



APRENDERPROGRAMAR.COM

PENSAR EN OBJETOS.
CREAR UNA ENTRADA DE
TECLADO O CONSOLA EN
JAVA COMO OBJETO CON
PROPIEDADES Y MÉTODOS.
(CU00661B)

Sección: Cursos

Categoría: Curso “Aprender programación Java desde cero”

Fecha revisión: 2029

Resumen: Entrega nº61 curso Aprender programación Java desde cero.

Autor: Alex Rodríguez

PENSAR EN OBJETOS EN JAVA. UNA ENTRADA DE TECLADO COMO OBJETO.

En un programa anterior pedíamos una entrada por teclado al usuario (ver apartados anteriores). El caso visto corresponde a una ejecución propia de la programación estructurada: almacenamos una entrada de teclado en una variable, y hacemos algo con esa variable (en este caso mostrarla por pantalla). Revisa ese programa para refrescar los conceptos.



Ahora vamos a tratar de pensar como programadores bajo el paradigma de la programación orientada a objetos. En la programación orientada a objetos toda entidad que sea susceptible de tener unas propiedades y unos métodos puede ser definida en una clase. Si pensamos en una entrada de texto por teclado podríamos pensar en:

- a) **Propiedades:** tendrá un contenido (cadena de caracteres). Podríamos añadir otras.
- b) **Métodos:** podemos tener como operaciones definidas sobre una entrada de caracteres el pedir la entrada (pedirEntrada), el obtener el contenido de la entrada (getEntrada), el obtener la primera palabra de la entrada (getPrimeraPalabra), y el obtener la longitud como número de caracteres que tiene la entrada (getLongitud).

Con este razonamiento, cuando queramos pedir una entrada de teclado, crearemos un objeto EntradaDeTeclado sobre el cual podemos ejecutar operaciones diversas. Escribe esta definición de clase:

```
import java.util.Scanner; //Importación del código de la clase Scanner desde la biblioteca Java
// Definimos una clase EntradaDeTeclado aprenderaprogramar.com
public class EntradaDeTeclado {
    private String entradaTeclado; //Variable de instancia (campo) de la clase

    public EntradaDeTeclado () { //Constructor
        entradaTeclado="";
        pedirEntrada(); //Esto es una llamada a un método interno. Al crear un objeto automáticamente ejecutamos el método
    } //Cierre del constructor

    public void pedirEntrada () { //Método de la clase
        Scanner entradaEscaner = new Scanner (System.in);
        entradaTeclado = entradaEscaner.nextLine (); } //Cierre del método pedirEntrada

    public String getEntrada () { return entradaTeclado; } //Cierre del método getEntrada
    public String getPrimeraPalabra () {
        /*IMPORTANTÍSIMO: EN JAVA LOS OBJETOS NO SE PUEDEN COMPARAR USANDO == POR ESO ESTO NO FUNCIONARÁ
        if (entradaTeclado.substring(0,1)=="j") {
            System.out.println ("Hemos detectado una j");
        } else { System.out.println ("Mira esto:" + entradaTeclado.substring(0,1) ); }*/
    }
}
```

```

for (int i=0; i < entradaTeclado.length() - 1; i++) {
    if (entradaTeclado.substring(i, i+1).equals(" ")) { //IMPORTANTÍSIMO: COMPARAMOS CON EQUALS
        return entradaTeclado.substring(0, i+1);    }
    }
return entradaTeclado; //Caso de que no se detecte ningún espacio devolvemos lo que haya
} //Cierre del método getPrimeraPalabra

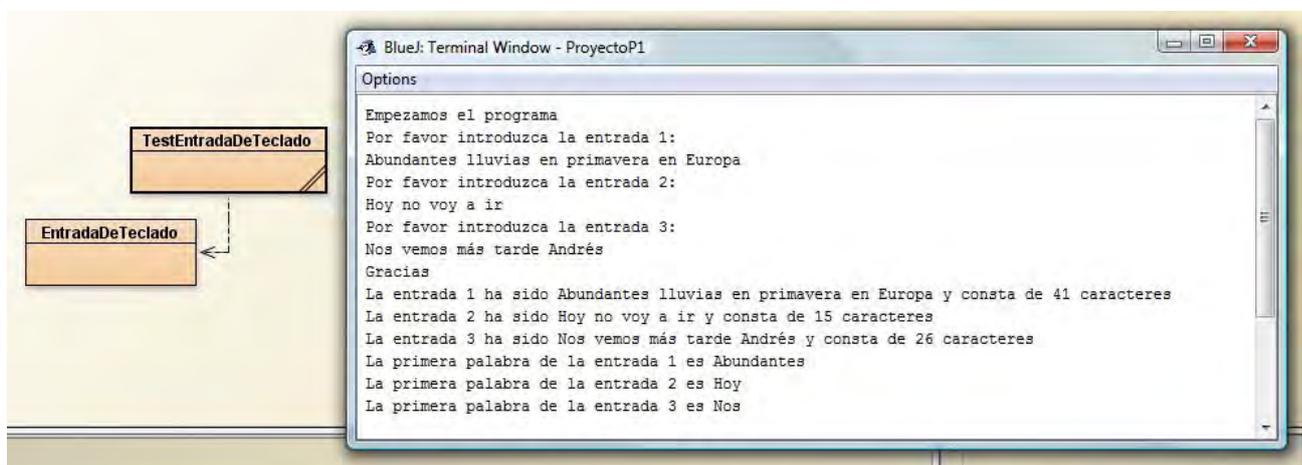
public int getLongitud () { //Método de la clase
    return entradaTeclado.length();
} //Cierre del método getLongitud
} //Cierre de la clase
    
```

