

## EJERCICIOS Map Curso 2020 2021

### MAP

1. Implementa el control de acceso al área restringida de un programa. Se debe pedir un nombre de usuario y una contraseña. Si el usuario introduce los datos correctamente, el programa dirá "Ha accedido al área restringida". El usuario tendrá un máximo de 3 oportunidades. Si se agotan las oportunidades el programa dirá "Lo siento, no tiene acceso al área restringida". Los nombres de usuario con sus correspondientes contraseñas deben estar previamente almacenados en una estructura de la clase **HashMap**.
2. Crea un mini-diccionario español-inglés que contenga, al menos, 20 palabras (con su correspondiente traducción). Utiliza un objeto de la clase HashMap para almacenar las parejas de palabras. El programa pedirá una palabra en español y dará la correspondiente traducción en inglés.
3. Realiza un programa que escoja al azar 5 palabras en español del mini diccionario del ejercicio anterior. El programa irá pidiendo que el usuario teclee la traducción al inglés de cada una de las palabras y comprobará si son correctas. Al final, el programa deberá mostrar cuántas respuestas son válidas y cuántas erróneas.
4. Escribe un programa que genere una secuencia de 5 cartas de la baraja española y que sume los puntos según el juego de la brisca. El valor de las cartas se debe guardar en una estructura HashMap que debe contener parejas (figura, valor), por ejemplo ("caballo", 3). El valor de las cartas es el siguiente: as → 11, tres → 10, sota → 2, caballo → 3, rey → 4; el resto de cartas no vale nada.

La secuencia de cartas debe ser una estructura de la clase ArrayList que contiene objetos de la clase Carta .

Ejemplo:

as de oros

cinco de bastos

caballo de espadas

sota de copas

tres de oros

Tienes 26 puntos

5. Crear un mapa, que debe contener como clave: el nombre de una asignatura (las asignaturas son las siguientes: PROGRAMACIÓN, LM y BD) y como dato: una lista de cinco alumnos que la cursan con sus notas (clase Alumno: nomApe y nota). Una vez creado este mapa, listarlo por asignatura (EN ORDEN ASCENDENTE DE NOMBRE Y APELLIDOS DE ALUMNO).

DATOS DE ENTRADA: Para PROGRAMACIÓN: Rubén Hernández,5; Susana Núñez,7; Elena Ruiz,6; Guillermo Caro,8; Tania Falero,5. Para LM: Hernán Barco,7; Vicente Guerra, 6; Carmen Vega,8; Belén Cuesta,5; Bertín Blades,7. Para BD: Alejandro Ortiz,5; Daniel Núñez,8; Valeria Fernández, 4; Agustín Soler, 9; Emilio Jerez,3.

A continuación, a los profesores de estas asignaturas les toca el gordo de navidad y deciden poner un cinco a todos los alumnos suspensos. Cambia las calificaciones correspondientes y lista de nuevo el mapa.

Crear tres clases: Alumno, ListAlumno y Principal.

6. Dada la siguiente clase:

```
class JugadorSeleccion {
    private int dorsal;
    private String nombre;
    private String demarcacion;

    public JugadorSeleccion() {
    }

    public JugadorSeleccion(int dorsal, String nombre, String demarcación) {
        this.dorsal = dorsal;
        this.nombre = nombre;
        this.demarcacion = demarcación;
    }

    public int getDorsal() {
        return dorsal;
    }

    public void setDorsal(int dorsal) {
        this.dorsal = dorsal;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getDemarcación() {
        return demarcacion;
    }
}
```

```

public void setDemarcación(String demarcación) {
    this.demarcacion = demarcación;
}
@Override
public String toString() {
    return this.dorsal+" -- "+this.nombre+" -- "+this.demarcacion;
}
}

