MODULARIZACIÓN

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN

FACULTAD DE INGENIERÍA, PRODUCCIÓN Y SERVICIOS

Fundamentos de la Programación

Alumnos: Arocutipa Gutierres Luis Edgar
Coaquira Mamani Cesar Paul

Profesor: Escobedo Quispe Richart Smith

AGOSTO DEL 2020

Contenido

Contenido

Introducción

Los algoritmos están presentes en todas nuestras acciones, para resolver situaciones cotidianas tales como cruzar una calle, preparar una taza de té o leer un libro.

Contenido

Conceptos iniciales

Módulos o subprogramas: Son cada una de las acciones que ejecutamos en nuestro algoritmo y que luego vamos a desarrollar.

Funciones: Una función es un subprograma invocado desde el programa principal (o desde otra función) que ejecuta una tarea determinada y retorna el control al programa o función que la invocó.

Contenido

Función o método definido según el programador

Estructura General de una función:

```
[especificadores] tipoDevuelto nombreMetodo([lista parámetros]) [throws listaExcepcion
{
    // instrucciones
   [return valor;]
}
```

1. **Especificadores**
2. **TipoDevuelto**
3. **NombreMétodo**
4. **Lista de Parámetros**(opcional)
5. **throws listaExcepciones**
6. **return**

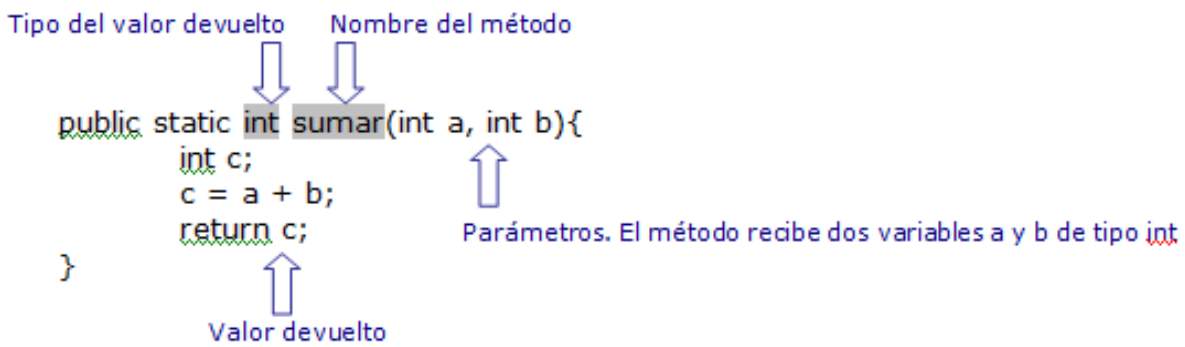
Función o método definido según el programador

Pasos para implementar un método

1. Describir lo que el método debe hacer.
2. Determinar las entradas del método, es decir, lo que el método recibe.
3. Determinar los tipos de datos de las entradas.
4. Determinar lo que debe devolver el método y el tipo del valor devuelto.
5. Escribir las instrucciones que forman el cuerpo del método.
6. Prueba del método: diseñar distintos casos de prueba.

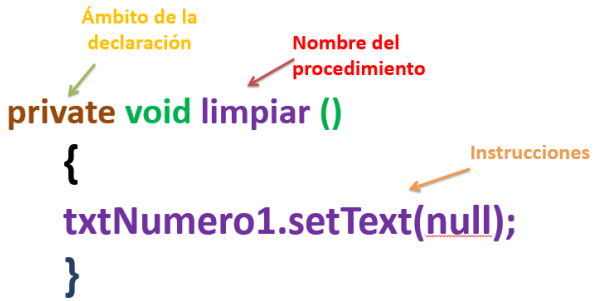
Invocar a una función: Cuando se llama a un método, la ejecución del programa pasa al método y cuando éste acaba, la ejecución continúa a partir del punto donde se produjo la llamada.

Ejemplo



Funciones definidas según el programador

Funciones de tipo void: Este tipo se utiliza para indicar que una función no retorna ningún valor.



Métodos no estáticos: Este tipo de métodos suelen ser usados más frecuentemente que los métodos estáticos. La diferencia radica en que estos métodos solo son corridos en un objeto y no en toda la clase como pasaba anteriormente.

```
public class Estudiante {
    private String nombre;
    public String getNombre() {
        return nombre;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
}
```

Funciones definidas según el programador

Método estático: Pertenecen a la clase(no están asociados a un objeto particular de la clase, por lo tanto no pueden acceder a las variables de instancia de un objeto(las cuales pertenecen a objetos particulares.

Constructor(this): Son métodos especiales que sirven para inicializar el estado de un objeto cuando lo creamos con el operador new.

-Su nombre ha de coincidir con el nombre de la clase.

