Clase 5

PROGRAMACIÓN 1

Objetivos del tema

- Describir el funcionamiento de las sentencias iterativas o bucles (for, while y do-while)
- Interpretar el resultado de una secuencia simple o combinada de sentencias de control
- Codificar una tarea utilizando la secuencia y combinación de sentencias iterativas y condicionales

Bucle o sentencia repetitiva

- Un bucle o sentencia repetitiva se utiliza cuando se requiere realiza una o muchas tareas varias veces (ninguna, una o más).
- En un bucle o sentencia repetitiva hay una expresión lógica (**condición simple o múltiple**) de terminación que:
  - si es cierta, ejecuta las sentencias entre llaves, y posteriormente vuelve a verificar la expresión lógica de terminación.
  - si es falsa, sale del bucle.

```
Ciclo (expresion logica){
    sentencia 1;
    …
    sentencia N;
}
```

Bucle o sentencia repetitiva

- Hay casos donde la expresión lógica de terminación puede ser evaluada luego de ejecutar una vez las sentencias entre llaves

```
{
    sentencia 1;
    …
    sentencia N;
} Ciclo (expresion logica)
```

Sentencia for

- La sentencia for **SOLO** se utiliza cuando se conoce exactamente la cantidad de iteraciones.

```
for (valor de la iteracion inicial; expresion
  logica simple que incluye la iteracion actual;
  incremento de la iteracion en cada paso) {
  sentencia_1;
  sentencia_2;
  …
  sentencia_n;
}
```

Ejemplo

```
//Sentencia for
public class Programa {
 public static void main (String [] args) {
  final int MAX = 10;
  final int MULTIPLO = 5;
  System.out.println("Tabla de multiplicar del
       " + MULTIPLO);

  //Se puede declarar o no la variable i dentro del
  //inicio del for

  for (int i = 1 ; i <= MAX; i++) {
       System.out.println(MULTIPLO + " * " + i +
       " = " +         MULTIPLO *i );
  }
  }
}
```

Ejemplo

```
//Sentencia for anidada
public class Programa {
 public static void main (String [] args) {
  final int MAX = 10;
  final int MULTIPLO = 3;
  System.out.println("Tablas de multiplicar del
      1, 2 y 3");
  for (int i=1; i<=MULTIPLO; i++) {
      System.out.println("Tabla de multiplicar
      del " + i);
      for (int j=1; j<=MAX; j++) {
             System.out.println(j + " * " + i + "
      = " + j*i );
      }
  }
  }
}
```

Sentencia while

- Es un bucle o sentencia repetitiva con una expresión lógica **simple o múltiple** al principio. Se ejecutan las sentencias entre llaves mientras sea cierta la expresión lógica. La sentencia puede que no se ejecute ni una sola vez.
- La sentencia while se utiliza cuando se conoce exactamente o no la cantidad de iteraciones.

```
[inicializacion;]
while (expresion logica no estatica que
  varia segun las sentencias internas) {
  sentencia_1;
  …
  sentencia_n;
}
```

Ejemplo de problema

Para este problema deberá utilizar todos los tipos de sentencias desarrolladas hasta el momento.

- Utilizando una sentencia repetitiva calcular el capital acumulado a 10 años para un capital inicial de 100 y una tasa de interés de 4%.

Ejemplo

```
//Sentencia while, puede no ejecutarse
public class Programa {
 public static void main (String [] args) {
  final int MAX = 10;
  final double interes = 4;
  double capital = 100;
  int anios = 1;
  while (anios < MAX) {
      capital += capital*interes/100;
      anios++;
  }
  System.out.println("Capital final es = " +
  capital);
  }
}
```

Ejemplo sentencia do-while

```
//do - while, se ejecuta al menos una vez
public class Programa {
 public static void main (String [] args) {
  final int MAX = 10;
  final double interes = 4;
  double capital = 100;
  int anios = 1;
  do  {
      capital += capital*interes/100;
      anios++;
  } while (anios < MAX);
  System.out.println("Capital final es = " +
  capital);
  }
}
```

Ejemplo de problema

Para este problema deberá utilizar todos los tipos de sentencias desarrolladas hasta el momento.

- Ingresar un valor positivo válido.

Ejemplo valor positivo con while

```
import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Programa {
  public static void main(String[] args){
    int valori = 0;
    BufferedReader entrada = new BufferedReader(new
  InputStreamReader(System.in));
    while (valori<=0){ //los valores positivos no incluyen al 0
      try { //try define un bloque de manejo de posibles excepciones
        System.out.println ("Ingrese valor positivo: ");
        valori = new Integer(entrada.readLine());
        System.out.println("int : "    + valori);
      }
      catch (Exception exc ) {
        System.out.println( exc );
        valori = 0;
      }
    }
  }
}
```

Ejemplo valor positivo c/do while

```
import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Programa {
  public static void main(String[] args){
    int valori = 0;
    BufferedReader entrada = new BufferedReader(new
  InputStreamReader(System.in));
    do {
      try { //try define un bloque de manejo de posibles excepciones
        System.out.println ("Ingrese valor positivo: ");
        valori = new Integer(entrada.readLine());
        System.out.println("int : "    + valori);
      }
      catch (Exception exc ) {
        System.out.println( exc );
        valori = 0;
      }
    }while (valori<=0);
  }
}
```

En la excepción se debe reinicializar la variable de control debido a que se desconoce que valor tiene asociado producto de la excepción

Ejemplo de problema

Para este problema deberá utilizar todos los tipos de sentencias desarrolladas hasta el momento.

