

1.

Crea el proyecto **teatrov.1**

Crea el paquete: **paq_Clases** que contiene las clases: **Funcion** y **Teatro**.

Diseña la clase **Funcion** con las siguientes propiedades privadas, nombre, precio.

Constructores:

1. Con las propiedades nombre y precio.
2. Por defecto (nombre en blanco y precio 0)
3. Dando una funcion f, la función creada tomará los datos de la función f.

Métodos getter:

1. **getNombre()**, tal que devuelve el nombre de la función actual
2. **getPrecio()**, tal que devuelve el precio de la función actual.

Métodos setter:

1. **setNombre**, tal que dando un nuevo nombre lo sustituye por el actual
2. **setPrecio**, tal que dado un nuevo precio lo sustituye por el actual, siempre y cuando el precio nunca sea negativo, si es así se lanzará una excepción del tipo **IllegalArgumentException** con el mensaje pertinente.

Métodos:

1. **toString**, tal que devuelva en formato cadena el contenido de la función actual, el nombre y el precio.
2. **equals**, tal que dada otra función devuelve true si la función actual es igual a la de entrada o false si no lo es.

Diseña la clase **Teatro**, con las propiedades privadas nombre y funciones. La propiedad funciones es un array de 4 elementos donde se almacenará las 4 funciones del teatro.

Constructores:

4. Con las propiedades nombre y funciones (se introduce un array de funciones ar, si el array ar tiene una longitud distinta a 4, el constructor debe lanzar una excepción de tipo **IllegalArgumentException** con el mensaje “El número de funciones ha de ser 4”).
5. Por defecto (nombre en blanco y funciones se creará con un array de Funciones).
6. Dando un nombre, el teatro nuevo tomará ese nombre y se creará el array vacío.

Métodos getter:

1. **getNombre()**, tal que devuelve el nombre del teatro actual
2. **getFunciones()**, tal que devuelve un array con las funciones del teatro actual.

Métodos setter:

1. **setNombre**, tal que dando un nuevo nombre lo sustituye por el actual
2. **setFunciones**, tal que dado un array de funciones f, si el tamaño del nuevo array f coincide con el tamaño del funciones del teatro actual, entonces las funciones del teatro actual tomarán el valor de las funciones del array f, en caso contrario, el método lanzará una excepción del tipo **IllegalArgumentException** con el mensaje “Error, solo se admiten 4 funciones”).

Métodos:

1. **toString**, tal que devuelva en formato cadena el contenido del teatro actual, el nombre y aquellas funciones del array funciones del teatro actual que no sean null.
2. **addFuncion**, tal que dada una función nueva f, permita añadirla al array, para ello, se recorre el array desde el primer elemento hasta el último, cuando se detecte un valor null, se ingresa la nueva función. En el caso de que NO haya ningún valor null, lanzad una excepción **IllegalArgumentException** “No se puede añadir más funciones, ya son 4”.
3. **containsFuncion**, tal que dada una función f, comprueba si dentro del array de funciones se encuentra esa función, si fuera así, se devuelve la posición que ocupa esa función en el array, en caso contrario, que devuelva -1.
4. **removeFuncion**, tal que dada una función f, permita borrarla del array, es decir, donde se encuentre el nuevo valor será null. Si la función f, no está, el método enviará una excepción

con el mensaje “Esa función no se encuentra en el array”.

Crea el paquete: **paq_Ppal** que contiene la clase Ppal

Diseña la clase **Ppal**, que contiene el método main() con el siguiente código:

1.

Funcion f1 con los valores "Bodas de sangre",10

Funcion f2 con los valores "Magia",18;

Funcion f3 con los valores "La casa de Bernarda Alba",12

Funcion f4 con los valores "Garbancito",8

2. Crea un array v de tipo Funcion con tamaño fijo 4.
3. En el primer valor del array guarda f1
4. En el segundo valor del array guarda f2
5. En el tercer valor del array guarda f3
6. Crea un nuevo teatro romea con el nombre: “Romea” y el array v
7. Muestra el teatro romea.
8. Guardar en un array w de tipo Funcion el array Funciones del teatro romea.
9. Mostrar por pantalla ese array w. // Intenta utilizar foreach
10. Crea un función f5 con los valores “Cascanueces”,20
11. Añade esa función f5 al teatro romea.
12. Muestra los datos del teatro romea.
13. Comprueba si la función f5 está en el teatro romea, muestra el mensaje “Está”.
14. Borra la función f5 del teatro romea.
15. Muestra los datos del teatro romea.
16. Si la función f3 existe en el teatro romea, entonces cambia su precio a 19€.

