



3.- Operaciones de lectura, escritura y variables de cadenas de caracteres.

3.1.- Lectura de variables de tipos primitivos.

Para poder colocar un valor tecleado por el usuario de un programa en una celda de memoria de una variable, es necesario leerla del teclado, para ello inovaProg proporciona facilidades que simplifican esta operación. En general la lectura de cualquier tipo de variable primitivo se hace con la siguiente sintaxis:

```
<nombreVariable>=Ip.wlee(<"Texto descriptivo">,  
                           <nombreVariable>);
```

O bien

```
<nombreVariable>=Ip.lee(<"Texto descriptivo">,  
                        <nombreVariable>);
```

La facilidad **Ip.wlee** permite la lectura de una ventana y la facilidad **Ip.lee** lo hace desde la consola.

Los tipos que pueden leerse utilizando estas rutinas son: **int**, **float**, **double**, **char** y **String**.

Note como todos los tipos mencionados se leen de la misma manera y se valida que el valor tecleado por el usuario tenga una configuración válida.

Por ejemplo las instrucciones:

```
int edad=0;  
edad=Ip.wlee("Teclee su edad");
```

Muestra la figura 2.1 invitando al usuario del programa a teclear su edad

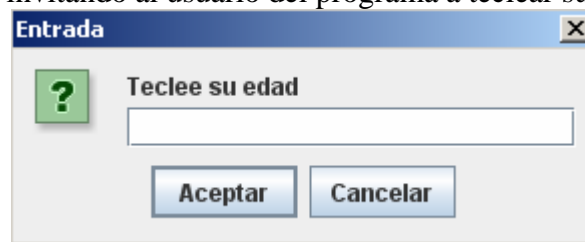


Figura 3.1 Dialogo que muestra inovaProg para leer una variable.

3.2.- Lectura de respuestas a preguntas.



Primer Curso De Programación Utilizando Java



En ocasiones no se leen valores, sino opciones o respuestas a una pregunta de tipo “sí o no”, las respuestas se almacenan en variables de tipo booleanos y el inovaProg facilita leer respuestas de preguntas proporcionando las facilidades `Ip.wLeeRespuesta` e `Ip.wescribePregunta`, ambas se utilizan de la misma manera, leen un sí o un no y lo almacenan en forma booleana en una variable tipo ***boolean***.

La sintaxis es:

```
<nombreVariableTipoboolean>=Ip.wleeRespuesta(  
    <"Texto de la pregunta">,  
    <nombreVariableTipoboolean>);
```

O bien:

```
<nombreVariableTipoboolean>=Ip.wescribePregunta(  
    <"Texto de la pregunta">,  
    <nombreVariableTipoboolean>);
```

El efecto logrado es que el programa presenta una ventana con el texto de la pregunta y dos botones uno etiquetado como sí y otro como no, por ejemplo si tenemos la instrucción :

```
boolean respuesta=false;  
respuesta=Ip.wleeRespuesta("Es la primera vez que  
utiliza este programa?");
```

Obtenemos la ventana mostrada en la figura 3.2.

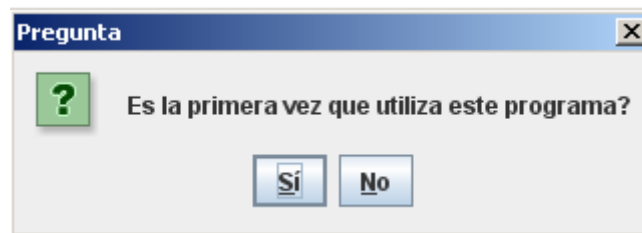


Figura 3.2 Dialogo que muestra inovaProg para leer una respuesta a pregunta

Note que el valor por defecto es la respuesta afirmativa que le da un valor de ***true*** a la variable booleana que la recibe, esta se obtiene simplemente oprimiendo la tecla de entrada.

3.3.- Escritura de variables en ventana y en consola.

Para mostrar el contenido de la celda de memoria de una variable el inovaProg proporciona las facilidades `Ip.wescribe` e `Ip.escribe`, que utilizan la siguiente sintaxis:

Para escribir en una ventana:

```
<nombreVariable>=Ip.wescribe(<"Texto descriptivo">,
```



Primer Curso De Programación Utilizando Java



```
<nombreVariable>);
```

O bien para escribir en la consola:

```
<nombreVariable>=Ip.escribe(<"Texto descriptivo">,  
    <nombreVariable>);
```

Estas instrucciones permiten examinar los valores contenidos en las variables y mostrar resultados.

