3. LECTURA DE DATOS DESDE EL TECLADO

Java ofrece una forma fácil de leer datos desde el teclado haciendo uso del objeto ***BufferedReader***. Este objeto cuenta con diferentes métodos que podemos utilizar dependiendo de la operación que se quiera realizar.

3.1 LECTURA DE CARACTERES

Para poder leer un caracter desde el teclado se hace necesario utilizar el método ***read(),*** cuyo objetivo es leer el caracter actual del buffer de entrada y su correspondiente sintaxis es:

```
variable = (char) System.in.read();
```

Para poder tener acceso al método ***read()*** es necesario incluir el paquete ***java.io***, lo cual demanda del programador incluir como primera línea del programa la instrucción `import java.io.*;` Este método hace que el compilador genere una excepción ***IOException*** que se hace necesario manejar adicionando la declaración ***throws*** para evitar la interrupción del programa. El siguiente ejemplo muestra la forma de uso del método ***read()*** así como el uso de ***throws***.

```
import java.io.*;
public class LecturaCaracteres
{
      public static void main(String[]args) throws
      IOException
      {

            Letras lectura1 = new Letras();
            lectura1.leerLetra();
            System.out.println("Caracter leido: "+lectura1.
            letra1);
      }
}
```

```
class Letras
{
      char letra1;
      void leerLetra() throws IOException
      {
                  System.out.println("Digite un caracter?");
                  letra1 = (char)System.in.read();
      }
}
```

3.2 LECTURA DE CADENAS

La mayoría de veces no solo necesitamos leer el carácter actual del buffer de entrada, sino que necesitamos leer toda una cadena de caracteres, tarea que no resultaría nada práctica si pretendiéramos hacerlo a través del método ***read()***. Para esta trabajo java cuenta con el método ***readLine()***, el cual permite crear un objeto tipo **String**(cadena) en el que se almacenan todos los caracteres de la línea leída. Además de ser necesario incluir el paquete java.io, el programador debe establecer el flujo de entrada, lo cual se hace colocando como primera línea del método ***main()*** la instrucción que contenga un objeto de la clase ***BufferedReader*** y la referencia a ***InputStreamReader.*** Para evitar que el programa realice acciones incontroladas es necesario el manejo de la excepción ***IOException*** adicionando el modificador ***throws IOException*** al método ***main().*** La línea leída debe ser almacenada mediante el método ***readLine()*** en una variable de tipo ***String***. El siguiente ejemplo ilustra la forma de hacer la lectura de una cadena de caracteres desde el teclado:

```
import java.io.*;
public class LecturaCadenas
{
      public static void main(String[]args)throws IOException
      {
            BufferedReader BufferedReader1 = new
            BufferedReader(new InputStreamReader(System.in));
             Cadenas Nombre = new Cadenas();
             Nombre.leerCadena(BufferedReader1);
             System.out.println("El nombre leído fue:
             "+Nombre.cadena1);
      }
}
class Cadenas
```

```
{
        String cadena1;
        void leerCadena(BufferedReader lector1)throws
        IOException
        {
                System.out.println("Digite su nombre: ");
                cadena1 = lector1.readLine();
        }
}
```

3.3 LECTURA DE DATOS NUMÉRICOS

Al igual que con los caracteres alfabéticos, los caracteres numéricos que se ingresan por teclado también son leídos como si fueran cadenas de tipo ***String***. Para poderlos almacenar en variables numéricas se hace necesario que una vez leídos se les haga la conversión en el programa.

3.3.1 Lectura de números enteros

Para convertir una cadena de caracteres numéricos en un número entero (***int***) es necesarioo:

- Convertir la cadena en un objeto de la clase envoltorio ***Integer***, a través de la función miembro estática ***valueOf***.
- Convertir el objeto de la clase envoltorio ***Integer*** en una variable de tipo ***int***, a través de la función ***intValue()***

El siguiente ejemplo cuyo objetivo es registrar y posteriormente visualizar el número de registro de un taxi dentro de una empresa de transporte, ilustra la forma de leer una cadena y convertirla en un dato de tipo entero:

```
mport java.io.*;
public class LeerEnterosTeclado
{
        public static void main(String[]args)throws IOException
        {
                BufferedReader BufferedReader1 = new
                BufferedReader(new InputStreamReader (System.in));
                 Taxis taxi1 = new Taxis();
                 taxi1.registrarTaxi(BufferedReader1);
                 System.out.println("Número interno del taxi:
                 "+taxi1.numInterno);
```

```
        }
}

class Taxis
{
        int numInterno;
        void registrarTaxi(BufferedReader Lectura)throws
        IOException
        {
                System.out.println("Registre el número interno
                del taxi: " );
                /* Conversión de una cadena leída desde el
                teclado en un dato de tipo int */
                numInterno = Integer.valueOf(Lectura.readLine()).
                intValue();
        }
}
```

3.3.2 Lectura de números con decimales

Para convertir una cadena de caracteres numéricos en un número con decimales (double) es necesario:

- Convertir la cadena en un objeto de la clase envoltorio ***Double***, a través de la función miembro estática ***valueOf***.
- Convertir el objeto de la clase envoltorio ***Double*** en una variable de tipo ***double***, a través de la función ***doubleValue()***

