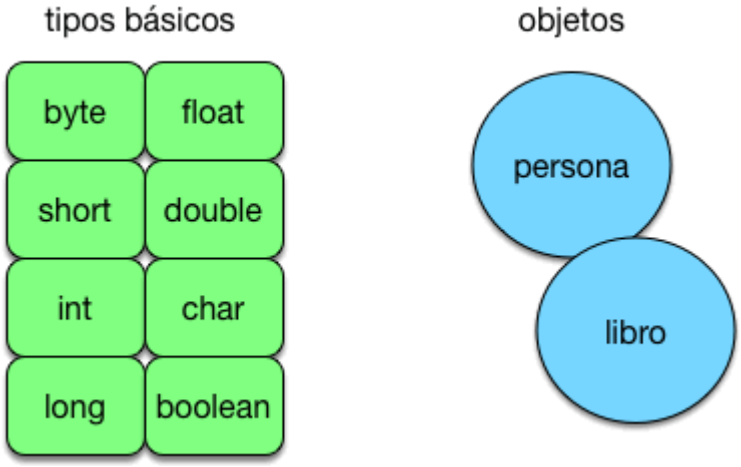
Java Boxing es uno de los conceptos básicos de Java que mas dudas genera a los programadores cuando empiezan a trabajar con el lenguaje. Vamos a explicarlo a detalle. ¿Qué es exactamente el concepto de Java Boxing?. En Java existen dos tipos de estructuras fundamentales , los tipos básicos y los objetos.

CURSO SPRING BOOT
GRATIS
APUNTATE!!



Los tipos básicos únicamente son 8. Los tipos enteros , los flotantes , char y boolean. Todo lo demás son objetos en Java.

Java Boxing y Genéricos

Los problemas comienzan cuando comenzamos a trabajar con tipos genéricos como por ejemplo los ArrayList . Podemos crear un ArrayList de cualquier tipo excepto de los tipos básicos.

```
package com.arquitecturava;
```

```
import java.util.ArrayList;

public class Principal {

    public static void main(String[] args) {

        ArrayList<Integer> lista = new ArrayList<Integer>();
        // ArrayList<int> lista2 = new ArrayList<int>(); // Esto es
incorrecto
        ArrayList<Integer> lista2 = new ArrayList<Integer>(); //
Corrección usando Integer

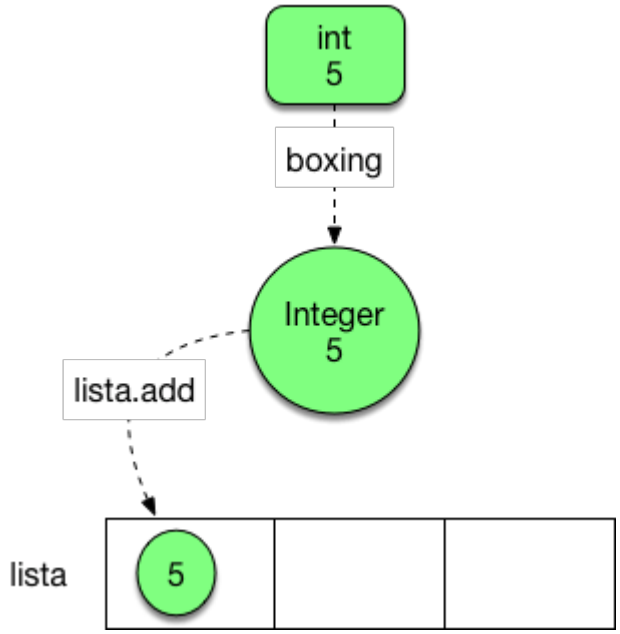
    }

}
```

