	2
2010	187	2	3
2011	186	2	3
2012	188	3	2
2013	188	2	3
2014	188	2	3
2015	191	2	0
2016	191	0	2

```

void CalcularPromedio ( ) {
    double Suma = 0.0;
    for (z = 0; z < 25; z ++){
        {cout << "Entre los votos: ";
        cin >> Voto;
        Suma += Voto;
        }
    }
    Promedio = Suma / 25;
}

```

}

En este caso, los valores de los votos no quedan registrados en memoria, solo el último valor que se entra, debido a que se utiliza en el ciclo la misma variable (Votos) para leer y almacenar los votos alcanzada en cada año ya que el ejercicio no exige el almacenamiento de los valores del conjunto de datos. Sin embargo, en ocasiones es necesario hacerlo, por ejemplo, supongamos que se nos exija:

Situación problemática: Determinar en cuantos años los votos a favor fueron superiores al promedio de países que votaron en la Asamblea General de las Naciones Unidas, en apoyo a la resolución que plantea la necesidad de poner fin al bloqueo económico impuesto al pueblo cubano por el gobierno de los Estados Unidos.

Planteamiento del problema docente:

Para darle solución al problema planteado, necesitamos comparar los votos a favor alcanzados en las votaciones de cada año, con el promedio de países que votaron a favor calculado por lo que es necesario que los votos estén registradas en la memoria de la computadora por lo que nuestra tarea consiste en:

Problema docente: Encontrar la forma de almacenar, en la memoria de la computadora, los distintos valores que toma una misma variable.

Existen en todos los lenguajes de programación estructuras que nos permiten almacenar valores en la memoria de las computadoras. Una de ellas son los arreglos lineales.

2. Análisis consciente del problema a resolver basado en el conocimiento de métodos de trabajo y técnicas de programación.

a) *Reflexión acerca del problema y división en sus partes componentes.*

La solución del problema conlleva a la realización de tres acciones fundamentales:

- Almacenar en la memoria de la computadora los votos a favor haciendo uso de los arreglos lineales.
- Calcular el promedio de países que votaron en apoyo a la resolución en todos estos años.
- Contar los años en que los votos a favor fueron superiores al promedio.

- b) *Elaboración de procedimientos donde se asignen nombres a las variables y los procedimientos cuya nomenclatura resulte una expresión del contenido de los valores que se guarden o de las acciones que se realizarán.*

```
void EntrarVotos () {  
    for (int z = 0; z < 25; z++)  
        {cout << "Entre los votos: ";  
        cin >> Voto [z];  
        }  
}
```

```
void CalcularPromedio () {  
    double Suma = 0.0;  
    for (z = 0; z < 25; z ++)  
        Suma += Voto [z];  
    Promedio = Suma / 25;  
}
```

```
int ContarAños () {  
    CalcularPromedio ();  
    CantidadAños = 0;  
    for (z = 0; z < 25; z++)  
        if (Voto [z] > Promedio)  
            ++CantidadAños;  
    return CantidadAños;  
}
```

3. Trabajo en el entorno de desarrollo del lenguaje de programación C++ para integrar los procedimientos elaborados en la solución del problema.

```
int Voto [25] = {0};  
double Promedio = 0.0;  
int CantidadAños = 0;
```

```
void Menu ();  
void EntrarVotos ();  
void CalcularPromedio ();  
int ContarAños ();
```

```
int main() {  
    int Opcion = 0;  
    Menu ();  
    cin >> Opcion;  
    while (Opcion != 3) {  
        switch (Opcion) {  
            case 1: EntrarVotos ();  
                break;  
            case 2: cout << "Cantidad de años en que los votos a favor fueron superiores al promedio de países que votaron en apoyo a la
```

```

        resolución: " << ContarAños ( ) << endl;
        break;
    default: cout << " Opci\242n no valida " << endl;
        break;
    }

Menu ();
    cin >> Opcion;
}

void Menu () {
    cout << " Men\243 " <<endl;
    cout << " 1 para entrar los votos a favor." << endl;
    cout << " 2 para consultar cantidad de pa\u00edses." << endl;
    cout << " 3 para salir." << endl;
    cout << " Opci\242n: ";
}

```

Conclusiones:

Es posible contribuir a que los alumnos aprendan a programar si se precede el trabajo en el entorno de desarrollo del lenguaje de programación por un análisis consciente del problema a resolver basado en el conocimiento de métodos de trabajo y técnicas de programación.

Bibliografía:

- Deitel, H. M. y Deitel., P. J. (2005). *C\u00f3mo Programar en C/C++*. Santiago de Cuba., PROGRAF.
- Lipschutz, S. (1989). *Estructura de datos*. Ciudad de la Habana., Edici\u00f3n Revolucionaria.