UNIVERSIDAD DEL VALLE DE GUATEMALA

Guatemala 11 de Enero de 2024
Adrián López - 231361
Luis Fernando Mendoza -

Algoritmos y Estructura de Datos

Hoja de Trabajo 1, Programa de Radio

Objetivos:

a. Utilizar GIT (u otro software para control de versiones) para guardar las versiones del programa.
b. Practicar la definición de interfaces y reutilización de código.
c. Emplear JUnit para casos de prueba.
d. Repasar el diagrama UML de clases.
e. Repasar el lenguaje JAVA.

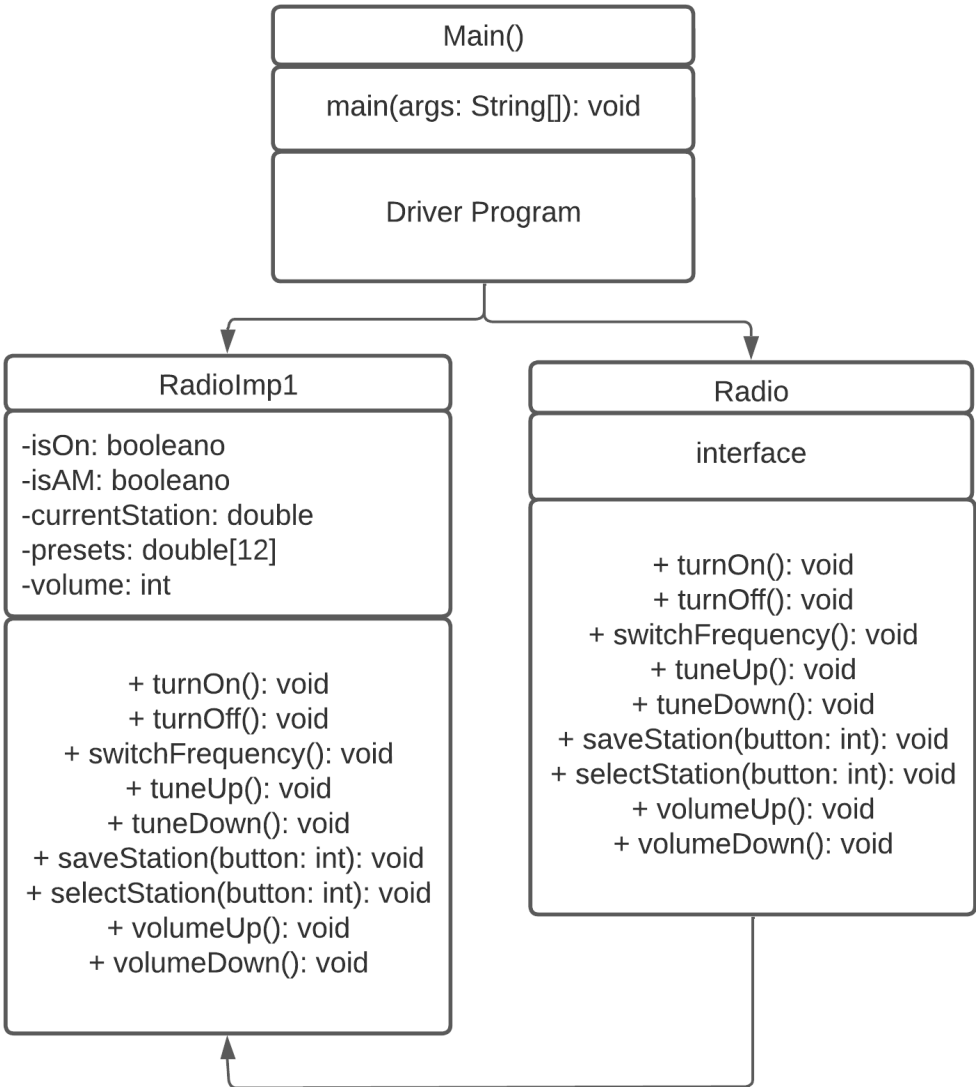
"Design a data structure to simulate the workings of a radio. The state of the radio is on or off, and it may be used to listen to an AM or FM station. A dozen modifiable push buttons (identified by integers 1 through 12) allow the listener to store and recall AM or FM frequencies. AM frequencies can be represented by multiples of 10 in the range 530 to 1610. FM frequencies are found at multiples of 0.2 in the range 87.9 to 107.9."

El programa debe de:

1. Prender la radio.
2. Cambiar de AM a FM a AM.
3. Avanzar en el dial de las emisoras. Al llegar al final del dial inicia nuevamente.
4. Permite guardar una emisora en uno de los 12 botones.
5. Permitir seleccionar la emisora puesta en un botón.
6. Apagar la radio.

-

Diagrama UML:



Evidencia de Desarrollo (Github):
https://github.com/adirnnn/HDT1-Radio-Program.git

Análisis:
Interfaz Radio:
Métodos:

- turnOn(): Enciende el radio y muestra un mensaje indicando que el radio está ahora encendido.
- turnOff(): Apaga el radio y muestra un mensaje indicando que el radio está ahora apagado.
- switchFrequency(): Cambia la frecuencia del radio entre AM y FM y muestra un mensaje indicando la nueva frecuencia seleccionada.

- tuneUp(): Sintoniza hacia arriba la estación actual, ajustando la estación según la frecuencia (AM o FM) y muestra un mensaje indicando la nueva estación sintonizada.
- tuneDown(): Sintoniza hacia abajo la estación actual, ajustando la estación según la frecuencia (AM o FM) y muestra un mensaje indicando la nueva estación sintonizada.
- saveStation(int button): Guarda la estación actual en un botón específico, mostrando un mensaje indicando en qué botón se guardó la estación.
- selectStation(int button): Selecciona la estación previamente guardada en un botón específico y muestra un mensaje indicando la estación seleccionada.
- volumeUp(): Aumenta el volumen en incrementos de 2, ajustando al volumen máximo si se excede, y muestra un mensaje indicando el nuevo nivel de volumen.
- volumeDown(): Disminuye el volumen en incrementos de 2, ajustando al volumen mínimo si se excede, y muestra un mensaje indicando el nuevo nivel de volumen.

Clase RadioImp1:
La clase RadioImp1 es la implementación del programa mediante la interfaz Radio y proporciona una implementación concreta de cada uno de los métodos de la interfaz. Esta fue hecha específicamente con el objetivo en mente de poder ser intercambiada por otra implementación distinta. Además de los métodos de la interfaz, contiene los siguientes atributos:

- isOn: Un booleano que indica si el radio está encendido o apagado.
- isAM: Un booleano que indica si la frecuencia actual es AM o FM.
- currentStation: Un double que almacena la estación actual del radio.
- presets: Un arreglo de dobles que almacena las estaciones guardadas en los botones.
- volume: Un entero que representa el nivel de volumen actual.