Lab4A: Mobile Device

Inheritance and File

- This lab exercise is divided into four stages – Stage 1, Stage 2, Stage 3 and Stage 4.
- You need to complete Stage 1 without errors before you can proceed to Stage 2, and complete Stage 2 without errors before you can proceed to Stage 3 and Stage 4.

Question:

Consider the following UML inheritance class diagram shown in Figure1 and the class interface tables:

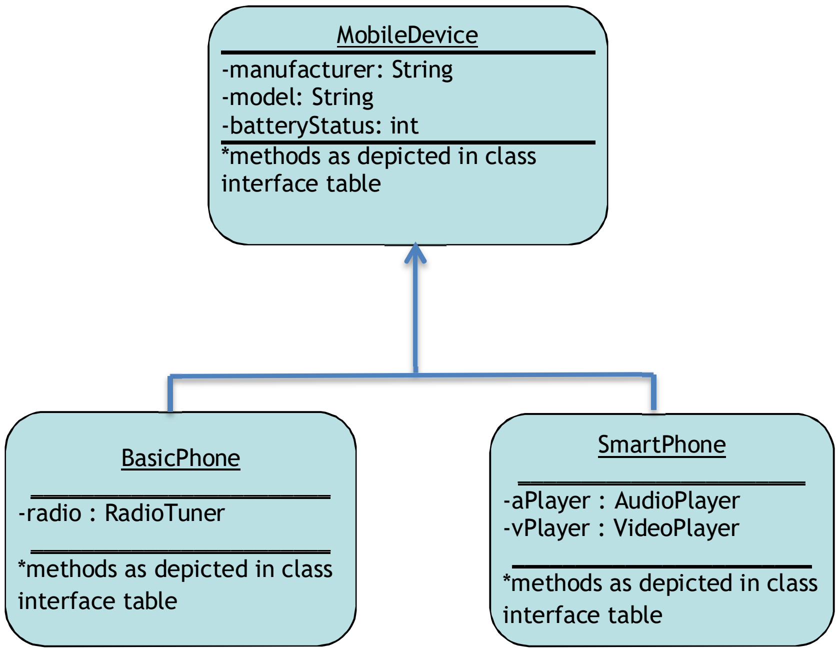

Figure 1

Problem Description:

In this exercise, tester app will read data from 2 input files namely `mobilephone1.dat` and `mobilephone2.dat`.

A basic phone can perform the following functions:

- display the details of the phone
- check if the phone battery needs to be recharged
- display the current setting of the radio station and frequency
- tune the phone's radio to a new station and frequency

While a smart phone can perform the following functions:

- display the details of the phone
- check if the phone battery needs to be recharged
- display the current audio clip played
- set a new audio clip to be played
- display the current video clip played
- set a new video clip to be played

Stage 1:

1. Create a Java project named `Lab4AStage1`.
2. Copy file `Tester1.java` into your project.
3. Create a text file named `mobilephone1.dat` and copy the content from the file provided. (Right-click on project->new File)
4. Define a class named `RadioTuner` (in file RadioTuner.java) according to the following class interface:

RadioTuner	**Description**
- station:String - frequency:double	- variable to store the station name - variable to store the frequency of an FM radio station
+ RadioTuner() + RadioTuner(st :String, fr: double) + setStation (st : String) : void + setFrequency (fr : double) : void + getStation (): String + getFrequency() : double	- Default constructor to set the station and frequency to default values - station = "Mix FM" - frequency = 94.5; - Constructor to set the set the station and frequency to the values specified by the user - Method to set the fm radio station value specified by the user - Method to set the frequency to the value specified by the user - Method to get the name of the radio station - Method to get the frequency of FM radio station

Define a class named `AudioPlayer` (in file AudioPlayer.java) according to the following class interface:

AudioPlayer	**Description**
- audioClip:String	- variable to store the name of the audio clip

+ AudioPlayer()
 	
+ AudioPlayer(ac :String)
 	
+ setAudioClip(ac : String) : void
 	
+ getAudioClip(): String
	- Default constructor to set the audio clip to a default value "You Raise Me Up"
	
	- Constructor to set the audioClip to the values specified by the
	
	- Method to set the audioClip to the value specified by the user
	
	- Method to get the audio clip

Define a class named `VideoPlayer` (in file VideoPlayer.java) according to the following class interface:

VideoPlayer	Description
- videoClip:String	- variable to store the name of the video clip

+ VideoPlayer()
 	
+ VideoPlayer(vc :String)
 	