El siguiente ejemplo cuyo objetivo es registrar y posteriormente visualizar el valor de la cuota correspondiente al impuesto de rodamiento para un bus dentro de una empresa de transporte, ilustra la forma de leer una cadena y convertirla en un dato de tipo double:

```
import java.io.*;
public class LeerDoublesTeclado
{
        public static void main(String[]args)throws IOException
        {
                BufferedReader BufferedReader1 = new BufferedReader
                (new InputStreamReader(System.in));
```

```
            Buses Bus1 = new Buses();
            Bus1.leerImpuestoRodamiento(BufferedReader1);
            System.out.println("Valor rodamiento:
            $"+Bus1.impuestoRodamiento);
      }
}

class Buses
{
      double impuestoRodamiento;
      void leerImpuestoRodamiento(BufferedReader Lectura)
      throws IOException
      {
            System.out.println("Valor impuesto de
            rodamiento:");
            /* Conversión de una cadena leída desde el
            teclado en un dato de tipo doublé */
            impuestoRodamiento =
            Double.valueOf(Lectura.readLine()).doubleValue();
      }
}
```

3.4 CONVERSIÓN DE CADENAS EN BOOLEANOS

También se hace necesario que cadenas leídas desde teclado (Ejemplo: "SI"/"NO") deban ser almacenadas en variables de tipo ***boolean*** con valores true/false. Esta conversión se puede programar mediante un bloque ***if-else*** en la que también se utilizan los métodos ***toUpperCase()*** y ***equals(cadena)***.

El método ***toUpperCase()*** convierte una cadena de caracteres a su equivalente en caracteres alfabéticas en mayúscula.

Con el método ***equals(cadena)*** es posible comparar el contenido de una cadena***(String).***

A continuación se presenta un ejemplo en el que a través de un bloque ***if-else*** se le asignan valores a una variable de tipo ***boolean***, a partir de una cadena de caracteres leída desde el teclado. Además, este ejemplo también permite visualizar el uso de los métodos ***toUpperCase()*** y ***equals(cadena).***

```
import java.io.*;
public class LeeBooleanosTeclado
{
	public static void main(String[]args)throws IOException
	{
		BufferedReader BufferedReader1 = new BufferedReader
		(new InputStreamReader(System.in));
		Casas Casa1 = new Casas();
		Casa1.leerCasas(BufferedReader1);
		if(Casa1.garaje==true)
				System.out.println("La casa tiene
					garaje!");
		else
			System.out.println("La casa no tiene
			garaje!");
	}
}
class Casas
{
	boolean garaje;
	String auxgaraje;
	void leerCasas(BufferedReader Lectura)throws IOException
	{
		System.out.println("Garaje?  SI/NO");
		auxgaraje = Lectura.readLine();
		auxgaraje = auxgaraje.toUpperCase();
		if(auxgaraje.equals("SI"))
			garaje = true;
		else
			garaje = false;
	}
}
```

3.5 EJERCICIO DE APLICACIÓN *

El siguiente ejemplo ilustra de forma íntegra los ítems tratados en los anteriores numerales. El objetivo de este programa es calcular el valor a pagar por concepto de administración por parte de un propietario de apartamento. Además se muestran los resultados obtenidos al ejecutar el programa con datos correspondientes al número del apartamento, clase, garaje y valor total.

```
import java.io.*;
public class AdministracionEdificio
{
	public static void main(String[]args)throws IOException
	{
	BufferedReader BufferedReader1 = new BufferedReader(new
	InputStreamReader(System.in));
	Apartamentos Apto1 = new Apartamentos();
	Mensualidades Mayo = new Mensualidades();
	Apto1.leerDatosApartamento(BufferedReader1);
	System.out.println("CUOTA DE AMINISTRACION");
	System.out.println("APARTAMENTO: "+Apto1.numero +" TIPO:
	"+Apto1.tipo+ " GARAJE: "+Apto1.auxgaraje +
	" VALOR: " + Mayo.calcularadministracion(Apto1));
	}
}
class Apartamentos
{
	double numero;
	int tipo;
	boolean garaje;
	String auxgaraje;
	void leerDatosApartamento(BufferedReader entrada) throws
	NumberFormatException, IOException
	{
		System.out.println("DIGITE NUMERO DEL
		APARTAMENTO: ");
		System.out.flush();
		numero = Double.valueOf(entrada.readLine()).
		doubleValue();
		System.out.println("DIGITE EL TIPO DE
		APARTAMENTO: (1,2,3,4)");
		System.out.flush();
		tipo = Integer.valueOf(entrada.readLine()).
		intValue();
		System.out.println("EL APARTAMENTO TIENE
		GARAJE?  SI/NO ");
		System.out.flush();
		auxgaraje = entrada.readLine();
		auxgaraje = auxgaraje.toUpperCase();  //El método
		toUpperCase Convierte una cadena a MAYUSCULAS
		if (auxgaraje.equals("SI"))  //El método equals()
		pemite evaluar el contenido de una cadena
```

```
                garaje = true;
            else
                garaje = false;
        }

}
class Mensualidades
{
        double calcularadministracion(Apartamentos Apartamento1)
        {
            double valcuota;
            if(Apartamento1.tipo == 1)
                valcuota = 30000;
            else if(Apartamento1.tipo == 2)
                valcuota = 40000;
            else if(Apartamento1.tipo == 3)
                valcuota = 50000;
            else
                valcuota = 70000;
            if(Apartamento1.garaje==false)
                return valcuota;
            else
                return valcuota + 20000;
        }
}
```

```
DIGITE NUMERO DEL APARTAMENTO:
304
DIGITE EL TIPO DE APARTAMENTO: (1,2,3,4)
4
EL APARTAMENTO TIENE GARAJE?  SI/NO
SI
          CUOTA DE ADMINISTRACIÓN
  APARTAMENTO: 304.0 TIPO: 4 GARAJE: SI
              VALOR: 90000.0
```

Datos ingresados y salida después de la ejecución del programa