
DESARROLLO DE ALGORITMOS

Trabajo Práctico de Repaso

IMPORTANTE: Diseñe algoritmos aplicando conceptos de modularización. Lea detenidamente el enunciado y trate de pensar cuales son los módulos que debe definir. En cada algoritmo no olvide especificar un nombre, definir comentarios apropiadamente y especificar el tipo de todas las variables utilizadas. Cada vez que diseñe un módulo tenga en cuenta en primera instancia su propósito, sus parámetros formales y valor de retorno. En cada invocación a un módulo, revise cuidadosamente los parámetros actuales, el orden y la compatibilidad de tipos entre parámetros actuales y formales.

Los siguientes ejercicios fueron tomados de correcciones realizadas a alumnos de esta facultad en años anteriores.

1. ¿Qué le falta a la definición de esta función?

```
/* * Calcula la distancia entre dos puntos de coordenadas, cada punto está  
formado por un X e Y */
```

```
public distanciaPuntos(xPto1,yPto1,xPto2,yPto2) { . . . }
```

2. ¿Qué problema tienen estos dos módulos?

```
(**es publicidad o no*)
```

```
MODULO esPublicidad (string tipo) RETORNO booleano
```

```
    SI (tipo="publicidad")ENTONCES
```

```
        ESCRIBIR ("True")
```

```
    SINO
```

```
        ESCRIBIR("False")
```

```
    FIN SI
```

```
FIN MODULO
```

```
/**
```

```
* Verifica si un valor es menor a otro del radio de una  
circunferencia para ver si esta dentro o fuera
```

```
* @param $radio
```

```
* return string
```

```
*/
```

```
public string circunferencia(float radioEscuela, float radio)
```

```
{
```

```
    if (radio<=radioEscuela)
```

```
    {
```

```
        System.out.println("dentro");
```

```
    }
```

```
    else
```

```
    {
```

```
        System.out.println("fuera");
```

```
    }
```

```
}
```

3. Si bien este código es correcto y funciona perfectamente, hay 3 cuestiones que se podrían mejorar. ¿Puede decir cuáles?

Tenga en cuenta la definición de estos módulos

MODULO costoPublicidad (string color, entero caracteres) RETORNO float

MODULO costoClasificado (entero lineas) RETORNO float

MODULO esPublicidad (string tipo) RETORNO boolean

PROGRAMA PRINCIPAL costoDelAviso

(*determina el costo del aviso*)

float letras, cantLineas, costoXLineas, precioDelAviso

string opcion, colorAviPub

ESCRIBIR "Ingrese el tipo de aviso: publicidad o clasificado"

LEER (opcion)

SI(opcion="publicidad") ENTONCES

 ESCRIBIR("Ingrese la cantidad de caracteres (letras):")

 LEER(letras)

 ESCRIBIR("Ingrese si el aviso es a color o negro:")

 LEER(colorAviPub)

 precioDelAviso ← costoPublicidad (colorAviPub, letras)

 ESCRIBIR("El costo del aviso es: precioDelAviso")

SINO

 ESCRIBIR ("Ingrese la cantidad de líneas que componen al aviso:")

 LEER(cantLineas)

 costoXLineas ← costoClasificado(cantLineas)

 ESCRIBIR ("El costo del aviso es: costoXLineas")

FIN SI

4. ¿Qué problema tiene este módulo?

(* Este módulo calcula el costo de un aviso clasificado*)

MODULO avisoClasificado (ENTERO cantLineas) RETORNA FLOAT

 FLOAT costo

 costo ← 150.00

 SI (cantLetras > 3) ENTONCES

 costo ← (150.00 (cantLineas – 3)*25.00)

 FIN SI

FIN MODULO

5. ¿Qué problema tiene este módulo?

(* Este módulo calcula si un punto se encuentra dentro de la circunferencia con centro (x1,y1)*)

MODULO pertenece(ENTERO x1, ENTERO y1, ENTERO x2, ENTERO y2) RETORNA
STRING

```
STRING resultado,FLOAT distancia
resultado <-- "fuera"
distancia <-- distanciaPuntos(x1,y1,x2,y2)
```

```
SI (distancia <= r) ENTONCES
    resultado <-- "dentro"
FIN SI
```

```
RETORNA resultado
```

FIN MODULO

6. ¿Qué error tiene este módulo?

