

Ejercicio No.6: Comprobacion

1. Una clase es la definición de un objeto. Así como las personas nos diferenciamos de las plantas, los objetos se diferencian entre sí por sus clases. Cada clase puede poseer atributos y métodos. Los atributos son características (valores constantes, variables u otros objetos) que luego cualquier objeto creado de una clase va a tener. Los métodos son funciones o acciones que también son utilizables por los objetos de la clase. Un objeto o instancia es, por decirlo así, una unidad de una determinada clase. Los objetos e instancias son los que a la hora de correr un programa trabajan y funcionan. Por otro lado, la clase es el diseño de los objetos.
2. El estado de un objeto es cómo se encuentra éste en un momento específico. Algunos editores de texto como *Eclipse* o *IntelliJ* permiten a los programadores debugear sus proyectos, corriendo dicho proyecto línea por línea. Estos debug wizards, como suelen llamarlos, le permiten a uno visualizar el estado de todos los objetos y variables en uso. Básicamente lo que muestran es el nombre del objeto o variable y todos los valores que estos tengan guardados; este es el estado de un objeto.
3. Un commit es generar un aporte o avance a un proyecto o repositorio en github.
4. La función map recibe de parámetro a otra función que luego aplica a todos los objetos especificados de la lista especificada.
5. Para que los atributos de los objetos sean manipulables únicamente a través de los métodos de los mismos y por lo tanto, la lógica de cómo funcionan los objetos y cómo éstos utilizan sus atributos esté definida por el programador y no el usuario.

6.

Cañonera
<ul style="list-style-type: none"> - encendido/apagado: Boolean - input: Int (dependiendo del entero, se utiliza diferente input) - volumen: Int - brillo: Int - enfoque: Int
<ul style="list-style-type: none"> - encender/apagar: void - cambiarInput: void - subirVol: void - bajarVol: void - subirBrill: void - bajarBrill: void - subirEnfoque: void - bajarEnfoque: void

7.

Cajero
<ul style="list-style-type: none"> - dineroActual: Int - usuarioActual: String - abierto/cerrado: Boolean (se refiere a la gabeta donde está el dinero) - activado/bloqueado: Boolean (se refiere a si el usuario fue autenticado o no)
<ul style="list-style-type: none"> - cobrar(monto: Int, numTarjeta: Int): Boolean (el booleano es para determinar si se logró cobrar o no) - ingresarUsuario(username: String, password: String)

- abrir/cerrar: int

Factura
- identificador: Int - numeroTickets: Int - nombrePelicula: String - numeroSala: Int - montoCobrado: Double

ImpresoraTickets
- factura: Factura - cantTinta: Int - hayTinta: Boolean
- Imprimir (usa su propio atributo factura): void (imprime físicamente una factura)

8. Es el nombre con el cual uno llama a dicha función.
9. Es un objeto que puede que sea de tipo Int o de tipo String, por decir un ejemplo. Esto se utiliza en ocasiones en las que se le pide a un usuario que ingrese algún dato y dependiendo de que ingrese el usuario, el objeto de tipo maybe puede tomar “forma” de string o de int.
10. Me gusta la dinámica de esta clase. Aprovechando que estamos aprendiendo diferente a las otras secciones y que estamos usando kotlin además de java, me gustaría ver más contenido en canvas y/o en el repositorio de github del curso de programación sobre otros idiomas de programación, contenido acerca de cómo usar la nube de IBM o otros servicios de ese tipo para correr aplicaciones o ejercicios extras. Claro, esto sería extra al curso. En cuanto al curso, todo está fluyendo muy bien.