

Examen 3ª Evaluación POO DAM

Instrucciones:

1. Clonar el repositorio del examen
2. Crear una rama con nombre: Nombre_apellidos_terceraEvaluacionDAM
3. moverse a la nueva rama creada
4. empezar a realizar los ejercicios
5. al terminar hacer commit y push de la rama creada
6. Verificar en la página git que se ha subido la rama

Realizar los ejercicios de las dos partes, creando las clases y carpetas necesarias, se podrán crear las clases/métodos de apoyo que se consideren necesarias.

Parte 1

- Crear el paquete `es.core.networks.dam.poo.examenTerceraEvaluacion.parte`, dentro crear una clase `Partel`, que contendrá los siguientes métodos estáticos:
 1. `cadenas()`: crear en este método 3 cadenas de caracteres como arrays de `char` para meter vuestro nombre, primer apellido y segundo apellido mostrar la longitud del apellido primero
 2. `dam()`: Crear en este método un `String` con la palabra "DAM" de dos formas distintas y mostrarlos por pantalla
 3. `literal()`: obtener en este método la longitud del literal "curso de DAM";
 4. `arrayString()`: Crear un array de `Strings` con las palabras de la frase: "Esta es la clase de POO". concatenar los elementos primero y tercero del array y mostrar el resultado. iterar por el array para encontrar la palabra menor y la mayor
 5. `buscaEspacios(String frase)`: que reciba un `string` y detecte si existen espacios en él si es así deberá devolver el mensaje sin espacios junto con un mensaje indicando cuántos espacios contiene.
 6. `usoStringBuffer()`: Este método pide por consola el nombre, el primer apellido y el segundo guardando cada uno en un `string`, después creará un `stringbuffer` con la composición de todos en el orden nombre, apellido 1, apellido 2, después eliminará el nombre del `stringbuffer` y mostrará el resultado

Parte 2:

Crear el paquete `es.core.networks.dam.poo.examenTerceraEvaluacion.partell`, dentro crear una clase `Partell`, que contendrá los siguientes métodos estáticos:

1. Dada la matriz:

1 2 3

4 5 6

7 8 9

pintar la matriz por consola tal cual se puede ver aquí.

pintar la matriz por consola invirtiendo las columnas por las filas (quedando así):

1 4 7

2 5 8

3 6 9

2. Crear un método que sume dos matrices de cualquier dimensión y las muestre por pantalla, usar como parámetros las matrices:

2 4 8	1 3 5
-------	-------

1 3 7	2 4 6
-------	-------

3 5 9	3 5 7
-------	-------

3. Crear el método getNumeroExtensiones() que utilizando el string de ficheros

static String

`datos="hola.rm;usuarios.rm;test.txt;texto.rm;claves.rm;cosas.java;gatitos.rm;perritos.java;saludos.txt;adios.rm;fuego.txt;taza.java;java.txt;cuanto.txt;trece.java";`

separe el string en un array con los distintos ficheros que contiene. Para cada uno de ellos separar nombre y extensión y guardar la extensión en una tabla hash como key, el valor será un contador con valor inicial 1, si ya existiese la clave se incrementará el valor del contador para contar el número de ficheros con esa extensión. Una vez procesado todo el string de ficheros mostrar las extensiones que hay y la cantidad de ficheros que hay de cada una.

4. Crear la interfaz estadísticas que contendrá los métodos double getArea() y double getPerimetro. Crear la clase abstracta FiguraGeometrica que implementa la interfaz Estadísticas, con el método String getname() que devuelve una frase descriptiva de la clase.

Crear las clases círculo y cuadrado que heredan de FiguraGeometrica sobrecribir los métodos del padre y de la interfaz con las fórmulas matemáticas apropiadas.

crear 3 objetos tipo circulo y 3 de tipo cuadrado

crear un array de tipo circulo e insertar los circulos

crear un array de tipo cuadrado e insertar los cuadrado

crear un array de tipo FiguraGeometrica e insertar todos los objetos

crear un array de tipo Estadisticas e insertar todos los objetos

recorrer todos los arrays llamando al método calcularArea y calcularPerimetro .

