

Ejercicios de Java

Tipos primitivos, referencias, paso y devolución de referencias:

1. Crear la clase Point(X,Y) con los métodos get y set de las coordenadas.
2. Crear la clase Line(Point1, Point2) con los métodos:
 - a. get y set de los puntos.
 - b. deltaX y deltaY: valor absoluto de la diferencia de las coordenadas de los puntos.

3. Para las referencias:

```
A = "string";  
B = "string";  
C = A;
```

cual es el resultado de:

```
A == B  
A.equals(B)  
A == C  
A.equals(C)  
B == C  
B.equals(C)
```

4. En el main(), para las referencias:

```
A = new Point(3,5);  
B = new Point(3,5);  
C = A;
```

cual es el resultado de:

```
A == B  
A.equals(B)  
A == C  
A.equals(C)  
B == C  
B.equals(C)
```

5. En el main(), para el código:

```
L = new Line(new Point(3,5), new Point(3,5));  
P = L.getP1();  
Ant = L.deltaX();  
P.setX(10);  
Pos = L.deltaX();
```

¿Es Ant igual Posterior?

6. En la clase principal (main()), crear el siguiente método:

```
void add(Point p) { p.setX(p.getX()+1); }
```

probar el siguiente código:

```
L = new Line(new Point(3,5), new Point(3,5));  
P = L.getP1();  
Ant = L.deltaX();  
Add(P);  
Pos = L.deltaX();
```

¿Es Ant igual Posterior?

7. Sobrescribir los métodos equals y clone de las clases Point y Line. Resolver y volver a probar los tres casos anteriores.

Aplicaciones de Consola:

Juego de “Adivina el número que estoy pensando”, un número del 0 al 100, ya te diré si es mayor o menor que el mío, pero tienes 10 intentos como mucho. `Math.random() * 100`.

Decodificar las cadenas con el siguiente formato:

`3+4+3,4-7*1=`

en los siguientes componentes:

`3 +`

`4 +`

`3,4 -`

`7 *`

`1 =`

mostrando el resultado en la consola.

Crear la clase **Calculadora** que acumule y permita obtener los resultados parciales de las operaciones obtenidas en la decodificación anterior.

Crear la clase de utilidad **Valid** que permita la validación de cadenas. Todos los métodos deben ser de clase, recibir como primer parámetro la cadena a validar y devolver **true** es válido. Debe contar al menos con los siguientes métodos:

`public static boolean isEmpty(String value);`

`public static boolean isNumeric(String value);`

`public static boolean isLenMax(String value, int len); //Inferior a la longitud máxima.`

`public static boolean isLenMin(String value, int len); //Superior a la longitud mínima.`

`public static boolean isPositive(String value);`

`public static boolean isPositive(double value);`

adicionales:

`public static boolean isEMail(String value);`

`public static boolean isURL(String value);`

Ficheros:

Leer un fichero de entrada y generar un fichero de salida, realizando los cálculos necesarios, con los siguientes formatos:

```
FileIN: Entrada.txt
      3+4+3,4-7*1=
FileOUT: Salida.txt
      3
      +4
      +3,4
      -7
      *1
      -----
      3,400000
```

Ampliaciones:

- Fichero de entrada con múltiples líneas.
- Fichero de salida comprimido
- Descompresor del fichero de salida: zip → txt
- Compresor del fichero de entrada: txt → zip
- Procesar entrada/salida comprimida

Interfaz de usuario (swing)

Crear una calculadora similar a la calculadora de Windows XP. Las funciones de memoria son opcionales.



Crear un editor MDI, múltiples ventanas anidadas en la principal (JInternalFrame). La aplicación debería contar con un menú, una barra de botones, ...