Las instrucciones:

```
int edad=20;  
Ip.wescribe("El valor de la variable edad es "  
    ,edad);
```

Producen la figura 3.3



Figura 3.3 Dialogo que muestra inovaProg para una instrucción de escritura con Ip.wescribe.

3.4.- Escritura de mensajes en ventana y en consola.

En ocasiones es necesario mostrar mensajes más complejos que el simple valor de una variable, los mensajes pueden contener el valor de más de una variable y puede ser necesario dividirlos en varios renglones e incluso formatear las cantidades numéricas a un número definido de decimales o bien presentarlas como cantidades de dinero.

En esta sección mostraremos las mencionadas operaciones de escritura.

La sintaxis para mostrar un mensaje en una ventana es:

```
Ip.wescribeMensaje(<nombreVariableConMensaje>);
```

O bien

```
Ip.wescribeMensaje(<"Literal o expresión">);
```

Para mostrar el mensaje en la consola:

```
Ip.escribeMensaje(<nombreVariableConMensaje>);
```

O bien



Primer Curso De Programación Utilizando Java



```
Ip.escribeMensaje(<"Literal o expresión">);
```

El primer caso es el de mensajes con valores de mas de una variable, para ello utilizamos una variable de tipo cadena de caracteres que contenga los valores y los textos que deseamos mostrar, por ejemplo suponiendo que tenemos dos variables como se muestran a continuación:

```
int edad=20;
```

```
String nombre="Juan Perez";
```

Y deseamos mostrar en una ventana el mensaje "La edad de Juan Perez es de 20 años"

Definimos la variable

```
String mensaje="";
```

Utilizamos el operador de cadenas de caracteres que permite concatenar o unir dos cadenas para definir la operación:

```
mensaje= "La edad de "+ nombre +  
          " es de " + edad + " años";
```

```
Ip.wescribeMensaje(mensaje);
```

O bien si queremos mostrarlo en la consola:

```
Ip.escribeMensaje(mensaje);
```

Obtenemos la salida mostrada en la figura 3.4

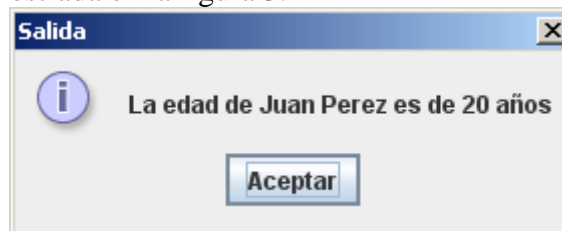


Figura 3.4 Ventana que muestra inovaProg para una instrucción `Ip.wescribeMensaje`.

En el segundo caso podemos requerir un mensajes de varios renglones, para ello podemos utilizar los caracteres de escape "`\n`" para iniciar un renglón o "`\t`" para introducir un tabulador, por ejemplo si deseamos poner en la ventana de la figura 3.4 el encabezado "Datos del interesado", podemos definir el mensaje como sigue para obtener la figura 3.5

```
mensaje= "Datos del interesado \nLa edad de "+  
          nombre +  
          "es de " + edad + " años";
```

Note el uso del signo `+` para unir los textos y las variables y el uso de espacios en blanco para separar los valores de las variables de los textos.



Primer Curso De Programación Utilizando Java

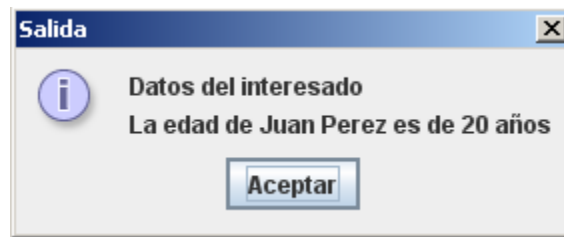


Figura 3.5 Ventana de un mensaje con varios renglones.

En el tercer caso podemos requerir formatear los números especificando la cantidad de decimales. Para ello utilizamos la siguiente sintaxis, en el lugar donde usualmente aparecería el número:

Ip.formatoNumero(<nombreVariableNumerica>);

Por defecto se imprimirán solo dos decimales, o bien especificando cantidad de decimales:

***Ip.formatoNumero(<nombreVariableNumerica>,
<cantidadDecimales>);***

La cantidad de decimales se puede especificar en una variable entera o directamente con una literal (el numero deseado directamente escrito).

