Instituto Tecnológico de Estudios Superiores de Los Cabos

Nombre de la Carrera: ING. SISTEMAS COMPUTACIONALES

“Programas Lógica Funcional”

Asignatura:

Lógica Funcional

Docente: Fernando Antonio Delgadillo Hernandez

Grupo: 7IS-01M

Estudiante(s): Carlos Raúl López Arellano

Número de control: 21380066

Los Cabos, B.C.S., FECHA: 01/10/2024

Introducción

En este reporte, se presentará los programas que hemos hecho durante este tiempo en la materia de lógica funcional.

Veremos todo lo que se implementó en la programada de estos programas que fue la recursividad.

Índice

Contenido

Desarrollo

1. Reporte en PDF

2. Programas

¿Qué es la recursividad?

La recursividad es una característica de los lenguajes de programación que permite que un subprograma se invoque a sí mismo.

La recursividad es útil para resolver problemas definibles en sus propios términos.

La recursividad es, en cierta medida, análoga al principio de inducción.

No existen problemas intrínsecamente recursivos o iterativos; cualquier proceso iterativo puede expresarse de forma recursiva y viceversa.

La recursividad, aunque da lugar a algoritmos más simples y compactos resulta más lenta y consume más recursos al ejecutarse sobre el ordenador.

Primeros Programas:

Actividad:

Un programa donde pueda ingresar un número y que se muestre el resultado de su factorial y otro si es impar o par ese número, dejar el claro que no se pueden usar bucles solo funciones.

En este caso, yo hare los programas en el mismo código para que no tenga dos .java.

```java
import java.util.Scanner;

public class programa {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println(x:"Ingresa un número: ");
        int numero = sc.nextInt();

        System.out.println("El factorial de " + numero + " es: " + factorial(numero));

        if (esPar(numero)) {
            System.out.println("El número " + numero + " es par.");
        } else {
            System.out.println("El número " + numero + " es impar.");
        }

        System.out.println(x:"Ingresa la base y el exponente para calcular la potencia: ");
        int base = sc.nextInt();
        int exponente = sc.nextInt();
        System.out.println("La potencia de " + base + " elevado a " + exponente + " es: " + potencia(base, exponente));

        System.out.println("La suma de Gauss de " + numero + " es: " + sumaGauss(numero));

        System.out.println(x:"Ingresa dos números para hacer la multiplicación usando sumas: ");
        int a = sc.nextInt();
        int b = sc.nextInt();
        System.out.println("La multiplicación de " + a + " y " + b + " es: " + multiplicacion(a, b));

        if (esPrimo(numero, divisor:2)) {
            System.out.println("El número " + numero + " es primo.");
        } else {
            System.out.println("El número " + numero + " no es primo.");
        }

        sc.close();
    }
```

Primero que nada, utilizamos lo que es un Scanner para permitir que el usuario ingrese un número desde el teclado. Este número será almacenado en la variable número.

Ponemos lo que es la clase del main para que se ejecuta cuando inicias el programa.

Después llamamos a la función factorial(numero) para calcular la factorial del número ingresado por el usuario y mostramos el resultado en la consola.

```
public static int factorial(int n) {
    if (n == 0) {
        return 1;
    } else {
        return n * factorial(n - 1);
    }
}
```

En esta función es recursiva, usamos como base n no es 0, la función retorna n * factorial(n - 1), es decir, multiplica el número actual por el factorial del número anterior. Así por ejemplo, para n = 5, llamará a factorial (4), luego a factorial(3), y así sucesivamente, hasta llegar a factorial(0).

```
public static boolean esPar(int n) {
    return n % 2 == 0;
}

public static int potencia(int base, int exponente) {
    if (exponente == 0) {
        return 1;
    } else {
        return base * potencia(base, exponente - 1);
    }
}
```

Esta función usa el operador %, que devuelve el resto de una división. Si n % 2 == 0, entonces el número es par. Si no, es impar.

Luego otra función que es potencia para hacer los siguientes programas, pero lo que hace esto es si el exponente no es 0, multiplica la base por el resultado de la potencia con un exponente menor.

Segundos Programas:

- Función para calcular la potencia de un número.
- Función para calcular la suma de Guss (recursiva)
- Función para hacer una multiplicación usando sumas
- Función para determinar si un número es primo

```java
public static int sumaGauss(int n) {
    if (n == 0) {
        return 0;
    } else {
        return n + sumaGauss(n - 1);
    }
}

public static int multiplicacion(int a, int b) {
    if (b == 0) {
        return 0;
    } else {
        return a + multiplicacion(a, b - 1);
    }
}

public static boolean esPrimo(int n, int divisor) {
    if (n <= 2) {
        return (n == 2);
    }
    if (n % divisor == 0) {
        return false;
    }
    if (divisor * divisor > n) {
        return true;
    }
    return esPrimo(n, divisor + 1);
}
```

Y aquí están las otras funciones de los segundos programas, como había dicho, lo hice en un solo programa.

Resultados:

```
Ingresa un número:
6
El factorial de 6 es: 720
El número 6 es par.
Ingresa la base y el exponente para calcular la potencia:
2
2
La potencia de 2 elevado a 2 es: 4
La suma de Gauss de 6 es: 21
Ingresa dos números para hacer la multiplicación usando sumas:
4
4
La multiplicación de 4 y 4 es: 16
El número 6 no es primo.
PS C:\Users\RuLom\OneDrive\Documentos\Redes programa>
```

Conclusión

En conclusión, Estos programas me ayudaron a solo utilizar en forma recursiva y solo utilizar funciones para estas tipo de situaciones, también para entender cómo cada parte del programa interactúa y cómo se implementan las diferentes funciones de manera recursiva.

de Programación, A. y. L. (s/f). *Recursividad. Introducción*. Uniovi.es. Recuperado el 2 de octubre de 2024, de http://di002.edv.uniovi.es/~dani/asignaturas/transparencias-leccion6.PDF