

Asignatura Introducción a la Programación

Ingeniería Informática

Guía de Aprendizaje

Tema : Programación funcional.

Unidad didáctica: Llamada o invocación de las funciones y métodos de clase.

1. Objetivos de la unidad didáctica:

- Caracterizar la llamada o invocación de una función o método de clase.
- Identificar los principales elementos que la llamada o invocación de una función o método de clase.

2. Introducción a la unidad didáctica:

En guías de aprendizajes anteriores se había abordado que la mayoría de los programas de cómputo que resuelven los problemas reales son mucho más extensos que los programas que hemos desarrollado hasta ahora. La experiencia ha demostrado que la mejor manera de desarrollar y mantener un programa extenso es construirlo a partir de pequeñas piezas sencillas, o **módulos**.

Se había definido que existen tres tipos de módulos en Java: métodos, clases y paquetes. Y que el módulo más pequeño de los tres anteriores era: los métodos.

En guía anteriores teníamos como ejemplo de métodos de clase los siguientes métodos:

```
public static boolean desigualdadTriangular(double ladoA, double ladoB, double ladoC){  
    if (ladoA+ladoB>ladoC && ladoA+ladoC>ladoB && ladoB+ladoC>ladoA){  
        return true;  
    } else {  
        return false;  
    }  
}  
  
public static String clasificarTrianguloPorLados(double ladoA, double ladoB, double ladoC){  
    if (ladoA==ladoB && ladoA==ladoC){  
        return "Equilatero";  
    } else if ( (ladoA==ladoB && ladoA!=ladoC) ||  
        (ladoA==ladoC && ladoA!=ladoB) ||  
        (ladoC==ladoB && ladoA!=ladoC)){
```

```

        return "Isosceles";
    } else {
        return "Escaleno";
    }
}

public static void main(String[] args){
    double A,B,C,x1=0,x2=0;

    Scanner input = new Scanner(System.in);

    System.out.print("Entre el valor de A: ");

    A = input.nextDouble();

    System.out.print("Entre el valor de B: ");

    B = input.nextDouble();

    System.out.print("Entre el valor de C: ");

    C = input.nextDouble();

    if (desigualdadTriangular(A,B,C) == true){

        System.out.print("Con esos valores se puede conformar un triangulo ");

        System.out.println(clasificarTrianguloPorLados(A,B,C));

    } else {

        System.out.println("Con esos valores no se puede conformar un triangulo");

    }

}

```

Con la realización de la guías de aprendizaje anteriores se determino que los métodos o funciones de clases en su conformación o implementación se compone de dos secciones o partes esenciales: header/body. El header o cabecera declarativa de una función o método es la sección donde se define los elementos básicos y fundamentales de la función. Mientras en el body o cuerpo de función es donde se define los pasos e instrucciones del algoritmo que ejecutará el método o función una vez que sea invocado, estos pasos e instrucciones van a depender de cierta manera de lo que se defina en el header del método. Aplicando lo anterior en los métodos anteriormente descritos quedaría de la siguiente manera.

```

public static boolean desigualdadTriangular(double ladoA, double ladoB, double ladoC){

    if (ladoA+ladoB>ladoC && ladoA+ladoC>ladoB && ladoB+ladoC>ladoA){

        return true;

    } else {

        return false;

    }

}

public static String clasificarTrianguloPorLados(double ladoA, double ladoB, double ladoC){

```

```

        if (ladoA==ladoB && ladoA==ladoC){

            return "Equilatero";

        } else if ( (ladoA==ladoB && ladoA!=ladoC) || (ladoA==ladoC && ladoA!=ladoB) || (ladoC==ladoB && ladoA!=ladoC)){

            return "Isosceles";

        } else {

            return "Escaleno";

        }

    }

}

public static void main(String[] args){

    double A,B,C,x1=0,x2=0;

    Scanner input = new Scanner(System.in);

    System.out.print("Entre el valor de A: ");

    A = input.nextDouble();

    System.out.print("Entre el valor de B: ");

    B = input.nextDouble();

    System.out.print("Entre el valor de C: ");

    C = input.nextDouble();

    if (desigualdadTriangular(A,B,C) == true){

        System.out.print("Con esos valores se puede conformar un triangulo ");

        System.out.println(clasificarTrianguloPorLados(A,B,C));

    } else {

        System.out.println("Con esos valores no se puede conformar un triangulo");

    }

}

```

El texto con color rojo representa la sección el header o cabecera del método mientras el color verde representa el body o cuerpo de implementación.

Un programa o software puede estar conformado por varios métodos o funciones independientemente que sean de clase a o no. Para que un método se ejecute debe ser invocado durante la ejecución del programa. A como invocar o llamar un método de clase va estar destinada la presente guía de aprendizaje.

