

SERVICIO NACIONAL DE APRENDIZAJE SENA  
SISTEMA INTEGRADO DE GESTIÓN  
Procedimiento Ejecución de la Formación Profesional Integral  
TALLER 2 JAVA –Métodos, Excepciones

FICHA:2558723 Julio 2023 Inst. Gisela Clavijo V

JULIAN DAVID GAMBA HERRERA

1. Escribe el resultado de ejecutar el siguiente fragmento de código:

```
public static double accesoPorIndice (double [] v, int j) throws  
    RuntimeException{  
    try{  
        if ((0 <= j) && (j <= v.length)){  
            return v[j];  
        }  
        else {  
            throw new Exception ("El indice " + j  
                                + " no existe en el vector");  
        }  
    }  
    catch (RuntimeException exc){  
        throw exc;  
    }  
}
```

Desde el siguiente "main":

```
public static void main(String [] args){  
  
    double [] v = new double [15];  
    accesoPorIndice (v, 16);  
}
```

**RESPUESTA:** El resultado será que al colocar un valor v que no se encuentra en el vector se activará la excepción "El indice " + j + " no existe en el vector" y el programa se detendrá

2. Escribe el resultado de ejecutar el siguiente fragmento de código:

```
public static double acceso_por_indice (double [] v, int j)  
    throws Exception{ try{  
        if ((0 <= j) && (j <=v.length)){  
            return v[j];  
        }  
        else {  
            throw new RuntimeException ("El indice " + j  
                                + " no existe en el vector");  
        }  
    }  
    catch (RuntimeException  
        exc){ throw exc;  
    }  
}
```

Desde el siguiente cliente "main":

```
public static void main(String [] args){  
  
    double [] v = new double [15];  
    acceso_por_indice(v, 16);  
}
```

**RESPUESTA:** El resultado será que al colocar un valor v que no se encuentra en el vector se activará la excepción "El índice " + j + " no existe en el vector" y el programa se detendrá

3. ¿Cuál sería la salida por pantalla del siguiente programa?

```
Public class ManejoExcp_1 {  
    private static int metodo ( ) {  
        int valor =0;  
        try {  
            valor = valor +1;  
            valor = valor +  
                Integer.parseInt("42");  
            valor = valor + 1;  
            System.out.println ( "Valor al final del try : " + valor ) ;  
        }  
        catch (NumberFormatException e ) {  
            valor = valor + Integer.parseInt("42");  
            System.out.println ( "Valor al final del catch : " + valor) ;  
        }  
        finally {  
            valor = valor + 1;  
            System.out.println ( "Valor al final de finally : " + valor) ;  
        }  
        valor = valor + 1;  
        System.out.println ( "Valor antes del return : " + valor);  
        return valor;  
    }  
  
    public static void main (String [ ] args) {  
        try {  
            System.out.println (metodo ( ));  
        }  
        catch (Exception e) {  
            System.out.println ( "Excepcion en metodo ( ) " );  
            e.printStackTrace ( );  
        }  
    }  
}
```

**RESPUESTA:**

**Valor al final del try : 44**

**Valor al final de finally : 45**

**Valor antes del return : 46**

Esa sería la salida por pantalla puesto que valor empieza en 0 y se le suma 1, ahora valor = 1. Luego a valor se le suma la cadena convertida a entero 42, ahora valor es 43, se le vuelve a sumar 1 y se imprime "Valor al final del try : 44". luego no entramos en el bloque del catch, seguimos y en el bloque finally se suma 1 y se imprime "Valor al final de finally : 45". Luego al salir del finally se le vuelve a sumar 1 y se imprime "Valor antes del return : 46". El método devuelve el valor 46 que es el que se imprime en el main.