A su vez crea una clase para probar la clase EntradaDeTeclado con este código:

```

// Nuestra clase de prueba aprenderaprogramar.com
public class TestEntradaDeTeclado {
    public static void main (String [] args) {
        System.out.println ("Empezamos el programa");
        System.out.println ("Por favor introduzca la entrada 1:");
        EntradaDeTeclado entrada1 = new EntradaDeTeclado();
        System.out.println ("Por favor introduzca la entrada 2:");
        EntradaDeTeclado entrada2 = new EntradaDeTeclado();
        System.out.println ("Por favor introduzca la entrada 3:");
        EntradaDeTeclado entrada3 = new EntradaDeTeclado();
        System.out.println ("Gracias");
        System.out.println ("La entrada 1 ha sido " + entrada1.getEntrada() + " y consta de " + entrada1.getLongitud() + " caracteres");
        System.out.println ("La entrada 2 ha sido " + entrada2.getEntrada() + " y consta de " + entrada2.getLongitud() + " caracteres");
        System.out.println ("La entrada 3 ha sido " + entrada3.getEntrada() + " y consta de " + entrada3.getLongitud() + " caracteres");
        System.out.println ("La primera palabra de la entrada 1 es " + entrada1.getPrimeraPalabra() );
        System.out.println ("La primera palabra de la entrada 2 es " + entrada2.getPrimeraPalabra() );
        System.out.println ("La primera palabra de la entrada 3 es " + entrada3.getPrimeraPalabra() );
    } //Cierre del main
} //Cierre de la clase
    
```

Ejecuta el programa de test y comprueba cuáles son los resultados. En esta imagen vemos un ejemplo.



Hay varias cuestiones a comentar de nuestro código:

- a) Dentro del constructor hemos realizado una invocación a un método interno de la clase. ¿Por qué? Porque hemos decidido que cada vez que se cree un objeto de la clase se ejecute ese método. En ocasiones un constructor no invoca a métodos, pero en otras ocasiones sí. Depende de lo que decidamos como programadores en función de las circunstancias y necesidades.
- b) En el método `getPrimeraPalabra` hemos escrito `if (entradaTeclado.substring (i, i+1).equals(" "))`. Con esta expresión, dentro de un bucle en el cual vamos recorriendo carácter a carácter el String introducido por el usuario, buscamos determinar dónde se encuentra el primer espacio vacío, que consideramos delimita la primera palabra. Fíjate que estamos usando un método para realizar comparaciones denominado `equals` que es aplicable a los objetos en general y a los String en particular. ¿Por qué no usamos una expresión como `if (entradaTeclado.substring (i, i+1) == " ")`? En primer lugar, porque no funciona. El motivo para ello es que los objetos no se pueden comparar como si se tratara de tipos primitivos. En su momento, dijimos que un **objeto es algo distinto a un tipo primitivo**, aunque “porte” la misma información y que una cadena de caracteres es un objeto. Una peculiaridad de los objetos es que no se pueden comparar usando el operador `==`. En el siguiente epígrafe explicaremos por qué.
- c) En el método `main` usamos expresiones del tipo `EntradaDeTeclado entrada2 = new EntradaDeTeclado();`. Esto es la forma abreviada de escribir:

```
EntradaDeTeclado entrada2;
entrada2 = new EntradaDeTeclado();
```

Simplemente estamos fusionando la escritura en una línea. A primera vista, parece una forma de escritura un poco redundante. Se repite por todas partes “EntradaDeTeclado”. No obstante, esto tiene su razón de ser como veremos más adelante cuando hablemos de herencia y polimorfismo.

Ejecuta el programa y realiza pruebas con él. Activa la parte de código que está inactiva como comentario y que trata de detectar una letra `j` usando el operador `==`. Comprueba si funciona o no. Modifica el código para que realice funciones adicionales que se te ocurran. Ten claro que Java no se aprende solo leyendo o estudiando y haciendo ejercicios: has de escribir código y hacer modificaciones y pruebas por tu cuenta para ir familiarizándote con el lenguaje. Debes tratar de diseñar y desarrollar alguna pequeña aplicación para ir aplicando conocimientos adquiridos. Piensa en un pequeño problema y trata de darle solución como programador.

Próxima entrega: CU00662B

Acceso al curso completo en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) --> Cursos, o en la dirección siguiente:

http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=68&Itemid=188