+ setVideoClip(vc: String): void
 	
+ getVideoClip(): String
	- Default constructor to set the video clip to a default value "Mr.Bean's Holiday";
	
	- Constructor to set the videoClip to the value specified by the user
	
	- Method to set the video clip to the value specified by the user
	
	- Method to get the video clip

Define a class named `MobileDevice` (in file Mobile.java) according to the following class interface (except for method `needCharging()` and `recharge()`)

MobileDevice	Description
- manufacturer:String - model:String - batteryStatus:int	- variable to store the manufacturer of the mobile device - variable to store the model of the mobile device - variable to store the battery status
+ MobileDevice(ma:String, mo:String, bs: int) + setManufacturer (ma : String) : void + setModel(mo : String) : void + setBatteryStatus(bs : int) : void + getManufacturer (): String + getModel() : String + getBatteryStatus() : int + printDetails() : void + needCharging() : boolean	- Default constructor to set the manufacturer, model and battery status to the values specified by the user - Method to set manufacturer to the value specified by the user - Method to set the model to the value specified by the user - Method to set the battery status to the value specified by the user - Method to get the manufacturer of mobile device - Method to get the model of mobile device - Method to get the battery status of mobile device - Method to display the current values of mobile device attributes. Output format as follows: Manufacturer: <manufacturer> Model: <model> Battery Status: <batteryStatus> - Method to check the status of the device battery using method getBatteryStatus() returns true if the value is <=10 (use constant `LOW_BATTERY`, false if otherwise (Postcondition: true is returned if, false is returned if otherwise)

+ recharge() : void	- Method to recharge the battery (and set the battery status to value 100 (use constant `FULL_BATTERY)` by using method setBatteryStatus()

Define a class named `BasicPhone` (in file BasicPhone.java), subclass of `MobileDevice` according to the following class interface (except for `setRadioSetting()`):

BasicPhone	**Description**
- radio: RadioTuner	- variable which refers to a RadioTuner object which stores information on the phone's radio station and frequency.
+ BasicPhone (String ma, String mo, int bs, RadioTuner ra) + getRadio() : RadioTune + setRadioSetting(st:String, fr:double): void + printDetails(): void	- Constructor with parameters to set the values of the attributes. Hint: call super(ma, mo, bs); - Method to return the RadioTuner object - Method to change/tune the radio to the specified station and frequency - Method to display the values of all variables of the basic phone. Hint: call super.printDetails() and other respective methods. Output format: Basic phone details Manufacturer: <manufacturer> Model: <model> Battery Status: <batteryStatus> Station: <station> Frequency: <frequency>

Define a class named `SmartPhone` (in file SmartPhone.java) according to the following class interface: (except for `setCurrentAudio()` and `setCurrentVideo()`)

SmartPhone	**Description**
- aPlayer : AudioPlayer - vPlayer : VideoPlayer	- AudioPlayer object which store information on the phone audio player - VideoPlayer object which stores information on the phone video player. -
+ SmartPhone(ma:String, mo: String, bs: int, ap: AudioPlayer, vp: VideoPlayer) + currentAudioPlaying() : void + currentVideoPlaying() : void + setCurrentAudio(ac: String) : void + setCurrentVideo(vc: String) : void	- Constructor with parameters to set the values of attributes as specified by the user. Hint: call super(ma, mo, bs); - Method to display the audio clip currently playing Note: check the sample output for output format - Method to display the video clip currently playing Note: check the sample output for output format - Method to set the current audio clip to the value specified by the user - Method to set the current video clip to the value specified by the user

+ printDetails() : void	- Method to display the values of all variables of the device. Hint: call super.printDetails() and other respective methods Output format: Smart phone details Manufacturer: <manufacturer> Model: <model> Battery Status: <batteryStatus> Audio playing: <audioClip> Video playing: <videoClip>