4. ¿Cuál sería la salida por pantalla del siguiente programa?

```
Public class ManejoExcp2 {
    private static int metodo ( ) {
        int valor =0;
        try {
            valor = valor +1;
            valor = valor + Integer.parseInt("W");
            valor = valor + 1;
            System.out.println ("Valor al final del try : " + valor ) ;
        }
        catch (NumberFormatException e ) {
            valor = valor + Integer.parseInt("42");
            System.out.println ("Valor al final del catch : " + valor) ;
        }
        finally {
            valor = valor + 1;
            System.out.println ("Valor al final de finally : " + valor) ;
        }
        valor = valor + 1;
        System.out.println ("Valor antes del return : " + valor);
        return valor;
    }

    public static void main (String [ ] args) {
        try {
            System.out.println (metodo ( ));
        }
        catch (Exception e) {
            System.out.println ("Excepcion en metodo ( ) " );
            e.printStackTrace ( );
        }
    }
}
```

**RESPUESTA:**

**Valor al final del catch : 43**

**Valor al final de finally : 44**

**Valor antes del return : 45**

**45**

Esa sería la salida por pantalla puesto que el valor empieza en 0 y se le suma 1, ahora valor = 1. Luego a valor se le intenta sumar la cadena "W" lo cual no es posible y se va al bloque catch donde continúa. En el bloque catch se le suma 42 a valor, ahora valor es 43 y se imprime "Valor al final del catch: 43". luego entramos en el bloque del finally se suma 1 y se imprime "Valor al final de finally : 44". Luego al salir del finally

se le vuelve a sumar 1 y se imprime "Valor antes del return : 45". El método devuelve el valor 45 que es el que se imprime en el main.

5. ¿Cuál sería la salida por pantalla del siguiente programa?

```
Public class ManejoExcp_3 {  
private static int metodo ( ) throws  
NumberFormatException{ int valor =0;  
try {  
valor = valor +1;  
valor = valor +Integer.parseInt("W");  
valor = valor + 1;  
System.out.println ("Valor al final del try : " + valor ) ;  
}  
catch (NumberFormatException e ) {  
valor = valor + Integer.parseInt("42");  
System.out.println ("Valor al final del catch :  
" + valor) ; throw e;  
}  
finally {  
valor = valor + 1;  
System.out.println ("Valor al final de finally : " + valor) ;  
}  
valor = valor + 1;  
System.out.println ("Valor antes del return : " + valor);  
return valor;  
}  
  
public static void main (String [ ] args) {  
try {  
System.out.println (metodo ( ));  
}  
catch (Exception e) {  
System.out.println ("Excepcion en metodo ( ) " );  
e.printStackTrace ( );  
}  
}  
}
```

**RESPUESTA:**

**Valor al final del catch : 43**

**Valor al final de finally : 44**

**Valor antes del return : 45**

**Excepcion en metodo ( )**

Esa sería la salida por pantalla puesto que el valor empieza en 0 y se le suma 1, ahora valor = 1. Luego a valor se le intenta sumar la cadena "W" lo cual no es posible y se va al bloque catch donde continúa. En el bloque catch se le suma 42 a valor, ahora valor es 43 y se imprime "Valor al final del catch: 43". luego entramos en el bloque del finally se suma 1 y se imprime "Valor al final de finally : 44". Luego al salir del finally se le vuelve a sumar 1 y se imprime "Valor antes del return : 45". Luego se pasa al bloque catch donde se imprime "Excepcion en metodo" y se muestran las llamadas de la excepción.

6. Escribe un método auxiliar de nombre "**caracterEntrada**" en Java que realice la siguiente acción:

- Recibe como parámetros una cadena (String) y un entero (int);

- Si el entero está entre 0 y la longitud de la cadena (puedes hacer uso del método "length()" de la clase String) devuelve el carácter en la posición correspondiente (puedes hacer uso del método "charAt(int)" de la clase String).

- En caso contrario, construye y lanza una excepción de tipo "Exception".

7. Construye un programa principal en Java que realice las siguientes acciones:

- Crea un objeto de la clase "Scanner" y lo asocia con la entrada estándar (la consola de MSDOS);

- Lee un objeto de tipo String de la misma en un objeto "lect\_teclado";

- Invoca al método "**caracterEntrada**" definido en el ejercicio anterior, con la variable String leída de la entrada estándar y el entero "7", mostrándolo por pantalla; - Captura la posible excepción, mostrando por pantalla un mensaje:

*"Has intentado recuperar una posición de la cadena de caracteres que no existe;"*.

3. Revisar y mejorar los ejercicios realizados en el Taller 1 de java, para añadirles el manejo de excepciones y que las operaciones realizadas sean métodos.