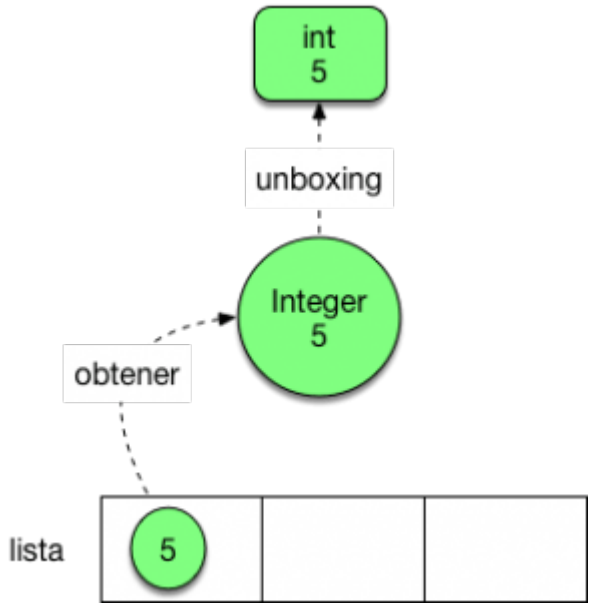
El entorno de desarrollo nos avisará de que no podemos generar un ArrayList de tipo “int” . Sin embargo no hay ningún problema para generar un ArrayList de tipo Integer. Esto es debido a que las clases genéricas no soportan el uso de tipos básicos. Ahora bien si añadimos las siguientes lineas de código :

```
 lista.add(new Integer(5));
 lista.add(5);
```

Ninguna de las lineas nos da error. Para muchas personas es sorprendente ya que estamos añadiendo un tipo básico a la lista y la lista es de objetos Integers. ¿Qué es lo que esta pasando exactamente?. Lo que sucede es que Java se da cuenta de lo que uno quiere hacer y convierte el tipo básico en un objeto Integer antes de ubicarle en la lista. Esto es lo que se denomina habitualmente Boxing (introducir en una caja).



De esta forma podemos añadir tipos básicos a listas genéricas. La misma operación pero de forma inversa (unboxing) es realizada por Java cuando deseamos recorrer la lista y acceder a los números.



A continuación se muestra el código:

```
import java.util.ArrayList;
```

```
public class Principal {

    public static void main(String[] args) {

        ArrayList<Integer> lista = new ArrayList<Integer>();

        lista.add(new Integer(5));
        lista.add(5);
        lista.add(10);
        lista.add(15);
        lista.add(20);

        for (int entero : lista) {
            System.out.println(entero);
        }
    }
}
```

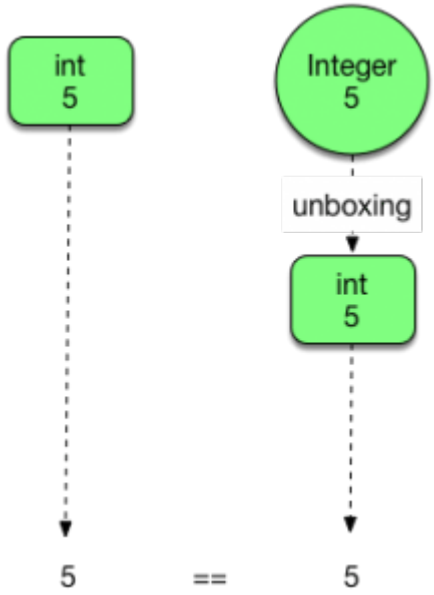
Java Boxing e igualdad

El concepto de unboxing se usa en otras situaciones y genera naturalidad en el código cuando comparamos por igualdad un objeto y un tipo Integer.

```
Integer numero1= new Integer(5);
int numero2=5;
System.out.println(numero1==numero2);
```

Si ejecutamos el resultado será true. Algo que en principio parece muy raro ya que el primero es un tipo básico y el segundo es un objeto. Esto se debe a que Java de forma

automática realiza un unboxing de “numero1”.



Pues estas son las curiosidades que tiene Java Boxing.

CURSO SPRING BOOT

GRATIS

APUNTATE!!

Otros artículos relacionados

- Java this vs this()
- Java Integer Wrapper y certificacion
- Java string literal vs string object
- Java Array for y sus trucos