[**JavaPlayer - La clase Integer Java - Propiedades clase Integer Java - ¿Como sumar dos Integer en Java? - ¿Como restar dos Integer en Java? - ¿Como dividir dos Integer en Java? - ¿Como multiplicar dos Integer en Java?**](http://java-white-box.blogspot.com/2014/01/javaplayer-la-clase-integer-java.html)

Hola como venimos bastante pero bastante básicos hoy vamos hablar de la clase **java.lang.Integer**.

**Clase Integer en java**

La clase **Integer** encierra un valor primitivo del tipo **int** en un objeto. Un único objeto de tipo entero contiene un único campo cuyo tipo es **int**. Ademas esta clase proporciona varios métodos para convertir un **int** en una cadena y una cadena en un **int** (ver casteos) así como otras constantes y métodos útiles para tratar un **int**.

**Declaración de un Integer en Java**

Para declarar un **Integer** en java es muy fácil, es el tipo de dato la variable.

**Ejemplo**

          Integer i;

          int j;

*"Ambos tipos son soportados mas halla que uno sea como objeto y otro sea primitivo".*

**Variables la Clase Integer en Java**

Las variables que contiene la clase **Integer** en Java son:

* **MAX\_VALUE:** Es el máximo valor que puede contener un **int**.(Margen superior)
* **MIN\_VALUE:** Es el mínimo valor que puede contener un**int**. (Margen inferior)
* **SIZE:** Es el numero de bits usados para representar un valor **int**.
* **TYPE:** Retorna el tipo de dato, en este caso como un **Integer** es una presentación del tipo primitivo **int**, retorna **int**.

**Constructores de la clase Integer en Java**

* **Integer(int value)**: Esta constructor crea un nuevo objeto **Integer** con el especifico valor **int** pasado por parámetro.
* **Integer(String s)**:Esta constructor crea un nuevo objeto **Integer** con el especifico valor **String** pasado por parámetro. En el caso que el **String** no se permita pasar a **Integer**, como por ejemplo "R", lanza una excepción del tipo **NumberFormatException**.

**Métodos de la Clase Integer en Java**

Estos son todos los métodos de la clase *Integer en Java*  
  
**static int bitCount(int i)**  
Este método devuelve el número de uno -bits en la representación binaria complemento de dos del valor int especificado.  
**byte byteValue()**   
Este método devuelve el valor de este entero como un byte .  
**int compareTo(Integer anotherInteger)**   
Este método compara dos objetos Integer numéricamente.  
**static Integer decode(String nm)**   
Este método descodifica una cadena en un entero .  
**double doubleValue()**   
Este método devuelve el valor de este entero como un doble .  
**boolean equals(Object obj)**   
Este método compara este objeto para el objeto especificado .  
**float floatValue()**   
Este método devuelve el valor de este entero como un float .  
**static Integer getInteger(String nm)**   
Este método permite determinar el valor entero de la propiedad del sistema con el nombre especificado .  
**static Integer getInteger(String nm, int val)**  
Este método permite determinar el valor entero de la propiedad del sistema con el nombre especificado .  
**static Integer getInteger(String nm, Integer val)**   
Este método devuelve el valor entero de la propiedad del sistema con el nombre especificado.  
**int hashCode()**   
Este método devuelve un código hash de este objeto Integer.  
**static int highestOneBit(int i)**  
Este método devuelve un valor int con a lo sumo un solo de un bit , en la posición de la orden más alto ( " más a la izquierda " ) de un bit en el valor int especificado.  
**int intValue()**   
Este método devuelve el valor de este entero como un int .  
**long longValue()**   
Este método devuelve el valor de este entero como un largo .  
**static int lowestOneBit(int i)**  
Este método devuelve un valor int con un máximo de una sola de un bit , en la posición de la orden más bajo ( " derecha ") de un bit en el valor int especificado.  
**static int numberOfLeadingZeros(int i)**  
Este método devuelve el número de bits cero anteriores a la orden más alta ( " izquierda ") de un bit en la representación binaria complemento de dos del valor int especificado.  
**static int numberOfTrailingZeros(int i)**   
Este método devuelve el número de bits cero después de la orden más bajo ( " derecha ") de un bit en la representación binaria complemento de dos del valor int especificado.  
**static int parseInt(String s)**  
Este método analiza el argumento de cadena como un entero decimal con signo .  
**static int parseInt(String s, int radix)**  
Este método analiza el argumento de cadena como un entero con signo en la base especificada por el segundo argumento .  
**static int reverse(int i)**  
Este método devuelve el valor obtenido invirtiendo el orden de los bits en la representación binaria el complemento a dos del valor int especificado.  
**static int reverseBytes(int i)**  
Este método devuelve el valor que se obtiene invirtiendo el orden de los bytes en representación de complemento a dos del valor int especificado.  
**static int rotateLeft(int i, int distance)**   
Este método devuelve el valor obtenido mediante la rotación de la representación binaria complemento de dos del valor int especificada dada por el número de bits especificado .  
**static int rotateRight(int i, int distance)**  
Este método devuelve el valor obtenido mediante la rotación de la representación binaria complemento de dos del valor int especificada derecha el número de bits especificado .  
**short shortValue()**   
Este método devuelve el valor de este entero como un corto .  
**static int signum(int i)**  
Este método devuelve la función signum del valor int especificado.  
**static String toBinaryString(int i)**  
Este método devuelve una representación de cadena del argumento entero como un entero sin signo en base 2.  
**static String toHexString(int i)**   
Este método devuelve una representación de cadena del argumento entero como un entero sin signo en base 16.  
**static String toOctalString(int i)**   
Este método devuelve una representación de cadena del argumento entero como un entero sin signo en base 8.  
**String toString ()**  
Este método devuelve un objeto String que representa el valor de este Integer.  
**static String toString (int i )**  
Este método devuelve un objeto String que representa el entero especificado .  
**static String toString (int i , int radix)**  
Este método devuelve una representación de cadena del primer argumento en la base especificada por el segundo argumento .  
**static Integer valueOf (int i )**  
Este método devuelve una instancia de tipo Integer que representa el valor int especificado.  
**static Integer valueOf (String s )**  
Este método devuelve un objeto Integer que contiene el valor de la cadena especificada.  
**static Integer valueOf (String s , int radix)**  
Este método devuelve un objeto Integer que contiene el valor extraído de la cadena especificada cuando se analiza con la base dada por el segundo argumento .  
  