Check your answer by invoking the `main` method in class `Tester1` (just run project as Java Application) and your output should be as follows:

```
Basic phone details
Manufacturer: Nokia
Model: 150DualSim
Battery Status: 10
Station: Mix FM
Frequency: 94.5

Basic phone details
Manufacturer: Nokia
Model: 3310
Battery Status: 30
Station: Mix FM
Frequency: 94.5

Basic phone details
Manufacturer: Samsung
Model: GuruFM
Battery Status: 40
Station: Mix FM
Frequency: 94.5

Basic phone details
Manufacturer: Samsung
Model: Rugby3
Battery Status: 90
Station: Mix FM
Frequency: 94.5

Basic phone details
Manufacturer: SonyEriccson
Model: Walkman
Battery Status: 5
Station: Mix FM
Frequency: 94.5

Smart phone details
Manufacturer: Samsung
Model: S8
Battery Status: 60
Audio playing: You Raise Me Up
Video playing: Mr.Bean's Holiday

Smart phone details
```

```
Manufacturer: Apple
Model: iPhone7
Battery Status: 10
Audio playing: You Raise Me Up
Video playing: Mr.Bean's Holiday

Smart phone details
Manufacturer: Huawei
Model: P8Lite
Battery Status: 10
Audio playing: You Raise Me Up
Video playing: Mr.Bean's Holiday

Smart phone details
Manufacturer: Oppo
Model: R9s
Battery Status: 80
Audio playing: You Raise Me Up
Video playing: Mr.Bean's Holiday
```

*Proceed to Stage 2 only after you have completed Stage 1 without errors.

Stage 2:

1. Create a Java project named `Lab4AStage2`.
2. Copy file `Tester2.java` into your project.
3. Copy files `RadioTuner.java`, `AudioPlayer.java`, `VideoPlayer.java`, `MobileDevice.java`, `BasicPhone.java` and `SmartPhone.java` in Stage 1 into your Stage 2 Java project.
4. Create a text file named `mobilephone2.dat` and copy the content from the files provided. (Right-click on project->new File) /Or copy the file provided into current project workspace.

Check your answer by running `Tester2` and your output should be as follows:

```
Basic phone details
Manufacturer: Nokia
Model: 150DualSim
Battery Status: 10
Station: Mix.fm
Frequency: 94.5

Basic phone details
Manufacturer: Nokia
Model: 3310
Battery Status: 30
Station: Ikim.fm
Frequency: 91.5

Basic phone details
Manufacturer: Samsung
Model: GuruFM
Battery Status: 40
Station: THR.fm
Frequency: 99.3

Basic phone details
Manufacturer: Samsung
Model: Rugby3
```

```
Battery Status: 90
Station: Hitz.fm
Frequency: 92.9

Basic phone details
Manufacturer: SonyEriccson
Model: Walkman
Battery Status: 5
Station: TraXX.fm
Frequency: 90.3

Smart phone details
Manufacturer: Samsung
Model: S8
Battery Status: 60
Audio playing: Assalamualaikum
Video playing: Robocop

Smart phone details
Manufacturer: Apple
Model: iPhone7
Battery Status: 10
Audio playing: Hello
Video playing: Terminator

Smart phone details
Manufacturer: Huawei
Model: P8Lite
Battery Status: 10
Audio playing: Crush
Video playing: iRobot

Smart phone details
Manufacturer: Oppo
Model: R9s
Battery Status: 80
Audio playing: Isabella
Video playing: Tunnel
```

*Proceed to Stage 3 only after you have completed Stage 2 without errors.

Stage 3:

1. Create a Java project named `Lab4AStage3`.
2. Copy file `Tester2.java` into your project.
3. Copy files `RadioTuner.java`, `AudioPlayer.java`, `VideoPlayer.java`, `MobileDevice.java`, `BasicPhone.java` and `SmartPhone.java` in Stage 1 into your Stage 2 Java project.
4. Create a text file named `mobilephone2.dat` and copy the content from the files provided. (Right-click on project->new File) /Or copy the file from previous project workspace.
5. Add into the `MobileDevice` class:
   i. A method named `needCharging()` as explained in the class interface table.
   ii. A method named `recharge()` as explained in the class interface table.

Check your answer by running `Tester3` and your output should be as follows:

```
Basic phone details
Manufacturer: Nokia
Model: 150DualSim
Battery Status: 10
Station: Mix.fm
Frequency: 94.5
Recharge completed: 100%

Basic phone details
Manufacturer: Nokia
Model: 3310
Battery Status: 30
Station: Ikim.fm
Frequency: 91.5

Basic phone details
Manufacturer: Samsung
Model: GuruFM
Battery Status: 40
Station: THR.fm
Frequency: 99.3

Basic phone details
Manufacturer: Samsung
Model: Rugby3
Battery Status: 90
Station: Hitz.fm
Frequency: 92.9

Basic phone details
Manufacturer: SonyEriccson
Model: Walkman
Battery Status: 5
Station: TraXX.fm
Frequency: 90.3
Recharge completed: 100%

Smart phone details
Manufacturer: Samsung
Model: S8
Battery Status: 60
Audio playing: Assalamualaikum
Video playing: Robocop

Smart phone details
Manufacturer: Apple
Model: iPhone7
Battery Status: 10
Audio playing: Hello
Video playing: Terminator
Recharge completed: 100%

Smart phone details
Manufacturer: Huawei
Model: P8Lite
Battery Status: 10
Audio playing: Crush
Video playing: iRobot
Recharge completed: 100%

Smart phone details
```

```
Manufacturer: Oppo
Model: R9s
Battery Status: 80
Audio playing: Isabella
Video playing: Tunnel
```

*Proceed to Stage 3 only after you have completed Stage 2 without errors.

