Programación Orientada a Objetos

Casting y

Métodos de Ordenamiento

Conversión de tipo (TYPE CASTING)

- Java permite que una variable definida de un tipo pueda convertirse en otro tipo.
- INT a LONG
  - int numEntero = 23;
  - long numLargo = (long)numEntero;
- INT a FLOAT
  - int num2 = 23;
  - float flotante1 = num2;
- BYTE a INT
  - byte numBait = -32;
  - int numEntero = numBait;
- BYTE a SHORT
  - short num = 13000;
  - byte numBait = 19;
  - num = numBait; //OK

Conversión de tipo (TYPE CASTING)

- INT a BYTE
  - int num3 = 45;
  - byte numBait = (byte)num3;
  - System.out.println("El numero ingresado es: " + numBait); //aparece 45 en pantalla
  - int num3 = 128;
  - byte numBait = (byte)num3;
  - System.out.println("El numero ingresado es: " + numBait); //aparece -128 en pantalla
  - int num3 = 130;
  - byte numBait = (byte)num3;
  - System.out.println("El numero ingresado es: " + numBait); //aparece -126 en pantalla

Conversión de tipo (TYPE CASTING)

- INT a SHORT
  - int num1 = 10;
  - short s = num1; //Error
  - short s = (int)num1;//OK

- LONG a SHORT
  - short s=0;
  - long numLargo = 200L;
  - s = (short)numLargo; //OK

Conversión de tipo (TYPE CASTING)

- INT a DOUBLE
  - int num = 100;
  - double d1 = num;
- FLOAT a DOUBLE
  - float f1 = 0.34f;
  - double d2 = f1;
- CHAR a DOUBLE
  - double d3 = 'A'; //Asigna 65.0 a d3
- DOUBLE a INT
  - int num = 200;
  - double d3 = 3.142;
  - num = d3; //ERROR
  - num = (int)d3; //OK

Ordenando un arreglo

- **Método de la burbuja**
- Empezamos del primer elemento y lo comparamos con el siguiente, de ser necesario, realizar el intercambio de lugar para que el elemento menor siempre quede adelante y volver a iniciar hasta que todo el arreglo se ordene.

Ordenamiento con el método de la burbuja

```
int num=10, i=0;
      long arreglo [], temporal=0;
      arreglo = new long [10];
      while (i<num) {
         arreglo[i]=Math.round(Math.random()*10000);
         System.out.println(arreglo[i]);
         i++;       }
      System.out.println("Despues del ordenamiento: ");
      for (i=0;i<num;i++) {
         for(int j=i+1;j<num;j++) {
            if (arreglo[j]>arreglo[i]) {
               temporal=arreglo[i];
               arreglo[i]=arreglo[j];
               arreglo[j]=temporal;
            }
         }
      }
      for (i=0;i<num;i++) {
         System.out.println(arreglo[i]);       }
```

Ordenamiento con el método de la burbuja

- **Ventajas:**
- Es bastante sencillo
- En un código reducido se realiza el ordenamiento
- Eficaz

- **Desventajas:**
- Consume bastante tiempo de computadora
- Requiere muchas lecturas/escrituras en memoria

Ordenando un arreglo

- **Método de selección**
- Seleccionar el menor elemento del arreglo y colocarlo en la primera ubicación mediante el intercambio de valores.
- Buscamos entre los restantes el menor y lo colocamos en la segunda ubicación, y así sucesivamente.

Ordenamiento con el método de selección

```
int num=10, i=0, k;
long arreglo [], temporal;
arreglo = new long [10];
while (i<num) {
    arreglo[i]=Math.round(Math.random()*10000);
    System.out.println(arreglo[i]);
    i++;
}
System.out.println("Despues del ordenamiento: ");
    for (i=0;i<num-1;i++) {
        k=i;
        temporal=arreglo[i];
        for(int j=i+1;j<num;j++) {
            if (arreglo[j]>temporal) {
                k=j;
                temporal=arreglo[j];
            }
        }
        arreglo[k]=arreglo[i];
        arreglo[i]=temporal;
    }
for (i=0;i<num;i++) {
        System.out.println(arreglo[i]);
    }
```

Ordenamiento con el método de selección

- **Ventajas:**
- Fácil implementación.
- No requiere memoria adicional.
- Realiza pocos intercambios.
- Rendimiento constante: poca diferencia entre el peor y el mejor caso.

- **Desventajas:**
- Lento.
- Realiza numerosas comparaciones.

Practicando con el Ordenamiento

- Modifique los ejemplo del método de burbuja y del método selección, para que el ordenamiento sea de menor a mayor.

- Investigue el método de ordenación Shell y el método por inserción y elabore un ejemplo del funcionamiento de cada uno.