(**dice si un punto se encuentra dentro o fuera del radio de una circunferencia dada*)

MODULO dentroFuera(Entero x1, y1, r, x2, y2) RETORNO String

```
Float distancia
distancia ← MODULO distanciaPuntos(ENTERO x1, y1, x2, y2)
String respuesta
SI(distancia <= r)ENTONCES
    respuesta← "dentro"
SINO
    respuesta← "fuera"
RETORNO respuesta
```

FIN MODULO

7. Este código tiene algo que se recomienda no hacer. ¿Qué es?

(**dice si es publicidad o no*)

MODULO esPublicidad(String tipoAviso)RETORNO boolean

```
SI ( tipoAviso = "publicidad")ENTONCES
    RETORNO true
```

SINO

RETORNO false

FIN MODULO

8. Este Pseudocodigo si bien funciona, se podría mejorar ¿Cómo?

PROGRAMA numerosDivisibles

ENTERO a, ENTERO b, ENTERO c

ESCRIBIR ("ingrese numero A")

LEER a

ESCRIBIR ("ingrese numero B")

LEER b

ESCRIBIR ("ingrese numero C")

LEER c

SI (esDivisible(a,c) AND esDivisible(b,c)) ENTONCES

ESCRIBIR ("Los numeros A y B son divisibles por C")

OTRO-SI (esDivisible(a,c) AND esDivisible(b,c) = false) ENTONCES

ESCRIBIR ("Solo el numero A es divisible por C")

OTRO-SI (esDivisible(a,c) = false AND esDivisible(b,c)) ENTONCES

ESCRIBIR ("Solo el numero B es divisible por C")

SINO

ESCRIBIR ("Ni A ni B son divisibles por C")

FIN SI

FIN PROGRAMA

9. Este código tiene algo que se no se debe hacer. ¿Qué es?

```
public static boolean verificarFrase(String frase)
{
    //verifica que una frase no este compuesta únicamente por espacios
    int limite,i;
    char aux;
    boolean verificado, esLetra;

    verificado=false;
    limite=frase.length()-1;

    for(i=0;i<=limite;i++)
    {
        aux=frase.charAt(i);
        esLetra=Character.isLetter(aux);

        if(esLetra==true)
        {
            i=limite;
            verificado=true;
        }
    }
    if(verificado==false)
    {
        System.out.println("no ingreso una oración válida");
    }
}
```

```
}  
    return verificado;  
}
```

Para los siguientes ejercicios: Diseñar los algoritmos, realizar traza e implementar en Java

1. Diseñar un algoritmo en pseudocódigo que lea un valor entero (N) y genere por pantalla los 10 primeros múltiplos del mismo.

Por ejemplo para N=7 deberá salir por pantalla: 7 14 21 28 35 42 49 56 63 70

2. Diseñar un algoritmo en pseudocódigo que dado un valor entero de 3 cifras (debe verificarlo) descomponga el número para mostrarlo a la inversa utilizando las operaciones MOD y DIV.

Por ejemplo si N=123 debe mostrar 321

3. Diseñar en pseudocódigo un algoritmo que permita leer números positivos hasta ingresar un valor negativo, y en base a ellos muestre:

- a) Cantidad de números pares
- b) Mayor número leído

4. Dado el siguiente código, indicar la información de la traza resultante

```
1  public class YourClassNameHere {  
2      public static void main(String[] args) {  
3  
4          int i=0, n=5, suma=0, suma2;  
5          for (i=1; i<=n; i++)  
6          {  
7              suma2=0;  
8              for (int j=1; j<=i; j++)  
9                  suma2=suma2+j;  
10  
11             suma=suma + suma2;  
12         }  
13  
14  
15         System.out.println(suma);  
16  
17     }  
18 }  
19 }
```

- a. El valor final de la variable suma2 es:
- b. El valor que se muestra por pantalla es:
- c. Cuando la variable i tiene el valor 3, cual es el valor final de la variable j?
- d. La cantidad de columnas de la traza es:
- e. El valor final de la variable i es:

