

## Cambio de Base

Se realizará una clase llamada CambioBase que permitirá hacer operaciones de una base a otra

Existe un constructor sin parámetros que establece que los números se introducen en base10 y se muestra el acumulado en base10 Así que haría el efecto de una calculadora normal para sumas y restas.

Existen dos métodos llamados: agregarCantidad() y quitarCantidad()

Las versiones de esos dos métodos que reciben números enteros. Ejemplo:  
agregarCantidad(5);

Estaríamos sumando al acumulado el número 5 en base10 ( lo mismo que una calculadora normal )

Para obtener el acumulado de estas operaciones lo hacemos mediante:

getAcumulado(); //este método devuelve una **String** que contendrá la cantidad acumulada

Si cualquiera de los métodos agregarCantidad, quitarCantidad reciben una cantidad negativa deben lanzar una excepción que diga: “no se admiten números negativos”

Ej.

agregarCantidad(-4);

***Nota: Si se toman números en base distinta de 10 se recomienda convertir internamente los números a base10 y hacer luego las sumas y restas en base10. Finalmente si el usuario quiere ver el resultado en otra base se hará la transformación del resultado***

Así pues si se quiere introducir un número en una base diferente de base10 se debe introducir como una String. Ejemplo:

agregarCantidad(“1000”,2);

Con lo anterior estaríamos sumando al acumulado el número 1000 en base2 que es el número 8 en decimal

Hay 3 constructores el constructor sin parámetros que se nombró antes y adicionalmente:

//este constructor establece que por defecto las String que le pasemos estarán en: baseEntrada  
CambioBase( int baseEntrada)

//este otro constructor establece por defecto baseEntrada y la salida de getAcumulado a baseSalida  
CambioBase(int baseEntrada, int baseSalida)

**2 de los constructores llamarán al tercero mediante: this**

Hay tres versiones del método agregarCantidad()

La primera se detalló antes. Es la que recibe un número entero: agregarCantidad(5) recordar que en ese caso se está agregando un número en base10

La segunda versión recibe una String: agregarCantidad("100")

En esta versión el programa toma el valor de baseEntrada que hemos establecido mediante el constructor. Si el constructor fue: CambioBase(3,7) entonces le estamos pasando el número 100 en base3 o lo que es lo mismo: 9 en base10

La tercera versión recibe una String y un int: agregarCantidad("100",2)

En este caso le estamos dando el número y también la base: 100 en base2 o lo que es lo mismo: 4 en base10

Existen las 3 versiones equivalentes para: quitarCantidad()

Hay dos versiones del método: getAcumulado()

La versión que no recibe parámetros devuelve una String con la cantidad acumulada en la base que hayamos establecido en el constructor. Por ejemplo: CambioBase(3,7) nos devolvería una String con la cantidad acumulada en base7

La versión que recibe un int: getAcumulado(8) devolverá la cantidad acumulada en la base que le pasamos como parámetro. En este caso base8

Puntuaciones:

2.5: entregar

5: Se crean los 3 constructores y Se realiza correctamente el operativo para base10 ( las versiones de los métodos cuando reciben números enteros: agregarCantidad(5) , quitarCantidad(7) y mostrar el acumulado como una String en base10 si ejecutamos: getAcumulado(10) )

7.5: Adicionalmente a lo anterior dos de los constructores llaman al tercero mediante this Se lanzan excepciones si se pasan cantidades negativas en agregarCantidad() , quitarCantidad()

10: Se consigue las conversiones de base del enunciado y la clase funciona correctamente para todos los casos descritos ( se realizan todos los métodos solicitados, etc )