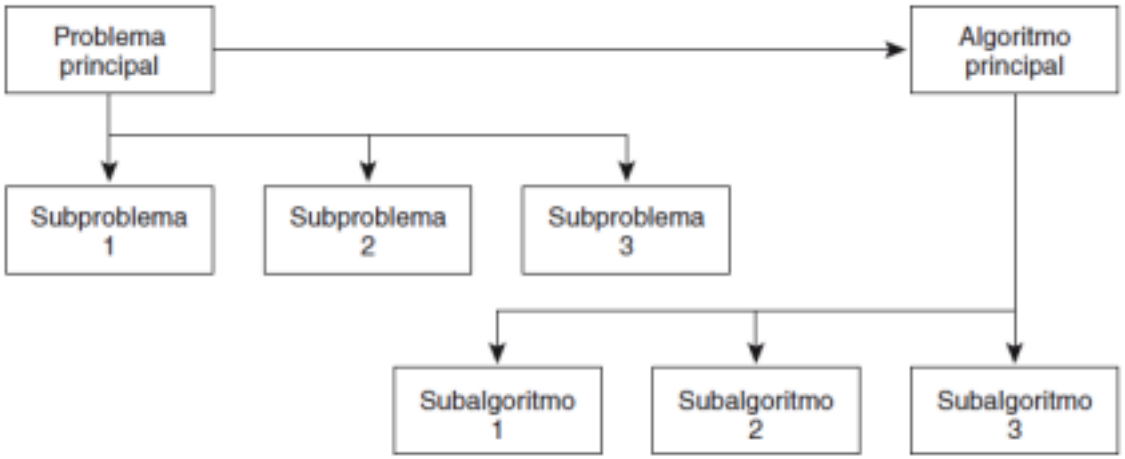
Material descargable

Fundamentos de la Programación



Repasemos.. ¿Qué son los subprogramas?

Un subproblema o subprograma se refiere a una porción de código dentro de un programa más grande que realiza una tarea específica. En PSeINT, **existen dos formas de implementar esta técnica**: **las funciones y los procedimientos**. Ambos pueden ser invocados desde cualquier parte del programa principal para ejecutar la tarea que han sido diseñados para realizar. Esto fomenta la reutilización del código y contribuye a organizar el programa en unidades más pequeñas y manejables.

Tanto las funciones como los procedimientos pueden recibir datos como entrada, procesarlos de acuerdo con una serie de instrucciones internas y, opcionalmente, devolver un resultado como salida. Esta capacidad los hace muy útiles para dividir problemas complejos en problemas más pequeños y manejables, facilitando así el desarrollo y la depuración del software.

En PSeInt, **la diferencia entre un procedimiento y una función radica principalmente en si devuelven o no un valor como resultado**. Si devuelve un valor, se les suele llamar funciones; si no devuelven ningún valor, se les suele llamar procedimientos o subprogramas. Sin embargo, ambos cumplen la misma función de encapsular una serie de instrucciones para llevar a cabo una tarea específica.

¿Qué son los procedimientos?

Los **procedimientos**, al igual que las funciones, son subprogramas diseñados para cumplir un fin específico dentro de un algoritmo. Pueden ser invocados tantas veces como sea necesario.

A diferencia de las funciones, los procedimientos no están obligados a retornar un valor. Sin embargo, si se necesita que un procedimiento modifique una variable, **se puede utilizar el paso de "parámetros por referencia",** lo que permite compartir la referencia de memoria de una variable. Esto significa que si un procedimiento modifica el valor de una variable, ese cambio se reflejará en el programa que lo invocó al retornar el control.

Aunque los procedimientos pueden realizar cálculos y actualizar valores en variables existentes, no pueden ser invocados dentro de otras funciones ni utilizados en expresiones, a diferencia de las funciones.

Formato general de un procedimiento:

- **Palabra reservada:** Subproceso
- **Nombre del procedimiento**: Nombre que utilizaremos para invocar el subprograma.
- **Conjunto de Parámetros**: Puede tener una cantidad variable de parámetros, que permiten que el programa se comunique, enviando información a través de ellos.

Dentro del procedimiento

- **Acciones:** Se pueden implementar diversas acciones dentro del bloque de código. Los procedimientos no están obligados a retornar un valor, pero pueden modificar variables externas si se utilizan parámetros por referencia.
- **Parámetros por referencia:** Si se necesita que el procedimiento modifique el valor de una variable, se deben definir parámetros por referencia. Esto permite que cualquier cambio realizado a la variable dentro del procedimiento se refleje fuera del mismo.

