

Universidad del Valle de Guatemala

Programación Orientada a Objetos

Tomás Gálvez P.



Laboratorio #4

Análisis y Diseño

Fernanda Esquivel de León - 21542

Francisco Javier Catillo Cerna - 21562

Gustavo Andrés González Pineda - 21438

Guatemala, 16 de noviembre de 2021

Descripción de las clases

Clase Carro

Esta clase abstracta madre almacenará toda la información de los carros. Poseerá métodos como setters y getters para las funcionalidades de los carros.

- **Propiedades**

- encendido: boolean que indicará si el carro se encuentra encendido (true) o apagado (false)
- modoRadio: string que guardará el modo del radio (AM/FM)
- emisoraActual: double que guardará la emisora actual de la radio
- emisorasGuardadas: array de 50 doubles que guardará las emisoras guardadas en la radio del carro
- volumenActual: int que guardará la cantidad de volumen actual del radio
- telefonoConectado: boolean que indicará si hay un teléfono conectado (true) o no (false)
- contactos: array de 5 contactos, donde se guardarán objetos de tipo contacto, los cuales almacenan la información de los contactos del radio
- estaEnLlamada: boolean que indicará si el usuario se encuentra en una llamada (true) o no

- **Métodos**

- Constructor: inicializa las propiedades de la clase
- subirVolumen(): sube una unidad del volumen al radio (+1)
- bajarVolumen(): baja una unidad del volumen al radio (-1)
- cargarEmisora(posiciónEmisora: int): sintoniza una de las emisoras guardadas del radio, esta recibe la posición de la misma
- getContactos(): devuelve un array con los contactos de la radio
- finalizarLlamada(): devuelve un string con un mensaje que indica que se finalizó la llamada
- getNombreFuncionProductividad(): método abstracto que devuelve un string con el tipo de función de productividad correspondiente al tipo de carro donde se implemente

Clase CarroC

Esta clase hija almacenará toda la información específica de los carros de tipo C.

- **Propiedades**
 - llamadaEnEspera: boolean que indicará si la llamada actual está en espera (true) o no (false)
- **Métodos**
 - Constructor: inicializa las propiedades de la clase

Clase CarroA

Esta clase hija almacenará toda la información específica de los carros de tipo A.

- **Propiedades**
 - ultimoContacto(): propiedad de tipo contacto que almacenará el último contacto al cual se le realizó una llamada
 - nombre: string que almacena el nombre del usuario para la tarjeta de presentación //CREO QUE ESTO HAY QUE QUITARLO, PERO POR SI ACASO LO DEJO
 - numero: string que almacena el número del usuario para la tarjeta de presentación //CREO QUE ESTO HAY QUE QUITARLO, PERO POR SI ACASO LO DEJO
- **Métodos**
 - Constructor: inicializa las propiedades de la clase

Clase CarroS

Esta clase hija almacenará toda la información específica de los carros de tipo S.

- **Propiedades**
 - N/A
- **Métodos**
 - Constructor: inicializa las propiedades de la clase

Clase Canción

Esta clase almacenará toda la información de las canciones de la radio. También poseerá getters para acceder a dicha información.

- **Propiedades**

- nombre: string que almacena el nombre de la canción
- duracion: string que almacena la duración de la canción en formato mm:ss
- autor: string que almacena el autor de la canción
- playlist: string que almacena la playlist a la que pertenece de la canción

- **Métodos**

- Constructor: inicializa las propiedades de la clase
- getNombre(): devuelve un string con el nombre de la canción
- getDuracion(): devuelve un string con la duración de la canción
- getAutor(): devuelve un string con el autor de la canción
- getPlaylist(): devuelve un string con la playlist a la que pertenece de la canción

Clase Contacto

Esta clase almacenará toda la información de los contactos del radio. También poseerá getters para acceder a dicha información.

- **Propiedades**

- nombre: string que almacena el nombre del contacto
- numero: string que almacena el número del contacto

- **Métodos**

- Constructor: inicializa las propiedades de la clase
- getNombre(): devuelve un string con el nombre del contacto
- getNumero(): devuelve un string con el número del contacto

Clase MercedesBenz

Esta clase manejará el programa.

- **Propiedades**

- carroActual: propiedad de tipo Carro que almacenará el carro actual que se está manejando
- listaPop: arraylist de canciones que almacenará canciones de género pop
- listaRock: arraylist de canciones que almacenará canciones de género rock

- listaRockAlternativo: arraylist de canciones que almacenará canciones de género rock alternativo
- **Métodos**
 - Constructor: inicializa las propiedades de la clase

Clase Vista

Esta clase mostrará en pantalla la información que el usuario requiera, además solicitará los datos que sean necesarios para la ejecución del programa y los enviará al controlador.

- **Atributos**
 - Scan: Scanner que leerá los datos ingresados por el usuario
- **Métodos**
 - mostrarInicio(): muestra un mensaje de bienvenida al programa.
 - mostrarMenu(): muestra el menú al usuario y devuelve un entero con la opción seleccionada.
 - print(): Metodo privado hecho para abreviar el print.
 - separador(): Se usa para seleccionar diferentes partes en la consola.
 - opcionInvalida(): Mensaje que se le presenta al usuario en caso ocurra un error mientras ingresa los datos.
 - inicio(): Despliega el menú de inicio.
 - deseaEncenderCarro(): menú de encender carro.
 - seleccionarModo(): le permite al usuario seleccionar el modo del carro.
 - seleccionarCarro(): le permite al usuario seleccionar el tipo de carro.
 - mostrarMenuProductividad(): Despliega el menú de productividad.
 - mostrarPantallaModoProductividad(): Despliega la pantalla del modo productividad.
 - mostrarModoProductividad(): Despliega el texto devuelto por los métodos.
 - deseaEncenderRadio(): Se le pregunta al usuario si quiere encender la radio.
 - mostrarMenuRadio(): muestra el menú de la radio.
 - mostrarMenuReproduccion(): muestra el menú de reproducción.
 - seleccionarPlaylist(): le permite al usuario seleccionar la playlist.
 - deseaConectarTelefono(): le permite al usuario conectar el teléfono.
 - mostrarMenuTelefono(): Despliega el menú del teléfono.
 - pantallaRadioApagado(): Despliega la pantalla del radio cuando está apagado.
 - pantallaEnModoReproducción(): Despliega la pantalla en el modo reproducción.

