



APRENDERPROGRAMAR.COM

OBJETOS NULL. ERROR O
EXCEPCIÓN JAVA.LANG.
NULLPOINTEREXCEPTION.
AUTOBOXING Y UNBOXING.
OBJETOS ANÓNIMOS.
(CU00668B)

Sección: Cursos

Categoría: Curso “Aprender programación Java desde cero”

Fecha revisión: 2029

Resumen: Entrega nº68 curso Aprender programación Java desde cero.

Autor: Alex Rodríguez

OBJETOS NULL Y JAVA.LANG.NULLPOINTEREXCEPTION EN JAVA

La palabra reservada “null” indica que una variable que referencia a un objeto se encuentra “sin objeto”, es decir, la variable ha sido declarada pero no apunta a ningún objeto. Esto puede deberse a que no se haya inicializado asignándole un objeto mediante la instrucción *new*, o a que hayamos borrado la referencia que contenía la variable.



Si a un objeto no inicializado (referencia *null*) se le trata de aplicar un método o se intenta hacerlo intervenir en un proceso que requiere un objeto inicializado, se obtiene una excepción tipo “NullPointerException”. Vamos a forzar que aparezca una excepción de este tipo. Para ello, crea dos clases y escribe el código que indicamos a continuación en cada una de ellas.

```
// Ejemplo aprenderaprogramar.com
import java.util.ArrayList;
public class ListaNumeros {
    private ArrayList<Integer> listaDeNumeros;

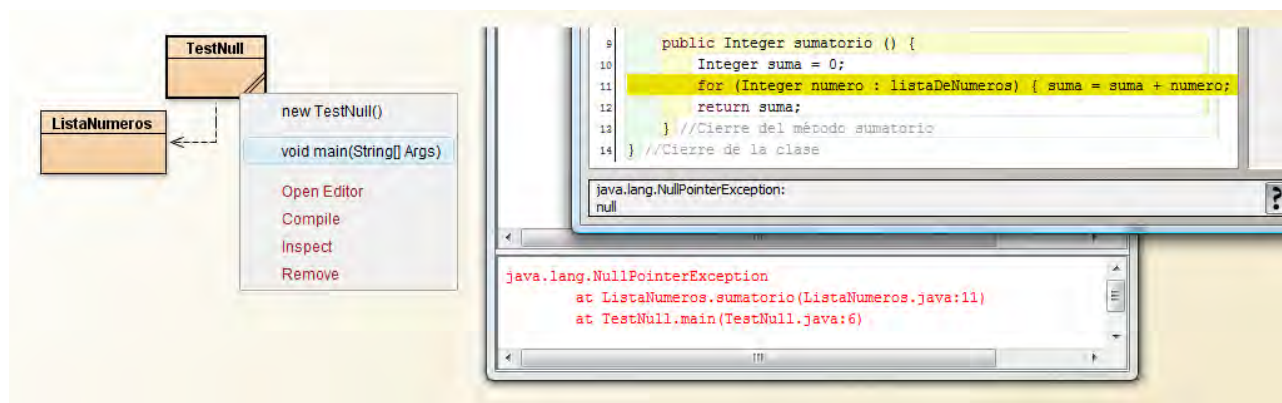
    public ListaNumeros () {} // El constructor está vacío

    public Integer sumatorio () {
        Integer suma = 0;
        for (Integer numero : listaDeNumeros) { suma = suma + numero; }
        return suma;
    } //Cierre del método sumatorio

    public void añadirItem (Integer item) {
        listaDeNumeros.add (item);
    } //Cierre del método
} //Cierre de la clase
```

```
// Ejemplo aprenderaprogramar.com
public class TestNull {
    public static void main (String [ ] Args) {
        ListaNumeros lista1 = new ListaNumeros();
        System.out.println ("El sumatorio actual es: " + lista1.sumatorio() );
    } //Cierre del main
} //Cierre de la clase
```

Ejecuta ahora la clase que contiene el método main. El resultado será algo así:



El programa se empieza a ejecutar y en un momento determinado, se detiene (error en tiempo de ejecución). En la ventana del editor nos aparece *java.lang.NullPointerException* y en la ventana de consola nos aparecen referencias a las líneas de las clases en las que ha saltado el error. ¿A qué se debe este error? Java empieza a ejecutar el programa y reserva memoria para un objeto ListaDeNumeros. Como el constructor de esa clase está vacío, la variable tiene como referencia *null*. Al tratar de hacer un recorrido con un for extendido salta el error que indica que no se puede aplicar ese proceso sobre una variable que referencia a *null*. Modifica el constructor introduciendo este código: `public ListaNumeros () { listaDeNumeros = new ArrayList<Integer> (); }`

Comprueba que ahora el programa se ejecuta sin que salte ningún error. **Los errores de tipo NullPointerException ocurren con frecuencia** al tratar de añadir un ítem a una colección (lista, conjunto, etc.) sin haber inicializado explícitamente esa colección. No basta con que exista una declaración de la variable: hay que inicializarla.

Recordar: una variable (campo) que referencia a un objeto y no ha sido inicializada explícitamente contiene por defecto el valor *null*. En ocasiones podemos comprobar si una variable referencia a un objeto inicializado haciendo comprobaciones de este tipo:

```

if ( ofertaMaxima == null ) { instrucciones A } else { instrucciones B }
    
```

Este tipo de comprobaciones constituyen una medida de seguridad para evitar que se pueda intentar realizar un proceso no permitido como aplicar un método a una variable que contiene null.

AUTOBOXING Y UNBOXING. CONVERSIÓN AUTOMÁTICA DE TIPOS ENVOLTORIO A PRIMITIVOS Y VICEVERSA.

Comentamos en su momento que cada tipo primitivo tenía un tipo equivalente objeto. A los tipos objeto los denominábamos envoltorios. Dado que las colecciones como ArrayList contienen objetos y no tipos primitivos, puede parecer que esta circunstancia complica la programación si queremos por ejemplo trabajar con colecciones de números. Java solventa esta situación realizando una conversión

automática desde el tipo primitivo al envoltorio (autoboxing) o viceversa (unboxing). En el programa anterior son válidas expresiones como `Integer suma = 0;` ó `lista1.añadirItem (7)`. El método `añadirItem` espera un objeto `Integer`. Sin embargo, le pasamos un tipo primitivo y lo acepta. Esto es debido al autoboxing que realiza el compilador Java en segundo plano evitando que salte un error.

El autoboxing y unboxing es una posibilidad destinada a facilitar el trabajo del programador, pero esto no elimina la necesidad de tener muy claro qué es un tipo primitivo y qué es un objeto y cuándo se requiere uno u otro. Comprobarás que no se puede, y no se debe, tratar de usar tipos primitivos y envoltorios como si se tratara de la misma cosa.

OBJETOS ANÓNIMOS

Consideremos el siguiente código:

```
//Ejemplo aprenderaprogramar.com
ListaNumeros lista1 = new ListaNumeros();
Integer num1 = new Integer (6);
lista1.añadirItem (num1);
lista1.añadirItem (7);
```

Cuando añadimos el objeto `num1`, conocemos el nombre del objeto que añadimos. Cuando añadimos mediante autoboxing un `Integer` que contiene el número 7, desconocemos el nombre de ese objeto. **Estos objetos que no tienen nombre específico se denominan “objetos anónimos”**. El uso de objetos anónimos es frecuente en toda situación en la que carece de interés disponer de nombres específicos para los objetos. La forma de crear objetos anónimos admite distintas sintaxis. Estos son algunos ejemplos:

```
//Ejemplos aprenderaprogramar.com
grupoDepositos.add (new Deposito (diametroNuevo, alturaNuevo, idNuevo) );
librería.add (new Libro (“Elogio de la locura”, “Erasmus de Rotterdam”));
club.add (new Persona (“Juan Romero Sánchez”, 32);
```

EJERCICIO

Responde a las siguientes preguntas. ¿Se puede acceder a un objeto anónimo contenido en un `ArrayList`? ¿Se puede inicializar un objeto de una clase que contiene tres atributos de tipo objeto declarando en el constructor que los atributos se inicializan mediante objetos anónimos? Puedes comprobar si tus respuestas son correctas consultando en los foros [aprenderaprogramar.com](http://www.aprenderaprogramar.com).

Próxima entrega: CU00669B

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