Ejemplo Uso de Procedimientos en PSeInt

```
Algoritmo EjercicioEjemploProcedimiento
    Definir varNumero1, varNumero2,varResultadoSubproceso Como Entero
    varResultadoSubproceso=0

    Escribir " QUERIDO USUARIO: Ingresa dos numeros: "
    Leer varNumero1, varNumero2

    calculandoSuma(varNumero1,varNumero2,varResultadoSubproceso) //Invoco al subproceso
FinAlgoritmo

SubProceso calculandoSuma(varNumero1,varNumero2, varResultadoSubproceso Por Referencia )
    varResultadoSubproceso = varNumero1 + varNumero2
    mensajeResultado(varResultadoSubproceso)
FinSubProceso

SubProceso mensajeResultado(varResultadoSubproceso)
    Escribir "El resultado de la suma de ambos numeros ingresados es:" , varResultadoSubproceso
FinSubProceso
```

Desglose del ejemplo:

- **Definición de variables**: En este paso, se definen las variables que se utilizarán en el algoritmo y los subprocesos. En este caso, se definen tres variables de tipo entero: `varNumero1`, `varNumero2` y `varResultadoSubproceso`. Estas variables se utilizarán para almacenar los números ingresados por el usuario y el resultado de la suma.¿Porque definimos una variable para resultado?, porque vamos a pasar la variable por referencia al procedimiento, así podemos “utilizarla” desde cualquier tora parte del programa con su valor actualizado.
- **Solicitud de entrada al usuario**: Se muestra un mensaje al usuario solicitando que ingrese dos números. Esto se logra mediante el comando `Escribir` para mostrar el mensaje en pantalla, seguido del comando `Leer` para obtener los valores ingresados por el usuario y almacenarlos en las variables `varNumero1` y `varNumero2`.
- **Invocación del subproceso**: En esta parte del algoritmo, se invoca al subproceso `calculandoSuma` pasando como argumentos los dos números ingresados por el usuario (`varNumero1` y `varNumero2`) y la variable `varResultadoSubproceso`. La invocación del subproceso se realiza mediante el nombre del subproceso seguido de los argumentos entre paréntesis.
- **Subproceso `calculandoSuma`**: Este subproceso recibe los dos números ingresados por el usuario (`varNumero1` y `varNumero2`) y la variable `varResultadoSubproceso` por referencia. En este subproceso, se calcula la suma de los dos números y se almacena en la variable

`varResultadoSubproceso`. La suma se realiza mediante el operador `+`. Luego, se llama al subproceso `mensajeResultado` pasando como argumento el resultado de la suma.

- **Subproceso `mensajeResultado`**: Este subproceso recibe como argumento el resultado de la suma (`varResultadoSubproceso`) y muestra un mensaje en pantalla con el resultado de la suma. Esto se logra mediante el comando `Escribir` para mostrar el mensaje en pantalla, seguido del resultado almacenado en la variable `varResultadoSubproceso`.

Beneficios de Usar Procedimientos

- **Modularidad**: Divide el código en partes más manejables y entendibles.
- **Reutilización**: Los procedimientos pueden ser reutilizados en diferentes partes del programa o en otros programas, la cantidad de veces que se necesite.
- **Mantenimiento**: Facilita el mantenimiento y la actualización del código.
- **Claridad**: Mejora la claridad y la legibilidad del código, haciendo que el propósito de cada parte sea más evidente.

Ámbito: Variables Locales y Globales

En la programación, es fundamental entender la distinción entre variables locales y globales, ya que influyen en la organización y el funcionamiento de nuestros programas.

Una variable local se declara y define dentro de un subprograma específico. Esto significa que solo está disponible dentro de este subprograma y no puede ser accedida directamente desde el programa principal o cualquier otro subprograma. Cuando se utiliza el mismo nombre de variable en diferentes subprogramas, cada uno se refiere a una ubicación de memoria diferente, lo que los hace independientes entre sí. Este aislamiento garantiza que los cambios realizados en una variable local en un subprograma no afecten a otras partes del programa.

Por otro lado, una variable global se declara en el programa principal y está disponible para todos los subprogramas. Estas variables son accesibles a través del paso de argumentos y pueden compartir información entre diferentes partes del programa sin necesidad de ser pasadas como parámetros. Sin embargo, su uso debe ser cuidadosamente considerado, ya que pueden complicar la comprensión y el mantenimiento del código al introducir dependencias entre diferentes partes del programa.

El uso de variables locales ofrece varias ventajas. En primer lugar, hace que los subprogramas sean independientes entre sí, ya que solo se comunican a través de

parámetros. Además, evita conflictos de nombres y facilita la gestión de la memoria al limitar el alcance de las variables a partes específicas del código.

Dependiendo del lenguaje de programación utilizado, la implementación de este concepto varía. Por ejemplo, en PSeINT, no se permite el uso de variables globales. En este caso, la única forma de "modificar" el valor de una variable declarada previamente en un subproceso es mediante el "Paso por Referencia".