```

Crear un TreeMap en la que la clave será el nombre del jugador y el valor será un objeto de la clase "JugadorSeleccion". El TreeMap lo declaramos de la siguiente forma y metemos en él los siguientes objetos:

```

System.out.println("***** TreeMap con Objetos y como Clave un String *****");
Map <String, JugadorSeleccion> jugadores = new TreeMap<String, JugadorSeleccion>();
jugadores.put("Casillas", new JugadorSeleccion(1, "Casillas", "Portero"));
jugadores.put("Ramos", new JugadorSeleccion(15, "Ramos", "Lateral Derecho"));
jugadores.put("Pique", new JugadorSeleccion(13, "Pique", "Central"));
jugadores.put("Puyol", new JugadorSeleccion(5, "Puyol", "Central"));
jugadores.put("Capdevila", new JugadorSeleccion(11, "Capdevila", "Lateral Izquierdo"));
jugadores.put("Xabi", new JugadorSeleccion(14, "Xabi Alonso", "Medio Centro"));
jugadores.put("Busquets", new JugadorSeleccion(16, "Busquets", "Medio Centro"));
jugadores.put("Xavi", new JugadorSeleccion(8, "Xavi Hernandez", "Centro Campista"));
jugadores.put("Pedrito", new JugadorSeleccion(18, "Pedrito", "Interior Izquierdo"));
jugadores.put("Iniesta", new JugadorSeleccion(6, "Iniesta", "Interior Derecho"));
jugadores.put("Villa", new JugadorSeleccion(7, "Villa", "Delantero"));

```

Si hacemos un "foreach" de este TreeMap, vamos a ver que el Map va a estar ordenado por orden alfabético de la clave. Es decir, por el nombre de los jugadores:

```

for (Entry<String, JugadorSeleccion> jugador : jugadores.entrySet()){
    String clave = jugador.getKey();
    JugadorSeleccion valor = jugador.getValue();
    System.out.println(clave+" -> "+valor.toString());
}

```

COMPROBAR: Como resultado de esta ejecución del código tenemos el siguiente:

```

Busquets -> 16 -- Busquets    -- Medio Centro
Capdevila -> 11 -- Capdevila   -- Lateral Izquierdo
Casillas  -> 1  -- Casillas    -- Portero
Iniesta   -> 6  -- Iniesta     -- Interior Derecho
Pedrito   -> 18 -- Pedrito     -- Interior Izquierdo
Pique     -> 13 -- Pique       -- Central
Puyol     -> 5  -- Puyol       -- Central
Ramos     -> 15 -- Ramos       -- Lateral Derecho
Villa     -> 7  -- Villa       -- Delantero
Xabi      -> 14 -- Xabi Alonso  -- Medio Centro
Xavi      -> 8  -- Xavi Hernandez -- Centro Campista
¿QUÉ OCURRE SI AÑADIMOS LO SIGUIENTE:?

```

//Cuidado con comparar objetos que son iguales pero no son lo mismo (IDENTIDAD)

```
JugadorSeleccion villa = new JugadorSeleccion(7, "Villa", "Delantero");
```

```
System.out.println("Esta este objeto 'villa' en el Map: jugadores.containsValue(villa) =  
"+jugadores.containsValue(villa));
```

COMO SALIDA OBTENDREMOS:

```
Esta este objeto 'villa' en el Map: jugadores.containsValue(villa) = false
```

EN CAMBIO, si ejecutamos lo que viene a continuación, estaremos preguntando por un objeto que sí que está dentro del Map:

// En este caso si que estamos preguntando por el mismo objeto

```
JugadorSeleccion navas = new JugadorSeleccion(22, "Navas", "Extremo Derecho");
```

```
jugadores.put("Navas", navas);
```

```
System.out.println("Esta este objeto 'navas' en el Map: jugadores.containsValue(navas)  
= "+jugadores.containsValue(navas));
```

Y COMO SALIDA MOSTRARÁ:

```
Esta este objeto 'navas' en el Map: jugadores.containsValue(navas) = true
```

7. Crear una lista de programadores con los siguientes datos. De cada programador se desea almacenar su nombre, su edad y mediante una estructura de mapa los lenguajes de programación (en orden alfabético) que conoce, con la fecha correspondiente de inicio de trabajo con dicho lenguaje. Se pide:

- A. Listar todos los datos ordenados alfabéticamente según el nombre del programador.
- B. Listar todos los datos ordenados según la edad del programador.