  
Viendo toda la especificación técnica de la clase Integer ahora vamos a ver su uso general.

**¿Como sumar dos Integer en Java?**

Todo lo aprendido en este blog es sumamente re usable (en teoría jaja), para *sumar dos Integer en java*utilizamos los [operadores aritméticos (si no te acordas pasate por este link)](http://java-white-box.blogspot.com.ar/2013/11/java-player-operadores-para-que-sirven.html), en nuestro caso el operador es el **+**.

**Ejemplo:**

Integer numeroA = new Integer(10);

Integer numeroB = new Integer(10);

Integer resultado = numeroA + numeroB;

         System.out.println("El resultado es: " + resultado);

**Salida por consola:**

El resultado es: 20

**¿Como restar dos Integer en Java?**

 Para *restar dos Integer en java*utilizamos los [operadores aritméticos (si no te acordas pasate por este link)](http://java-white-box.blogspot.com.ar/2013/11/java-player-operadores-para-que-sirven.html), en nuestro caso el operador es el **-**.  
  
  
**Ejemplo:**

 Integer numeroA = new Integer(10);

 Integer numeroB = new Integer(10);

 Integer resultado = numeroA - numeroB;

         System.out.println("El resultado es: " + resultado);

**Salida por consola:**

El resultado es: 0

**¿Como multiplicar dos Integer en Java?**

 Para *multiplicar dos Integer en java*utilizamos los [operadores aritméticos (si no te acordas pasate por este link)](http://java-white-box.blogspot.com.ar/2013/11/java-player-operadores-para-que-sirven.html), en nuestro caso el operador es el **\***.  
  
  
**Ejemplo:**

 Integer numeroA = new Integer(10);

 Integer numeroB = new Integer(10);

 Integer resultado = numeroA \* numeroB;

         System.out.println("El resultado es: " + resultado);

**Salida por consola:**

El resultado es: 100

**¿Como dividir dos Integer en Java?**

 Para *dividir dos Integer en java*utilizamos los [operadores aritméticos (si no te acordas pasate por este link)](http://java-white-box.blogspot.com.ar/2013/11/java-player-operadores-para-que-sirven.html), en nuestro caso el operador es el **\***.  
  
  
**Ejemplo:**

 Integer numeroA = new Integer(10);

 Integer numeroB = new Integer(10);

 Integer resultado = numeroA / numeroB;

         System.out.println("El resultado es: " + resultado);