

Problema 6: ArrayList y excepciones

Datos personales	
Apellidos:	Prieels
Nombre:	Cedric

1 *Búsqueda de múltiples elementos*

Descripción

Escribir un método al que se le pasa como parámetro un array de Strings y que retorne un ArrayList de Strings, conteniendo aquellos Strings del array original que comiencen por una letra mayúscula.

Observar que la búsqueda de múltiples elementos no tiene que ver con el algoritmo de búsqueda en tablas visto en clase.

Nota: puede utilizarse el método estático `isUpperCase()` de la clase `Character` para determinar si un carácter es una letra mayúscula.

Respuesta que se pide

```
public ArrayList<String> devuelvePalabras(String[] arrayStrings) {  
  
    ArrayList<String> miArrayList = new ArrayList<String>();  
  
    char primero = "";  
  
    for (int i=0; i<arrayStrings.length; i++) {  
  
        primero = arrayStrings[i].charAt(0);  
        if (isUpperCase(primerero)) {  
  
            miArrayList.add(arrayStrings[i]);  
  
        }  
  
    }  
  
    return miArrayList;  
  
}
```

2 *Excepción leve*

Se dispone de una clase que permite conocer los consumos eléctricos de los electrodomésticos de un hogar a lo largo del año. Obedece al diagrama de clases que se muestra

Hogar
...
+Hogar(String nombreFichero) +void listadoElectrodomestico (String nombreElectrodomestico) throws NoEncontrado +double consumoAnualHogar()

Puede observarse que el método ListadoElectrodomestico() tiene una cláusula throws que anuncia que el método puede lanzar la excepción NoEncontrado, definida en una clase aparte. Esta cláusula nos obliga a tratar esta excepción si invocamos al método.

Se pide hacer un programa principal en una clase aparte que haga lo siguiente:

- Crea un objeto de la clase Hogar a partir del fichero de nombre "electrodomesticos.txt"
- Invoca a listadoElectrodomestico() con el nombre "Lavadora"
- Invoca a listadoElectrodomestico() con el nombre "Luces"

Si en alguno de los pasos b) o c) se lanza NoEncontrado se muestra un mensaje de error y luego se sigue por el paso e), es decir, no se hace el d)

- Muestra en pantalla el consumo anual del hogar obtenido mediante consumoAnualHogar()
- Para probar si la excepción NoEncontrado se lanza correctamente intenta mostrar en pantalla (invocando a listadoElectrodomestico()) el listado del electrodoméstico inexistente "xxxx". En el manejador de la excepción se mostrará un mensaje de error.

Respuesta que se pide

```
public static void main() {  
  
    Hogar miHogar = new Hogar("electrodomesticos.txt");  
    try {  
        miHogar.listadoElectrodomesticos("Lavadora");  
        miHogar.listadoElectrodomesticos("Luces");  
        double consumo = miHogar.consumoAnualHogar();  
        System.out.println("Consumo total "+consumo);  
    } catch (NoEncontrado e) {  
        System.out.println("Este objeto no existe.");  
    } finally {  

```

```
        try {
            miHogar.listadoElectrodomesticos("xxxx");
        } catch (NoEncontrado e) {
            System.out.println("Error, este objeto no existe");
        }
    }
}
```

3 ***Excepciones leve y grave en el mismo método***

Descripción

Se dispone de un método declarado según la cabecera de abajo, perteneciente a la clase Experimento, que puede lanzar dos excepciones declaradas como clases independientes.

```
public static double medida() throws MedidaInestable, MedidaErronea
```

Escribir otro método que, desde otra clase diferente, trate de retornar una medida de un experimento llamando al método ya existente y tratando las excepciones del siguiente modo:

- si se lanza MedidaInestable pone un mensaje de error y retorna 0.0
- si se lanza MedidaErronea pone un mensaje de error y retorna Double.NaN

Respuesta que se pide

```
public double devuelveMedida() {
    try {
        return Experimento.medida();
    } catch (MedidaInestable e) {
        System.out.println("Medida inestable");
        return 0.0;
    } catch (MedidaErronea e) {
        System.out.println("Medida erronea");
        return Double.NaN;
    }
}
```