Stage 4:

1. Create a Java project named `Lab4AStage4`.
2. Copy file `Tester4.java` into your project.
3. Copy files `RadioTuner.java`, `AudioPlayer.java`, `VideoPlayer.java`, `MobileDevice.java`, `BasicPhone.java` and `SmartPhone.java` in Stage 3 into your Stage 4 Java project.
4. Create a text file named `mobilephone2.dat` and copy the content from the files provided. (Right-click on project->new File) /Or copy the file from previous project workspace.
5. Add into the `BasicPhone` class :
   i. A void method named `setRadioSetting` to change/tune the radio to new station and frequency input by user.
6. Add into the `SmartPhone` class :
   i. A void method named `setCurrentAudio()` to change the audio clip to new audio clip input by user.
   ii. A void method named `setCurrentVideo()` to change the video clip to new video clip input by user.

Check your answer by running `Tester4.` Followings are the output:

```
Basic phone details
Manufacturer: Nokia
Model: 150DualSim
Battery Status: 10
Station: Mix.fm
Frequency: 94.5
New station : Hot.fm
New frequency : 97.6
Basic phone details
Manufacturer: Nokia
Model: 150DualSim
Battery Status: 10
Station: Hot.fm
Frequency: 97.6

Basic phone details
Manufacturer: Nokia
Model: 3310
Battery Status: 30
Station: Ikim.fm
Frequency: 91.5
New station : Mix.fm
New frequency : 94.5
Basic phone details
Manufacturer: Nokia
```

```
Model: 3310
Battery Status: 30
Station: Mix.fm
Frequency: 94.5

Basic phone details
Manufacturer: Samsung
Model: GuruFM
Battery Status: 40
Station: THR.fm
Frequency: 99.3
New station : Ikim.fm
New frequency : 91.5
Basic phone details
Manufacturer: Samsung
Model: GuruFM
Battery Status: 40
Station: Ikim.fm
Frequency: 91.5

Basic phone details
Manufacturer: Samsung
Model: Rugby3
Battery Status: 90
Station: Hitz.fm
Frequency: 92.9
New station : Ikim.fm
New frequency : 91.5
Basic phone details
Manufacturer: Samsung
Model: Rugby3
Battery Status: 90
Station: Ikim.fm
Frequency: 91.5

Basic phone details
Manufacturer: SonyEriccson
Model: Walkman
Battery Status: 5
Station: TraXX.fm
Frequency: 90.3
New station : Mix.fm
New frequency : 94.5
Basic phone details
Manufacturer: SonyEriccson
Model: Walkman
Battery Status: 5
Station: Mix.fm
Frequency: 94.5

Smart phone details
Manufacturer: Samsung
Model: S8
Battery Status: 60
Audio playing: Assalamualaikum
```

```
Video playing: Robocop
New audioclip : Setia
New videoclip : Boboiboy
Smart phone details
Manufacturer: Samsung
Model: S8
Battery Status: 60
Audio playing: Setia
Video playing: Boboiboy

Smart phone details
Manufacturer: Apple
Model: iPhone7
Battery Status: 10
Audio playing: Hello
Video playing: Terminator
New audioclip : Menang
New videoclip : Allegiant
Smart phone details
Manufacturer: Apple
Model: iPhone7
Battery Status: 10
Audio playing: Menang
Video playing: Allegiant

Smart phone details
Manufacturer: Huawei
Model: P8Lite
Battery Status: 10
Audio playing: Crush
Video playing: iRobot
New audioclip : Kumohon
New videoclip : Divergent
Smart phone details
Manufacturer: Huawei
Model: P8Lite
Battery Status: 10
Audio playing: Kumohon
Video playing: Divergent

Smart phone details
Manufacturer: Oppo
Model: R9s
Battery Status: 80
Audio playing: Isabella
Video playing: Tunnel
New audioclip : Lullabies
New videoclip : Starwars
Smart phone details
Manufacturer: Oppo
Model: R9s
Battery Status: 80
Audio playing: Lullabies
Video playing: Starwars
```