- `mostrarReproducirCancion()`: Muestra la cancion que se está reproduciendo.
- `pantallaEnModoRadio()`: Muestra la pantalla en el modo radio.
- `pantallaEnModoTelefono()`: Muestra la pantalla en el modo teléfono
- `pantallaCarroApagado()`: Muestra la pantalla cuando el carro está apagado.
- `pedirEmisora()`: Le pregunta al usuario la emisora.
- `emisorasLlenas()`: Despliega un mensaje cuando las emisoras de radio están llenas y no se pueden guardar más.
- `askContactIndex()`: Le pregunta al usuario el contacto que desea elegir.
- `llamandoContacto()`: Despliega el contacto al que se está llamando.
- `showContacts()`: Muestra los contactos
- `llamadaFinalizada()`: Muestra un mensaje cuando se finaliza la llamada.
- `noEstaEnLlamada()`: Muestra un mensaje cuando el usuario solicita una funcionalidad que solo puede ser aplicada si éste está en llamada.
- `estaEnLlamada()`: Muestra un mensaje cuando el usuario solicita una funcionalidad que solo puede ser aplicada si éste NO está en llamada.
- `printMensajeTelefono()`: Despliega un mensaje según la funcionalidad específica aplicada a través de interfaces.

Clase Main

Esta clase será el main del programa, ejecutará el modelo y la vista

- **Atributos**
 - N/A
- **Métodos**
 - `main(args)`: inicia la ejecución del programa, es decir, realiza todas las acciones de este

Descripción de las interfaces

Interfaz ModoTelefono

Esta interfaz manejará las funcionalidades del modo teléfono del radio.

- **Propiedades**
 - N/A

- **Métodos**

- llamarContacto(c: Contacto): método que recibe un Contacto y devuelve un string con mensaje que indica que se está realizando una llamada a dicho contacto
- modoTelefonoEspecifico(): método que devolverá un mensaje dependiendo de la funcionalidad específica del tipo de carro donde se implemente el método

Interfaz ModoReproduccion

Esta interfaz manejará las funcionalidades del modo reproducción del radio.

- **Propiedades**

- N/A

- **Métodos**

- seleccionarLista(): selecciona una lista de reproducción de las disponibles en el radio
- cambiarCanción(): cambia la canción y devuelve un mensaje con la información de la canción actual
- escucharCancion(c: Contacto): selecciona una canción para escuchar y devuelve un mensaje que indica que se está reproduciendo dicha canción

Interfaz ModoRadio

Esta interfaz manejará las funcionalidades del modo reproducción del radio.

- **Propiedades**

- N/A

- **Métodos**

- cambiarModo(): cambia el modo de la radio (AM/FM)
- bajarEmisora(): cambia la emisora actual a una secuencia 0.5 menor (-0.5)
- subirEmisora(): cambia la emisora actual a una secuencia 0.5 mayor (+0.5)

Organización del trabajo

Requisitos funcionales

- Seleccionar Tipo de Carro
 - Encender Carro
 - SIEMPRE MOSTRAR PANTALLA
 - Entrar en Modo Radio
 - Encender Radio
 - Subir Volumen
 - Bajar Volumen
 - Adelantar una Emisora
 - Regresar una Emisora
 - Guardar Emisora actual
 - Cargar Emisora
 - Apagar Radio
 - Regresar a Modos
 - Entrar en Modo Reproducción
 - Seleccionar Lista de reproducción
 - (Mostrar Listas de reproducción y seleccionar una)
 - Adelantar Canción
 - Regresar una Canción
 - Escuchar canción
 - Entrar en Modo Teléfono
 - Conectar/Desconectar Teléfono
 - Mostrar Contactos
 - Llamar a Contacto
 - (Mostrar contactos y seleccionar uno)
 - Finalizar llamada
 - Entrar en Modo Productividad
 - Apagar Carro

Pantallas:

RADIO APAGADO

CARRO APAGADO

MODO: RADIO

BANDA: AM
EMISORA: 94.5

MODO: REPRODUCCIÓN

EN REPRODUCCIÓN:
nombreCancion
artista
duracion
genero

MODO: TELÉFONO

TELÉFONO: CONECTADO/DESCONECTADO
NO HAY NINGUNA LLAMADA EN CURSO/ EN LLAMADA

MODO: PRODUCTIVIDAD

PLANIFICACIÓN DEL VIAJE/ MOSTRAR NOMBRE DE TARJETA/ PRONÓSTICO
*depende de la clase del carro

PENDIENTES:

- Modo productividad //Franz
 - Implementar pantalla
 - Implementar el texto de los modos específicos en las clases hijas
- Modo Radio //Fer
 - Pantalla
 - Cambiar fm \longleftrightarrow am
 - cambiar emisora
 - guardar emisora
 - cargar emisora
- Modo reproducción //Franz
 - Seleccionar lista de reproducción
 - Cambiar canción
 - Adelantar
 - Atrasar
- Modo teléfono //Gus
 - Pantalla
 - Mostrar contactos
 - llamar contacto
 - finalizar llamada
 - las funcionalidades específicas en las clases hijas