

Problema 3:

Datos personales	
Apellidos:	
Nombre:	

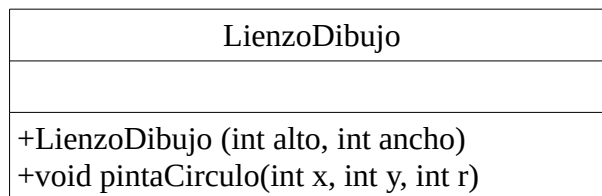
1 Conversiones de tipos

Objetivos

Practicar con las conversiones de tipos

Descripción

Disponemos de esta clase Java que sirve para hacer dibujos en una ventana y que describimos con su diagrama de clase:



Al constructor se le pasa el tamaño de la ventana en pixels. El método pintaCirculo() pinta un círculo de centro (x,y) y radio r, expresados en píxeles.

Escribir un método en una clase aparte, con la cabecera mostrada más abajo, que llamando a pintaCirculo() dibuje en el objeto dib dos círculos concéntricos uno de radio 10 y otro de radio 20 píxeles, cuyo centro está situado en el punto definido en coordenadas polares por las variables lon (en píxeles) y alpha (que está en grados). Las coordenadas del centro de los círculos serán (lon*cos(alpha), lon*sen(alpha)).

```
public void pintaCirculosConcentricos(LienzoDibujo dib, double lon, double alpha)
```

Observar que serán necesarias conversiones de tipo de double a int y que las funciones trigonométricas, en Java, trabajan en radianes. Para que en las conversiones de tipo haya la menor pérdida de precisión deberán hacerse lo más tarde posible.

Respuesta que se pide

<Poner aquí el código Java del método>

2 Strings

Objetivos

Practicar con métodos que manipulan Strings.

Descripción

Escribir un método al que se le pasa un string conteniendo dos o más palabras separadas entre sí por uno o más espacios en blanco, y que retorna la segunda de estas palabras. Puede haber espacios en blanco antes de la primera palabra y después de la última. No hace falta considerar el caso en que el string tenga menos de dos palabras.

Se cuenta para ello con los siguientes métodos de la clase String:

- `int indexOf(char c)`: busca la primera ocurrencia del carácter `c` en el String al que se aplica el método. Si lo encuentra, retorna la posición que ocupa en el String, teniendo en cuenta que los caracteres del String se numeran de 0 en adelante, de izquierda a derecha. Si no lo encuentra retorna -1.
- `String trim()`: retorna un String igual a aquel al que se aplica este método pero sin espacios en blanco ni al principio ni al final.
- `String substring(int ini, int fin)`: Retorna un fragmento del String al que se aplica este método conteniendo los caracteres que ocupan las posiciones desde `ini` (incluida) a `fin` (excluida).

Ver una descripción más completa y ejemplos de uso de estos métodos en la documentación de Java 8, bajo la clase String.

Respuesta:

<Poner aquí el código Java del método>

3 Atributos y métodos estáticos

Objetivo

Practicar con atributos y métodos estáticos

Descripción

Se dispone de una clase con atributos y métodos estáticos (de clase) y normales (de objeto) que obedece al siguiente diagrama:

Circulo
-static int numCirculosCreados -int xCentro, yCentro, radio +static final int grosorLapiz=2

Introducción al Software, Curso 2017-2018

+final int color
+Circulo (int x, int y, int radio, int color) +static int dameCirculosCreados() +pintaCirculo() -creaLienzo()

Escribir en otra clase aparte un main que a modo de ejemplo utilice todos los atributos y métodos públicos de la clase Circulo.

Respuesta

<Poner aquí el código Java de la clase con el main>

4 Métodos que calculan expresiones no aritméticas

Objetivo

Practicar métodos y expresiones no aritméticas

Descripción

Se desea saber:

- si una palabra comienza por un carácter que es un dígito numérico, de '0' a '9'
- si una palabra acaba con una minúscula

Notas:

- La clase String dispone entre otros de los métodos charAt() y length() para, respectivamente, obtener una letra individual o averiguar la longitud (número de letras) del string.

```
char charAt(int i)
int length()
```

- El método charAt() retorna la letra número i del string, teniendo en cuenta que se numeran de izquierda a derecha empezando desde cero.
- El método length() retorna el número de letras del string.

Se recomienda buscar en Internet ejemplos del uso de estos métodos.

- Para saber si una letra es minúscula lo haremos comprobando si está entre la a y la z (minúsculas)¹, usando el operador de comparación <. Para saber si es un dígito comprobaremos igualmente que está entre '0' y '9'.

Se pide:

- Un método al que se le pasa la palabra como parámetro de tipo String y que retorna un booleano que indica si la palabra comienza por un dígito numérico.

¹ Esto deja fuera las letras acentuadas y la ñ, pero por simplicidad ignoraremos estas letras

Introducción al Software, Curso 2017-2018

- Un método al que se le pasa la palabra como parámetro de tipo String y que retorna un booleano que indica si el último carácter es una letra minúscula o no.

Respuesta:

<Poner aquí el código Java de los dos métodos>