-Por definición no devuelve nada.

-La palabra reservada this permite acceder al objeto sobre el que se ejecuta el método.

```
public class Contacto
{
  private String nombre;
  private String email;

  public Contacto (String nombre)
  {
    this.nombre = nombre;
  }

  public Contacto (String nombre, String email)
  {
    this.nombre = nombre;
    this.email = email;
  }

}
```

Contenido

Alcance de las variables

Variable de Instancia: Es una variable definida para las instancias de una clase (cada objeto tiene su propia copia de la variable de instancia y abarca los métodos no estáticos de una clase.

-**Cuando es privada:** Todos los métodos pueden acceder al valor almacenado.

-**Cuando es pública:** Se puede acceder a ella desde cualquier lugar que disponga de una referencia a un objeto.

Variable Local: Comienza en su declaración y termina donde termina el bloque de código() que contiene la declaración

```
void method ()
{
  int i=0;                // Declara e inicializa i

  while (i<10) {          // i está definido aquí
    int j=0;              // Declara j
    ...                   // i y j definidos
  }                       // j ya no está definido

  System.out.print(i); // i todavía está definido
}                         // i deja de estar definido
```

Contenido

Ejercicios

Ejercicio 01-Básico: Programa que da la Suma y Resta de dos Números.

```
public class Prueba{
    public static void Sumar(int a, int b){
        int c;
        c=a+b;
        System.out.println("El resultado de la suma es: "+c);
    }

    public static int Restar(int a, int b){
        int c;
        c=a-b;
        return c;
    }

    public static void main(String[] args){
        Sumar(15,20);
        int resta=Restar(35,26);
        System.out.println("El resultado de la resta es: "+resta);
    }
}
```

Ejercicios

Ejercicio 02-Básico: Programa que muestra la tabla de ´´un número ingresado por teclado

```
import java.util.Scanner;
public class Prueba1 {
 Scanner dato=new Scanner(System.in);
 public void multiplicar(int n1){
System.out.println("Ingrese el numero");
  n1 = dato.nextInt();
 for(int a=1;a<=12;a++){
 int resul=a*n1;
System.out.println(" º "+ a + " x " +n1+ " = " +resul);
 }
 }
public static void main(String[]args){
 Prueba1 tabla = new Prueba1();
 tabla.multiplicar(0);
}
}
```

Ejercicios

Ejercicio 03-Avanzado: Programa que imprime la cantidad de euros que tiene en una cuenta según ingresa o retira dinero. Clase Principal:

```
package Cuenta;
public class CuentaApp {

    public static void main(String[] args) {

        Cuenta cuenta_1 = new Cuenta("DiscoDurodeRoer");
        Cuenta cuenta_2 = new Cuenta("Fernando", 300);


        cuenta_1.ingresar(300);
        cuenta_2.ingresar(400);


        cuenta_1.retirar(500);
        cuenta_2.retirar(100);


        System.out.println(cuenta_1); // 0 euros
        System.out.println(cuenta_2); // 600 euros

    }

}
```

Ejercicios

Ejercicio 03-Avanzado: Clase Cuenta:

```java
package Cuenta;
public class Cuenta {
    private String titular;
    private double cantidad;
    public Cuenta(String titular) {
        this(titular, 0); //Sobrecarga
    }
    public Cuenta(String titular, double cantidad) {
        this.titular = titular;
        if (cantidad < 0) {
            this.cantidad = 0;
        } else {
            this.cantidad = cantidad;
        }
    }
    public String getTitular() {
        return titular;
    }

    public void setTitular(String titular) {
        this.titular = titular;
    }

    public double getCantidad() {
        return cantidad;
    }
```

```java
public void setCantidad(double cantidad) {
    this.cantidad = cantidad;
}
public void ingresar(double cantidad) {
    if(cantidad > 0){
        this.cantidad += cantidad;
    }
}
public void retirar(double cantidad) {
    if (this.cantidad - cantidad < 0) {
        this.cantidad = 0;
    } else {
        this.cantidad -= cantidad;
    }
}
@Override
public String toString() {
    return "El titular " + titular + " tiene "
     + cantidad + " euros en la cuenta";
}

}
```

Contenido

Referencias

Ing. Pablo Augusto Sznajdleder

Algoritmos a fondo : con implementaciones en C y Java . - 1a ed. - .

Fuente: https://www.academia.edu/40129651/Algoritmos_a_fondo_Con_implementaciones_en_C_y_Java

Programación Java/Métodos en Java

Enrique García Hernández.

Fuente: https://puntocomnoesunlenguaje.blogspot.com/2012/04/metodos.html?fbclid=IwAR3ypOIKgRNqyP4qvzyUk5MHz4xaU2IxQ2SNxtQLLoWVWzlhmMsNSI6J