

# Examen recuperacion segundo parcial DAM

## Parte1

En el paquete `es.corenetworks.dam.segundaEvaluacion.recuperacion`, crear el sub paquete `es.corenetworks.dam.segundaEvaluacion.recuperacion.utilidades`, dentro de este paquete crear la clase `Constantes` con las siguientes constantes:

tipo entero:

- `NUM_REPETICIONES=10`
- `NUM_INTENTOS=4`
- `TAMANYO=5`

tipo double:

- `PI=3.1416`

array de enteros:

- `NUMBERS = {1, 2, 3, 6, 9, 10, 11, 16, 25, 28, 33, 75, 108};`

String:

- `SALUDO="bienvenido"`
- `DESPEDIDA="adios"`
- `INTRO = "Introduce tu nombre";`
- `CODE_200 = "Todo ok";`
- `CODE_301 = "Movimiento permanente de la pagina";`
- `CODE_302 = "Movimiento temporal de la página";`
- `CODE_400 = "Bad request";`
- `CODE_401 = "forbidden";`
- `CODE_404 = "No encontrado";`
- `CODE_500 = "Error interno del servidor";`
- `CODE_503 = "Servicio no disponible";`
- `ERROR_DEFAULT = "Introduce tu nombre";`

Crear el sub paquete `es.corenetworks.dam.segundaEvaluacion.recuperacion.excepciones`, dentro de este paquete crear las siguientes excepciones:

- `RadioNovalidoException`
- `MayorQueException`
- `MenorQueException`

Crear en el mismo paquete la clase `Utilidades` con los siguientes métodos públicos y estáticos:

- `obtenerNumeroAleatorioEntreCeroY100`
- `calculaArea` que recibe como parámetro double el radio y calcule el área de un círculo ( $\pi * R * R$ ). Usa la constante `PI`. Si el radio es 0 o mayor que 100 se debe lanzar la excepción `RadioNovalidoException`, no olvidar la cláusula `throws` en la cabecera del método.

- `calculaPerimetro` que recibe como parámetro `double` el radio y calcule el perímetro de un círculo ( $2 \cdot \pi \cdot R$ ). Usa la constante `PI`. Si el radio es 0 o mayor que 100 se debe lanzar la excepción `RadioNovalidoException`, no olvidar la cláusula `throws` en la cabecera del método.

Crear el subpaquete `es.corenetworks.dam.segundaEvaluacion.recuperacion.parte1`, dentro de este crear la clase `Parte1` de la siguiente forma:

Crear el método público `saludarUsuario` que no devuelve nada y no recibe parámetros. Este método utiliza la constante `INTRO` para mostrar ese mensaje al usuario, después usa la clase `Scanner` para obtener el input del usuario y utiliza la constante `SALUDO` y el input para saludar al usuario.

Crear el método `lanzarExcepcion(int numero)` que recibe un número como parámetro, el método deberá comprobar si el número es mayor o menor que 50 y lanzar la excepción `MayorQueException` con un mensaje descriptivo o `MenorQueException` con otro mensaje descriptivo.

Crear el método `capturarExcepcion` sin parámetros y que no devuelve nada. Este método llama al método `lanzarExcepcion`, deberá usar `try-catch` para controlar la excepción que lanza y mostrar el mensaje que contenga la excepción.

switch

Crear el método `manejadorErrores` que recibe un número entero de error y que usando un `switch` devuelve el string de la constante que pertenece al error, los errores que debe manejar son: 200, 301, 302, 400, 401, 404 y 500, si el entero recibido no es ninguno de estos se devolverá la constante `error_default`. como ejemplo si se recibe el código 200 se debe devolver la constante `CODE_200`, si se recibe el 404 se devolverá la constante `CODE_404`

## parte 2

1- crear el método público y estático `recorreArray` para recorrer usando `while` el array de las constantes y obtener el número mayor del array, imprimirlo por consola, si este número es par mostrar el mensaje "el numero es par"

2- crear el método público y estático `recorreArrayInverso` que utiliza `do-while` para recorrer el array en sentido inverso, calcular la suma de los elementos que se encuentren en una posición impar del array y mostrarla por pantalla.

3- crear el método público y estático `pares1a100` que utilizando `do while`, imprimir por pantalla los numero pares del 1 al 100

4- crear el método público y estático `deTresEnTres` que imprime por pantalla los números del 1 al 100 de 3 en 3 es decir, imprimiría 1, 4, 7, 10...

## Parte 3

Completar el código para que el programa funcione correctamente, de tal forma que se ejecute en bucle hasta que el usuario elija la opción de salir del programa

Además se debe completar el método `procesarEntrada` usando un `switch` para que calcule a petición del usuario el área el perímetro o salga de la aplicación, para ello debe utilizar los métodos creados en la clase `Utilidades` `calcularArea` y `calcularPerimetro`, dando como opción por defecto el mensaje "opcion no valida".