2.

Crea el proyecto **teatrov.2**

Crea el paquete: **paq_Clases** que contiene las clases: Funcion y Teatro.

Diseña la clase **Funcion** con las siguientes propiedades privadas, nombre, precio.

Constructores:

1. Con las propiedades nombre y precio.
2. Por defecto (nombre en blanco y precio 0)
3. Dando una funcion f, la función creada tomará los datos de la función f.

Métodos getter:

1. **getNombre()**, tal que devuelve el nombre de la función actual
2. **getPrecio()**, tal que devuelve el precio de la función actual.

Métodos setter:

1. **setNombre**, tal que dando un nuevo nombre lo sustituye por el actual
2. **setPrecio**, tal que dado un nuevo precio lo sustituye por el actual, siempre y cuando el precio nunca sea negativo, si es así se lanzará una excepción del tipo `IllegalArgumentException` con el mensaje pertinente.

Métodos:

1. **toString**, tal que devuelva en formato cadena el contenido de la función actual, el nombre y el precio.
2. **equals**, tal que dada otra función devuelve true si la función actual es igual a la de entrada o false si no lo es.

Diseña la clase **Teatro**, con las propiedades privadas nombre y funciones. La propiedad funciones es un array de funciones cuyo número de elementos es variable. (`ArrayList`).

Constructores:

1. Con las propiedades nombre y funciones (se introduce un array de funciones de tamaño ar, si el array ar tamaño variable `ArrayList`).
2. Por defecto (nombre en blanco y funciones se creará con un `ArrayList`)
3. Dando un nombre al teatro que se crea, se creará un nuevo `ArrayList` de tipo `Funcion` para las funciones.

Métodos getter:

1. **getNombre()**, tal que devuelve el nombre del teatro actual
2. **getFunciones()**, tal que devuelve un array con las funciones del teatro actual.

Métodos setter:

1. **setNombre**, tal que dando un nuevo nombre lo sustituye por el actual
2. **setFunciones**, tal que dado un array de funciones f.

Métodos:

1. **toString**, tal que devuelva en formato cadena el contenido del teatro actual, el nombre y aquellas funciones del array funciones del teatro actual que no sean null.
2. **addFuncion**, tal que dada una función nueva f, permita añadirla al array.
3. **containsFuncion**, tal que dada una función f, comprueba si dentro del array de funciones se encuentra esa función, devuelve true, en caso contrario false.
4. **removeFuncion**, tal que dada una función f, permita borrarla del array, si la función f, no está, el método enviará una excepción con el mensaje "Esa función no se encuentra en el array".

Crea el paquete: **paq_Ppal** que contiene la clase **Ppal**

Diseña la clase **Ppal**, que contiene el método `main()` con el siguiente código:

1.

Funcion f1 con los valores "Bodas de sangre",10

Funcion f2 con los valores "Magia",18;

Funcion f3 con los valores "La casa de Bernarda Alba",12

Funcion f4 con los valores "Garbancito",8

2. Crea un ArrayList v de tipo Funcion.
3. En el primer valor del ArrayList guarda f1
4. En el segundo valor del ArrayList guarda f2
5. En el tercer valor del ArrayList guarda f3
6. Crea un nuevo teatro romea con el nombre: "Romea" y el ArrayList v
7. Muestra el teatro romea.
8. Guardar en un ArrayList w de tipo Funcion el ArrayList Funciones del teatro romea.
9. Mostrar por pantalla ese ArrayList w. // Intenta utilizar foreach
10. Crea un función f5 con los valores "Cascanueces",20
11. Añade esa función f5 al teatro romea.
12. Muestra los datos del teatro romea.
13. Comprueba si la función f5 está en el teatro romea, muestra el mensaje "Está".
14. Borra la función f5 del teatro romea.
15. Muestra los datos del teatro romea.
16. Si la función f3 existe en el teatro romea, entonces cambia su precio a 19€.