

Programación I - Examen final

Enero 2014

Cinco municipios consumen agua de un mismo depósito, y se desea gestionar su uso, para ello se hará uso de la siguiente clase.

Clase Municipio

Sus atributos son:

- Nombre del municipio : Cadena de caracteres
- Código postal : Entero positivo
- Número de habitantes : Entero positivo
- Número de hogares : Entero positivo
- Consumo total anual (expresado en metros cúbicos – m₃) : Real positivo

Los métodos son:

- Constructor por defecto y constructor que inicializa los atributos con valores .
- Métodos para consultar y modificar todos los atributos de la clase (métodos **get** y **set**).
- Métodos para **leer** de teclado todos los atributos de la clase, salvo el consumo anual que se inicializa a cero.
- Método para **mostrar** por pantalla los atributos de la clase.

➤ **Métodos a implementar por los alumnos en este examen:**

- Método **consumoMedioHab**, que devuelve el consumo medio por habitante.
- Método **crearMensaje**, que devuelve el mensaje “999 hogares han consumido un total de 999 m3, con una media de 999 m3/habitante”.

Ejercicio 1: implementar los métodos

Se deben codificar los dos últimos métodos de la clase (**consumoMedioHab** y **crearMensaje**).

Ejercicio de usuario

Ejercicio 2. Método cargarMunicipios

En primer lugar se debe cargar la información de los 5 municipios que consumen agua del depósito. Se piden los datos de un municipio por teclado y se almacenan en un array (**vMunicipios**) que contiene 5 objetos de la clase Municipio. Este array **estará ordenado**, de forma ascendente, **por el código postal**.

A continuación, para gestionar el consumo del depósito, en el programa principal se presenta el siguiente menú:

1. Registrar consumo proximo
2. Estado del depósito
3. Visualizar estadística de municipio
4. Salir

Para diseñar más fácilmente la gestión de las opciones de este menú los alumnos deberán codificar previamente en este examen los tres subprogramas que se indican a continuación y los cuales serán utilizados en la implementación de dichas opciones:

Ejercicio 3. subprograma aguaDisponible

Sabiendo que el depósito tiene una capacidad de 100.000 m³, este método calcula el agua que aún no se ha consumido, restando el consumo realizado hasta el momento por todos los municipios (el consumo por municipio se encuentra en el array **vMunicipios**).

Ejercicio 4. subprograma indiceMunicipio

Este método busca en el array de municipios el código postal que recibe como parámetro y devuelve su índice o posición en dicho array si lo encuentra, y si no devuelve el valor -1 indicando con ello que el código postal no existe en el array.

Ejercicio 5. subprograma leerCPValido

Este método solicita que se introduzca por teclado un código postal y busca en el array de municipios el municipio al que le corresponde dicho código, utilizando para ello el subprograma anterior (**indiceMunicipio**), en caso de que el código sea erróneo se solicitara que se introduzca de nuevo el código postal hasta que este sea correcto y entonces este subprograma devolverá la posición o índice de dicho código postal dentro del array de municipios.

Se muestra a continuación una muestra de lo que se visualiza en la pantalla durante la ejecución de este subprograma:

```
Código postal del municipio: 48008 <R>  
¡Error, ese municipio no existe!  
Código postal del municipio: 48007 <R>
```

Una vez visualizado este proceso y comprobado que el código postal 48007 es correcto se devolverá la posición donde se encontró dicho código en el array.

A continuación en el examen se codificarán los subprogramas que gestionan las opciones del menú utilizando para ello los subprogramas anteriores (ejercicios 3,4 y 5):

Ejercicio 6. (Opción 1) - registrarConsumoProximo

Para gestionar la primera opción del menú se pide la siguiente información por teclado:

```
Código postal del municipio: 48008
¡Error, ese municipio no existe!
Código postal del municipio: 48007

Mes del próximo consumo [1..12]: 19
Mes del próximo consumo [1..12]: 9

Consumo estimado a realizar (m3): 120
```

Una vez leído el código postal por teclado y comprobado que existe en el array de municipios utilizando para ello el método **leerCPValido()** se debe solicitar que se introduzca el mes de consumo comprobando que este sea correcto y una vez introducidos todos los datos correctamente se debe comprobar que se puede realizar ese consumo de agua próximamente, utilizando el subprograma anteriormente implementado **aguaDisponible()**. Si hay agua suficiente se registra el consumo para ese municipio y mes en un array de dos dimensiones denominado **vConsumoMes** (una dimensión del array representa el municipio y la otra el mes). Además se actualizará el consumo total de ese municipio en el array **vMunicipios**.

En caso de que no haya suficiente agua se muestra un mensaje de error al usuario.

Ejercicio 7. (Opción 2) - estadoDeposito

Mediante esta opción del programa se muestra el consumo total realizado hasta el momento. Tened en cuenta que ya habéis implementado un método que calcula el agua disponible.

```
El consumo total realizado es de 54.535 m3
El municipio de mayor consumo por habitante es: Basauri
Sus 1245 hogares han consumido un total de 25876 m3,
con una media de 5 m3/habitante
```

Ejercicio 8. (Opción 3) - visualizarEstadisticaMunicipio

Este subprograma pedirá un código postal válido por teclado, por ejemplo:

```
Código postal del municipio: 48008
¡Error, ese municipio no existe!
Código postal del municipio: 48007
```

y tras validar que existe mostrará el consumo mensual de ese municipio de forma gráfica (una barra por cada 100 m3).

Por ejemplo:

```
El municipio de Bilbao ha tenido un consumo total de 25600 m3
Consumo mensual (- representan 100 m3)
1: ----
```

```
2: -----
3: ---
4: -----
...
...
11: -----
12: -----
```

La clase que contiene estos subprogramas es la que contiene además el método main el cual tendrá la siguiente estructura que deberá respetarse para la implementación de este examen.

```
public static void main(String[] args) {
    // Declaración y creación de los arrays que utiliza el programa
    ...
    // Procesos previos a la gestión del menú
    ...
    // Gestión del menú

    int opc;
    do{
        menu();
        System.out.print("Opción: ");
        opc = Utilidades.leerEntero();

        switch (opc){
            case 1: registrarConsumoProximo(...); break;
            case 2: estadoDeposito(...); break;
            case 3: visualizarEstadisticaMunicipio(...); break;
        }
    }while (opc != 4);
}
```