- Ingresar un valor válido.

Ejemplo carga sin valor inicial

```
import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Programa {
   public static void main(String[] args){
     int valori;
     boolean valorvalido = false;
     BufferedReader entrada = new BufferedReader(new
   InputStreamReader(System.in));
     do {
        try { //try define un bloque de manejo de posibles excepciones
           System.out.println ("Ingrese valor valido: ");
           valori = new Integer(entrada.readLine());
           System.out.println("int : "    + valori);
           valorvalido = true;
        }
        catch (Exception exc ) {
           System.out.println( exc );
           valorvalido = false;
        }
     }while (!valorvalido);
   }
}
```

Se desconoce que valor inicial de la variable, entonces va tomar el primer valor ingresado por el usuario. Además se usa una variable booleana para controlar la excepción

Dado un valor valido hacer…

```
…
public class Programa {
  public static void main(String[] args){
   Definir todas las constantes y variables
   BufferedReader entrada = new BufferedReader…;
   ingresar valor valido
   procesar o generar resultados
  }
}
```

Ejemplo de problema

Para este problema deberá utilizar todos los tipos de sentencias desarrolladas hasta el momento.

- Hacer un programa que dado un valor ingresado por el usuario entre 1 y 3 inclusive (**si ingresa otro valor siempre deberá repetir el proceso de ingreso**), imprima como salida “Bajo” en el caso de que ingrese 1, “Medio” si ingresa 2, y “Alto” si ingresa 3.

Solución

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Programa1 {
  public static void main(String[] args){
      //DECLARACION DE CONSTANTES
      final int MINIMO     = 1;
      final int MAXIMO     = 3;
      //DECLARACION DE VARIABLES
      int valori     = 0;
      //INGRESAR VALOR VALIDO
      BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
      do {
          try {
              System.out.println ("Ingrese integer entre 1 y 3: ");
              valori = new Integer(entrada.readLine());
          }
          catch (Exception exc) {
              System.out.println(exc);
              valori = 0; //SE REINICIA EL VALOR EN CASO DE HABER OCURRIDO UN ERROR
          }
      } while (!((valori >= MINIMO)&&(valori <= MAXIMO)));//CONDICIÓN MÚLTIPLE
      //PROCESAR
      switch (valori) {
          case 1: System.out.println ("Bajo"); break;
          case 2: System.out.println ("Medio"); break;
          case 3: System.out.println ("Alto"); break;
      }
  }
}
```

Práctico

- Escribir un programa que dado un número natural ingresado por el usuario imprima la tabla de multiplicar de ese número.
- Escribir un programa que dado un valor ingresado por el usuario menor que 10 y mayor a 1, muestre por pantalla una cuenta regresiva de números desde dicho valor hasta el 0 inclusive.
- Escribir un programa que solicite el ingreso de un número mayor a 50, y los muestre por pantalla en caso de ser múltiplo de 2 o 3.
- Escribir un programa donde el usuario ingrese un número entre 0 y 999, y muestre un mensaje de cuántos dígitos tiene. Además, si tiene 3 dígitos debe informar qué número es.

Resolver mientras que...

```
…
public class Programa {
  public static void main(String[] args){
   Definir todas las constantes y variables
   BufferedReader entrada = new BufferedReader…;
   while (no se cumpla expresión lógica de salida){
     ingresar valores validos vinculados a la
  expresión lógica
     if (no se cumpla expresión lógica de salida){
       ingresar otros valores validos
       generar resultados
     }
   }
  }
}
```

Resolver mientras que...

```
…
public class Programa {
  public static void main(String[] args){
   Definir todas las constantes y variables
   BufferedReader entrada = new BufferedReader…;
     do{
       ingresar valores validos vinculados a la expresión
    lógica
       if (no se cumpla expresión lógica de salida){
         ingresar otros valores validos
         generar resultados
       }
     }
     while (no se cumpla expresión lógica de salida);
   }
  }
}
```

Práctico

- Escribir un programa que mientras que el usuario ingrese un número distinto de 0, pida ingresar otro numero y lo imprima.

- Escribir un programa que mientras que el usuario ingrese un número distinto de 0, pida ingresar otros dos números e imprima el resultado de la multiplicación entre los dos últimos números ingresados.

- Construir un programa que solicite desde teclado un número de mes válido y posteriormente notifique por pantalla la cantidad de días de ese mes. En el caso de que ingrese 2 como número de mes (febrero) deberá además solicitar ingresar un número de anio entre 2000 y 2019 inclusive, y dependiendo de si es bisiesto o no imprimir la cantidad de días correspondiente.