Por ejemplo:

```
float peso= (float) 75.456;  
Ip.wescribeMensaje("Peso con dos decimales "+  
    Ip.formatoNumero(peso) +  
    "\nPeso con tres decimales "+  
    Ip.formatoNumero(peso,3));
```

Produce la ventana de la figura 3.6.

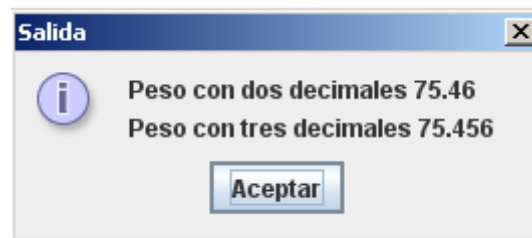


Figura 3.6 Formateo de numeros.

Por ultimo es muy común la necesidad de mostrar un numero con formato de una cantidad de dinero, para ello utilizamos la siguiente sintaxis, en el lugar donde usualmente aparecería el número:

Ip.formatoDinero(<nombreVariableNumerica>);



Primer Curso De Programación Utilizando Java



Por ejemplo:

```
float precio= (float) 000568975.456;  
Ip.wescribeMensaje("Precio "+  
                    Ip.formatoDinero(precio));
```

Produce la ventana de la figura 3.7.



Figura 3.7 Formato de dinero.

3.5.- Resumen de la sintaxis de las facilidades de inovaProg para operaciones de entrada salida.

<code><nombreVariable>=Ip.wlee(<"Texto descriptivo">, <nombreVariable>);</code>	Lee variable de ventana
<code><nombreVariable>=Ip.lee(<"Texto descriptivo">, <nombreVariable>);</code>	Lee variable de consola
<code><nombreVariableTipoboolean>=Ip.wleeRespuesta(<"Texto de la pregunta">, <nombreVariableTipoboolean>);</code>	Lee una respuesta (si/no) a una pregunta
<code><nombreVariableTipoboolean>=Ip.wescribePregunta(<"Texto de la pregunta">,</code>	Lee una respuesta (si/no) a una pregunta. (Forma alterna)



Primer Curso De Programación Utilizando Java



<code><nombreVariableTipobolean>;</code>	
<code><nombreVariable>=Ip.wescribe(<"Texto descriptivo">,<nombreVariable>);</code>	Muestra el valor de una variable con un texto descriptivo en una ventana
<code><nombreVariable>=Ip.escribe(<"Texto descriptivo">,<nombreVariable>);</code>	Muestra el valor de una variable con un texto descriptivo en la consola
<code>Ip.wescribeMensaje(<nombreVariableConMensaje>);</code>	Muestra el mensaje contenido en una variable en una ventana.
<code>Ip.wescribeMensaje(<"Literal o expresión">);</code>	Muestra el mensaje contenido en una expresión o literal tipo cadena de caracteres en una ventana.
<code>Ip.escribeMensaje(<nombreVariableConMensaje>);</code>	Muestra el mensaje contenido en una variable en la consola.
<code>Ip.escribeMensaje(<"Literal o expresión">);</code>	Muestra el mensaje contenido en una expresión o literal tipo cadena de caracteres en la consola.
<code>Ip.formatoNumero(<nombreVariableNumerica>);</code>	Regresa una cadena de caracteres con el valor de la variable numérica con solo dos decimales, sin ceros a la izquierda y agrupado por comas.
<code>Ip.formatoNumero(<nombreVariableNumerica>,<cantidadDecimales>);</code>	Regresa una cadena de caracteres con el valor de la variable numérica con el numero de decimales que se especifica, sin ceros a la izquierda y agrupado por comas.
<code>Ip.formatoDinero(<nombreVariableNumerica>);</code>	Regresa una cadena de caracteres con el valor de la variable numérica con solo dos decimales, sin ceros a la izquierda, con el símbolo de \$ a la izquierda y agrupado por comas.

3.5.- Actividades:

Compile y ejecute los ejemplos adjuntos.

3.7.- Ejercicio Propuesto:



Primer Curso De Programación Utilizando Java



Escriba un programa que lea su nombre su edad, su peso y su altura en variables del tipos adecuados para cada dato y para terminar despliegue sus datos en una sola ventana, poniendo cada dato en una nueva línea.