3. Orientaciones para el estudio:

1. Le sugerimos que para realizar el estudio de una materia debe elegir un lugar agradable, limpio, ventilado, cómodo, iluminado y si necesita recursos tecnológicos deben estar disponibles.
2. Cree hábitos de estudio sistemático; para esto:

- Planifique su tiempo y no deje para mañana lo que pueda hacer hoy; una buena planificación hace manejable las responsabilidades diarias que tenemos para con la vida, permite cumplir con todas las tareas programadas y así evita agobios de última hora.
- Calendarice las fechas más importantes de estudio y entrega de tareas. Localice con antelación los materiales que necesita para realizar el estudio individual o colaborativo.

3. Utilice técnicas de estudio:

- Elija un entorno de estudio que resulte agradable y sin elementos que lo que distraigan
- Estudie activamente, para esto lea en voz alta, tome notas, elabore esquemas o mapas conceptuales, realice resúmenes.
- Reflexione sobre lo que va aprendiendo, para esto relacione lo nuevo con lo anterior o conocido, asegúrese de que entiende y es capaz de aplicar lo que está aprendiendo antes de pasar adelante.

4. Interactúe con los materiales de estudio en tres fases:

- Fase de aproximación: Revise el objetivo de la unidad y después la acción o acciones a lograr; busque los materiales sugeridos para desarrollar la tarea, verifique cuál es el tiempo de que dispone.
- Fase de lectura profunda: aproxímese al material a través de una lectura ligera, poniendo especial interés en los títulos y subtítulos. Trate de relacionar lo que va leyendo con conocimientos adquiridos previamente.
- Fase de evaluación: Una vez realizada la lectura, intente realizar las actividades de auto-aprendizaje.

4. Requisitos Previos:

Tener conocimientos básicos de computación: los estudiantes matriculados deben conocer los procedimientos básicos para el manejo de PC y/o dispositivos móviles, aplicaciones de ofimática; así como uso de navegadores de páginas Web y el correo electrónico.

5. Actividades de auto-aprendizajes:

1. En el código presentado en la **Introducción a la unidad didáctica** se cuenta con tres métodos. Cuales de dichos métodos son invocados por otro método. Especifique en cada caso el método que realiza la invocación o llamada y el método que es llamado o invocado.

2. Cuantas veces puede ser invocado o llamado un método ?.

3. Que elementos del método se necesita conocer para poder realizar una invocación o llamada del mismo ?.

4. Cuando se invoca o se llama a un método y este recibe parámetros, en la invocación se debe tener presente los parámetros ?. En caso de tenerse en cuenta que relación debe existir entre los valores o variables utilizados como los parámetros en la invocación o llamada y los parámetros definidos en la declaración e implementación.

5. Como se invoca o llama a un método de clase que pertenece a otra clase diferente a la donde realizó la invocación o llamada. Pista: como se invoca o llama a los metodos de la clase *Math* de Java.

6. Puede ser la invocación o llamada de un método fungir como parametro de una invocación o llamada de otro método ?.

7. En el código presentado en la **Introducción a la unidad didáctica** cuantas invocaciones o llamadas a métodos se realizan. Cite cada una por el número línea de código.

6. Actividades de evaluación:

1.Declare e implemente una función o método de clase utilizando el lenguaje de programación Java que permita calcular y retornar la cantidad de combinaciones de N en K. Dicha función o método recibe por parámetro los valores de N y K. Haga un programa donde se invoque o llame a dicho método.

2.Declare e implemente una función o método de clase utilizando el lenguaje de programación de Java que permita calcular y retornar el factorial de N. Dicha función o método recibe por parámetro el valor de N. Haga un programa donde se invoque o llame a dicho método

3.Declare e implemente una función o método de clase utilizando el lenguaje de programación de Java que permita invertir un número entero N. Dicha función o método recibe por parámetro el valor de N. Ejemplo si la función recibe el numero 123 debe retornar el 321. Importante en el cuerpo del método o función solo se puede utilizar o declarar variables enteras. Haga un programa donde se invoque o llame a dicho método.

7. Resumen:

Para poder invocar o llamar a un método se necesita conocer el nombre o identificador del método así como los parámetros que recibe este. Cuando se invoca es necesario mantener la misma cantidad de parámetros y respetar el tipo de dato y el orden en que fueron declarados en el método. No existe limitaciones en cuanto a la cantidad de invocaciones que se le puede realizar a un método.

8. Glosario de términos:

Consulte la bibliografía y defina el concepto del siguiente término:

9. Bibliografía:

Aprenda Java como si estuviera en primero. *Colectivo de autores*. Cap 3 epígrafes 3.1, 3.2, 3.3, 3.4.

Cómo programar en Java. *Deitel, Paul J. Y Harvey M. Deitel*. Cap 6 completo

10. Recursos educativos digitales RED:

11. Próxima unidad: Funciones y métodos de clase recursivos.