



Java 7 SE Fundamentals





Seccion 4

Trabajando con objetos



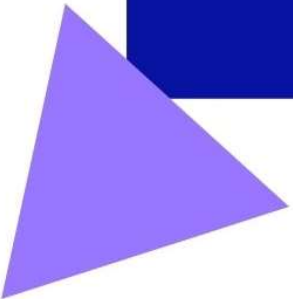
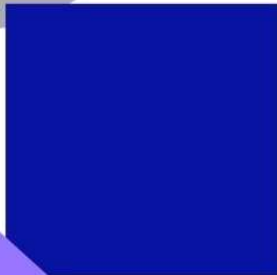
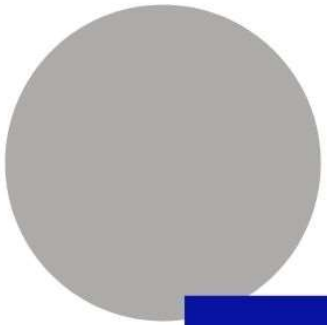
Declarar e inicializar objetos.

Usar referencias de objetos, para manipular datos.

Descripción general de la documentación de Java y cómo usarla.

La clase String y StringBuilder.

Almacenamiento de objetos en la memoria.



Estado de un Objeto

El estado de un objeto (también conocido como sus propiedades o atributos) está representado por campos de datos con sus valores actuales.

Comportamiento de un Objeto

El comportamiento de un objeto (también conocido como sus acciones) se define mediante métodos.

De donde proviene un Objeto

Los objetos del mismo tipo se definen utilizando una clase común. Una clase es una plantilla o modelo o que define cuáles serán los campos de datos y métodos de un objeto.

Relación entre Clase y Objeto

Un objeto es la instancia de una clase. Puede crear muchas instancias de cierta clase. La creación de una instancia se conoce como instanciación.





Ejemplo de creación de objetos

- Las clases son definiciones de objetos y los objetos se crean a partir de clases.

```
public class Pruebas {  
  
    /** Metodo principal */  
    public static void main(String[] args) {  
  
        // Creamos un circulo con radio = 1;  
  
        Circulo circulo1 = new Circulo();  
        System.out.println("El area del circulo 1 de radio "  
            + circulo1.radio + " es " + circulo1.getArea());  
  
        // Creamos un circulo con radio = 1, y despues es cambiado;  
  
        Circulo circulo2 = new Circulo();  
        circulo2.setRadio(25);  
        System.out.println("El area del circulo 2 de radio "  
            + circulo2.radio + " es " + circulo2.getArea());  
  
        // Modificamos el radio de circulo1  
        //calculamos de nuevo el area  
        circulo1.radio = 8;  
        System.out.println("El area del circulo 1 de radio "  
            + circulo1.radio + " es " + circulo1.getArea());  
    }  
}
```

Ejercicios practicos



Cree objetos de las clases Rectángulo, Televisor y la que usted diseño (creadas en la sección 2), además use sus atributos y métodos.



Descripción general de la documentación de Java y cómo usarla

Existen muchas clases predefinidas en Java, que son la base de funcionamiento. Resulta útil tener una documentación en la cual sea posible consultar toda la información relacionada con una clase, para hacer un correcto uso de ella en nuestros programas.

Java™ Platform
Standard Ed. 7

All Classes

Packages

java.applet
java.awt
java.awt.color
java.awt.datatransfer
java.awt.dnd
java.awt.event
java.awt.font
java.awt.geom
java.awt.im
java.awt.im.sni

All Classes

AbstractAction
AbstractAnnotationValueVisitor6
AbstractAnnotationValueVisitor7
AbstractBorder
AbstractButton
AbstractCellEditor
AbstractCollection
AbstractColorChooserPanel
AbstractDocument
AbstractDocument.AttributeContext
AbstractDocument.Content
AbstractDocument.ElementEdit
AbstractElementVisitor6
AbstractElementVisitor7
AbstractExecutorService
AbstractInterruptibleChannel
AbstractLayoutCache
AbstractLayoutCache.NodeDimensions
AbstractList
AbstractListModel
AbstractMap
AbstractMap.SimpleEntry
AbstractMap.SimpleImmutableEntry
AbstractMarshallerImpl

Overview Package Class Use Tree Deprecated Index Help

Prev Next Frames No Frames

Java™ Platform, Standard Edition 7 API Specification

This document is the API specification for the Java™ Platform, Standard Edition.

See: Description

Packages

Package	Description
java.applet	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
java.awt	Contains all of the classes for creating user interfaces and for painting graphics and images.
java.awt.color	Provides classes for color spaces.
java.awt.datatransfer	Provides interfaces and classes for transferring data between and within applications.
java.awt.dnd	Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.
java.awt.event	Provides interfaces and classes for dealing with different types of events fired by AWT components.
java.awt.font	Provides classes and interface relating to fonts.
java.awt.geom	Provides the Java 2D classes for defining and performing operations on objects related to two-dimensional geometry.
java.awt.im	Provides classes and interfaces for the input method framework.
java.awt.im.spi	Provides interfaces that enable the development of input methods that can be used with any Java runtime environment.
java.awt.image	Provides classes for creating and modifying images.
java.awt.image.renderable	Provides classes and interfaces for producing rendering-independent images.
java.awt.print	Provides classes and interfaces for a general printing API.
java.beans	Contains classes related to developing <i>beans</i> — components based on the JavaBeans™ architecture.
java.beans.beancontext	Provides classes and interfaces relating to bean context.
java.io	Provides for system input and output through data streams, serialization and the file system.