- f. Cuando la variable *i* tiene el valor 4, cual es el valor que toma la variable suma al finalizar el for?
 - g. La variable *n* toma cuantos valores diferentes?
 - h. La variable *i* toma varios valores diferentes. Cuantos son?
5. Dado el siguiente código, indicar la información de la traza resultante

```
1 public class YourClassNameHere {  
2     public static void main(String[] args) {  
3  
4         int i=0;  
5         int suma=0;  
6         while (i<5)  
7         {  
8             suma+=i;  
9             i++;  
10            System.out.println( i + " - " + suma);  
11        }  
12    }  
13 }  
14 }
```

- a. La cantidad de columnas de la traza es:
 - b. El valor final de suma es:
 - c. La variable *i* toma diferentes valores, cuántos son estos valores?
 - d. El valor final de *i* es:
 - e. La variable suma toma diferentes valores, cuantos son estos valores?
6. Dado el siguiente código, indicar cual es el resultado final. Seleccione una de las siguientes opciones.

```
public class TestErrores {  
    public static void main(String[] args) {  
        int i;  
        for(i=0; i<5;i++){  
            System.out.println(i);  
        }  
    }  
}
```

a)	b)	c)	d)
0	0	1	5
1	1	2	
2	2	3	
3	3	4	
4	4		
	5		

7. Realice PseudoCódigo e implemente en Java de un algoritmo para imprimir una media pirámide como la siguiente. La cantidad de renglones depende de un valor n ingresado por el usuario (en el caso del ejemplo es 9)

```
1
21
321
4321
54321
654321
7654321
87654321
987654321
```

8. Realice Pseudocódigo e implemente en Java un juego en el que el programa elija un número aleatorio entre 1 y 100 y el usuario tenga que adivinarlo. Cada vez que el usuario ingresa un valor, si adivina el numero, entonces el programa finaliza con un mensaje de éxito. Si no lo adivina, le muestre pistas como “Demasiado Alto” o “demasiado bajo”. Utilice la estructura repetitiva más conveniente.
9. Diseñar en pseudocódigo un algoritmo que permita ingresar una frase por teclado, y en base a ella busque y muestre la palabra más larga. Suponga que entre las palabras hay un único espacio en blanco.
¿Cómo debería modificar el algoritmo para considerar que entre las palabras puede haber más de un espacio en blanco?
10. Diseñar un módulo en pseudocódigo que dada una frase y un caracter, verifique si el caracter ingresado se encuentra en la frase.
a. Implemente solo usando únicamente las operaciones longitud (length) y posicion (charAt)
b. Implemente usando las operaciones de String más adecuadas

Implementar el algoritmo llamador que invoque al módulo.

11. Diseñar un módulo en pseudocódigo que dada una frase y una palabra, cuente cuántas veces aparece la palabra en la frase.

Piense una implementación usando solo las operaciones longitud (length) y posición (charAt) y otra usando todas las operaciones de String (las que usted crea convenientes).

Implemente la mejor solución y realice traza.

12. Diseñar en pseudocódigo un algoritmo que permita ingresar una cierta cantidad de palabras (N) y en base a ellas muestre:

- a) Si existen palabras capicúas
- b) La cantidad de palabras con más de dos vocales
- c) La palabra de mayor longitud.

13. Para encriptar un mensaje se cambian las vocales por los siguientes símbolos: *, /, +, - y # (correspondientes a la 'a', 'e', 'i', 'o' y 'u' respectivamente).

Diseñe un módulo que lea un mensaje (una cadena) y genere un mensaje encriptado.

Diseñe el algoritmo que luego de invocar al módulo muestre por pantalla mensaje final.

14. En algunos sistemas informáticos solo se aceptan contraseñas que cumplan con ciertos requerimientos a fin de mejorar el nivel de seguridad de las mismas.

Diseñe un algoritmo que dada una contraseña verifique si cumple con las siguientes

condiciones:

- a. Tiene exactamente 8 caracteres
- b. Tiene al menos 1 letra
- c. Tiene al menos 1 número
- d. Tiene al menos 1 carácter: '*' '/' '-' '\$' '%' '#'