Lab Activity 3

Introduction

Everyone is familiar with a television. It is the object we are going to create in this lab. First we need a blueprint. All manufacturers have the same basic elements in the televisions they produce as well as many options. We are going to work with a few basic elements that are common to all televisions. Think about a television in general. It has a brand name (i.e. it is made by a specific manufacturer). The television screen has a specific size. It has some basic controls. There is a control to turn the power on and off. There is a control to change the channel. There is also a control for the volume. At any point in time, the television's state can be described by how these controls are set. We will write the television class. Each object that is created from the television class must be able to hold information about that instance of a television in fields. So a television object will have the following attributes:

• **manufacturer**. The manufacturer attribute will hold the brand name. This cannot change once the television is created, so will be a named constant.

• **screenSize.** The screenSize attribute will hold the size of the television screen. This cannot change once the television has been created so will be a named constant.

• **powerOn.** The powerOn attribute will hold the value true if the power is on, and false if the power is off.

• **channel.** The channel attribute will hold the value of the station that the television is showing.

• **volume.** The volume attribute will hold a number value representing the loudness (0 being no sound). These attributes become fields in our class. The television object will also be able to control the state of its attributes. These controls become methods in our class.

• **setChannel.** The setChannel method will store the desired station in the channel field.

• power. The power method will toggle the power between on and off, changing the value stored in the powerOn field from true to false or from false to true.

• **increaseVolume.** The increaseVolume method will increase the value stored in the volume field by 1.

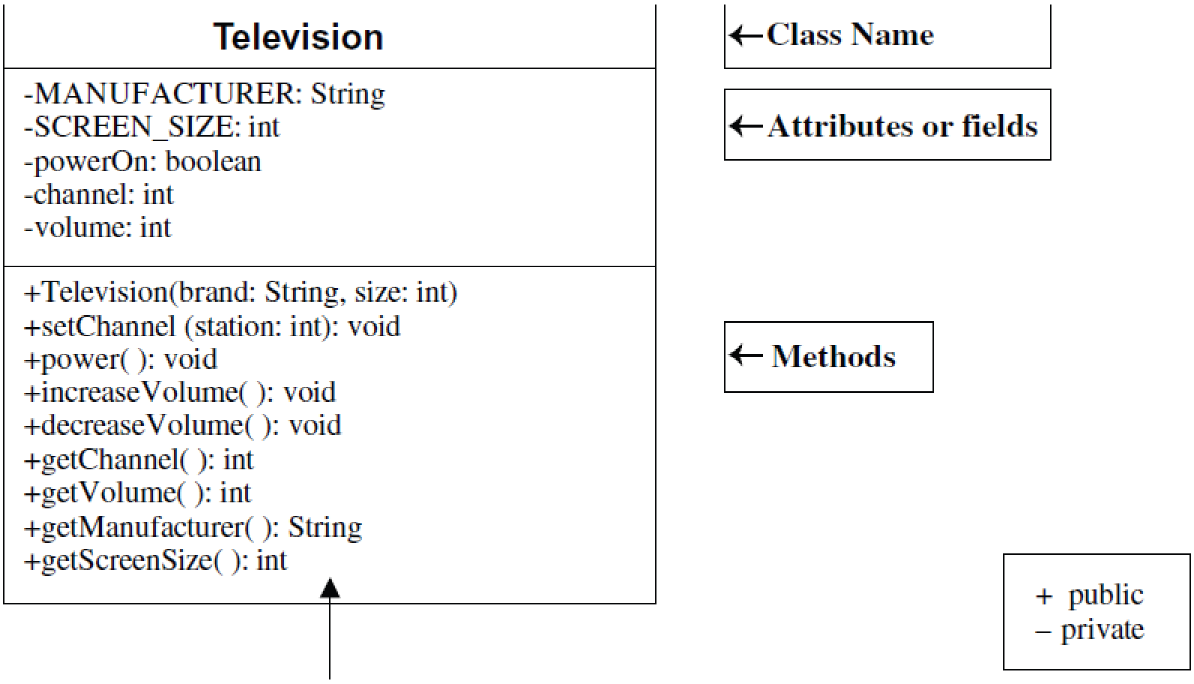
• **decreaseVolume.** The decreaseVolume method will decrease the value stored in the volume field by 1.

• **getChannel.** The getChannel method will return the value stored in the channel field.

• **getVolume.** The getVolume method will return the value stored in the volume field.

• **getManufacturer**. The getManufacturer method will return the constant value stored in the MANUFACTURER field.

• **getScreenSize.** The getScreenSize method will return the constant value stored in the SCREEN_SIZE field.



Television

-MANUFACTURER: String
-SCREEN_SIZE: int
-powerOn: boolean
-channel: int
-volume: int

+Television(brand: String, size: int)
+setChannel (station: int): void
+power( ): void
+increaseVolume( ): void
+decreaseVolume( ): void
+getChannel( ): int
+getVolume( ): int
+getManufacturer( ): String
+getScreenSize( ): int

←Class Name

←Attributes or fields

←Methods

+ public
– private

Task #1 Creating a New Class

1. In a new file, create a class definition called Television.
2. Put a program header (comments/documentation) at the top of the file
3. Declare the fields listed in the diagram.
4. Write a comment for each field indicating what it represents.
5. Save this file as *Television.h*
6. Compile and debug. Do not run.

Task #2 Methods

Create a Television.cpp file that contains the following:

1. Define methods called getVolume, getChannel, getManufacturer, and getScreenSize that return the value of the corresponding field.
2. Define a method called setChannel accepts a value to be stored in the channel field.
3. Define a method called power that changes the state from true to false or from false to true.
4. Define two methods to change the volume. One method should be called increaseVolume and will increase the volume by 1. The other method should be called decreaseVolume and will decrease the volume by 1.

Task #3 Running the application

1. You can only execute (run) a program that has a main method, so there is a driver program that is already written to test out your Television class. Create class TelevisionDemo.cpp that contains the main() function and test whether all the methods work properly or not.

------------------------ The End☹--------------------------------