La clase String, StringBuilder y StringBuffer

- Las clases **String**, **StringBuilder** y **StringBuffer** se utilizan para procesar cadenas.
- Una cadena es una secuencia de caracteres. En Java una cadena se trata como un objeto.

La clase String



- Un objeto String es inmutable: su contenido no se puede cambiar una vez que se crea la cadena.
- Un objeto String es inmutable; su contenido no se puede cambiar. ¿El siguiente código cambia el contenido de la cadena?
- ```
String s = "Java";
s = "HTML";
```

Spoiler NO!

# Ejercicios practicos



- Cree objetos de la clase String.
- Verificar si dos cadenas son iguales.
- Dadas dos cadenas, y suponiendo que las queremos colocar en orden , verificar cual se coloca primero.
- Contar cuantas letras tiene una cadena.





## La clase **String**, **StringBuilder**

- Las clases **String**, **StringBuilder** se utilizan para procesar cadenas.
- Una cadena es una secuencia de caracteres. En Java una cadena se trata como un objeto.

## La clase **StringBuilder**



- La clase **StringBuilder** es similar a la clase **String** excepto que la clase **String** es inmutable.
- En general, la clase **StringBuilder** se puede usar de la misma manera que la clase **String**, solo que esta clase es mas flexible.
- Tu puedes agregar mas contenido a un objeto de la clase **StringBuilder**, mientras que eso no es posible con la clase **String**.

# Ejercicios practicos

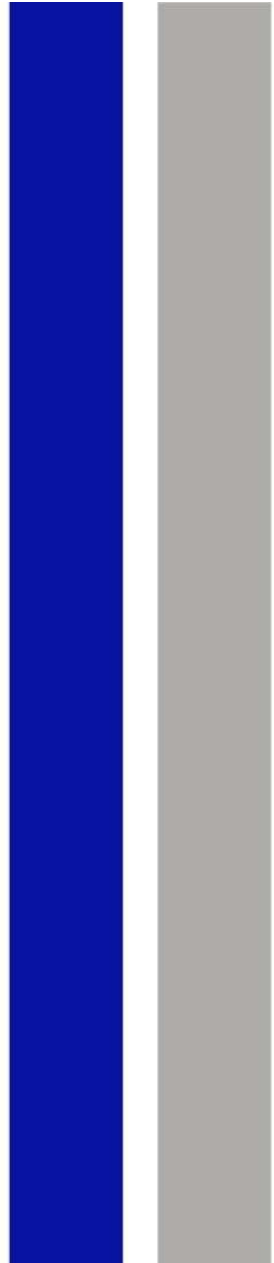
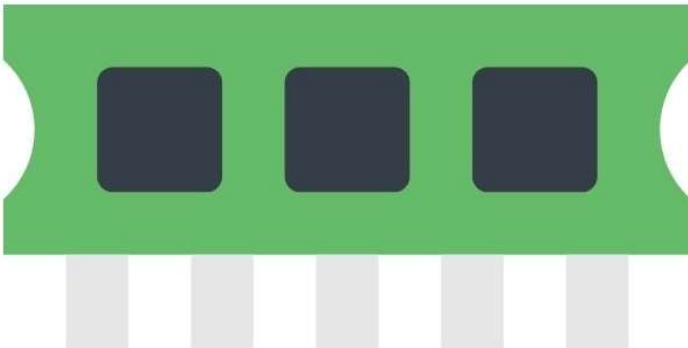


- Cree objetos de las clases String y StringBuilder.
- Contrastar las diferencias que tienen.
- Generar una cadena invertida.



## Almacenamiento de los objetos en la memoria.

- Todos los datos para variables de tipo primitivo, son guardados en un lugar de la memoria de nuestra computadora, llamada Stack.
- Para los tipos que usan referencias (Es decir para los objetos), lo que se guarda también en el Stack, es la referencia (dirección de memoria) donde se ubica dicho objeto.
- Cuando tenemos algo como:  
Circulo **circulo1** = new Circulo();  
Circulo **circulo2** = **circulo1**;  
**circulo2** solo tiene una copia de la referencia de **circulo1**, ambos hacen referencia al mismo objeto.





# Almacenamiento de los objetos en la memoria.

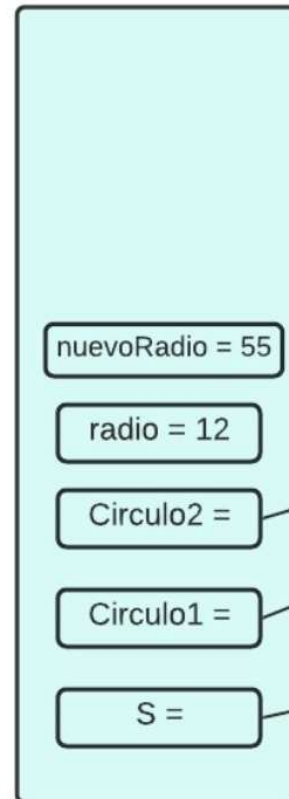
Programa

```
public class Pruebas {
 /** Metodo principal */
 public static void main(String[] args) {
 String s = "Cadena";
 S = "HTML";

 Circulo circulo1 = new Circulo();
 Circulo circulo2 = circulo1;

 int radio = 12;
 int nuevoRadio = radio;
 int nuevoRadio = 55;
 }
}
```

Stack



Heap

