Icon

Description automatically generated

Name – Devanshu Dave

Date - 10/11/2022

MODE 4201-05

Title - Interfaces in Dart , Basic Widgets in Flutter.

Table Of Contents.

Contents

[Part 1 - Interfaces in Dart Programming Language. 3](#_Toc119103338)

[Dart Programming Language. 3](#_Toc119103339)

[Interfaces. 3](#_Toc119103340)

[Part 2 – Basic Widgets in Flutter. 7](#_Toc119103341)

[References 9](#_Toc119103342)

# Part 1 - Interfaces in Dart Programming Language.

## Dart Programming Language.

* This is the programming language used for mobile development. It is free to use as it is an open-source language developed by Google. It was also approved by ECMA as a standard. It is a multi-platform language meaning with the help of this programing language we can develop both iOS apps and android apps. It is an object-oriented programming language meaning one should have good knowledge of object-oriented programming to use dart programming language.

## Interfaces.

* The interface is a part of object-oriented programming. What are interfaces? Interfaces are simply the blueprint of the class. In Dart, there is no interface keyword or anything like that as we have in other programming languages. But to use the interface in the Dart Programming language we use the ‘implements’ keyword to use interface. In a dart programming language, if we use the ‘implements’ keyword it means that we are telling to forcefully use the methods inside that class in the new class. Let’s see with the example.

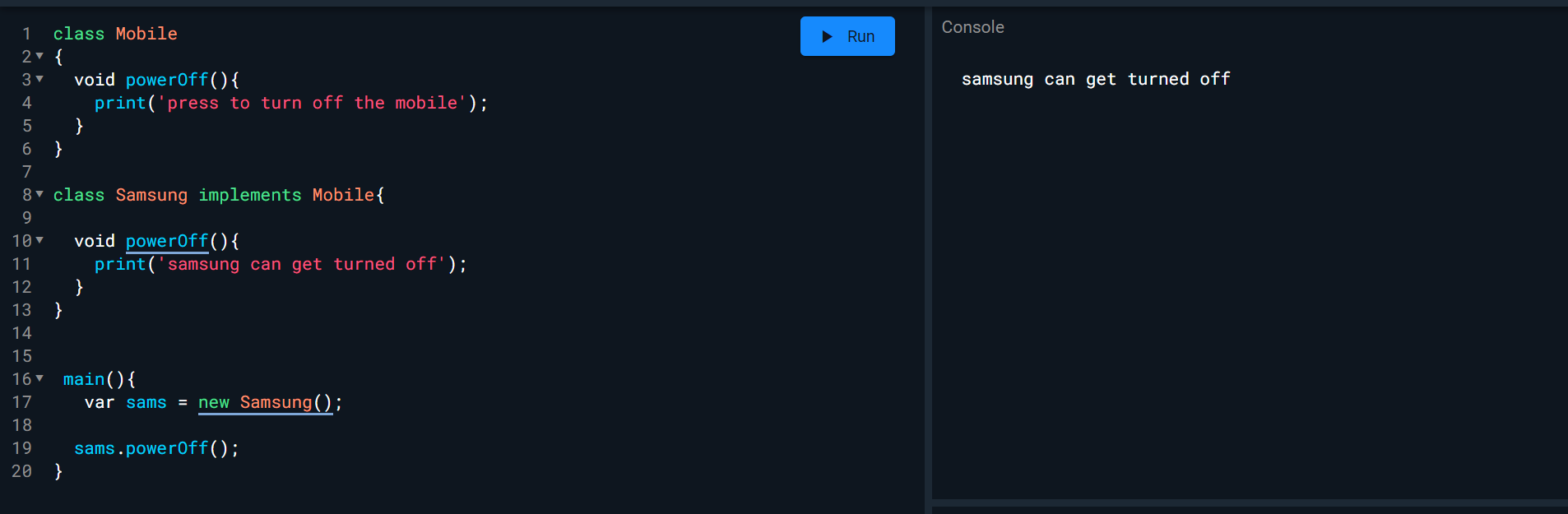
* As shown in the image below, we created one class named Mobile which is not an abstract class but a normal class which can have method definition plus method implementation. Inside that class, we created a method named power off with a simple print statement. In the second class, as shown in the image below, we have used the implements keyword which means we are using the first class named mobile as an interface and telling to implement all the methods in our interface class to the normal class named Samsung forcefully. As in the below example, we didn’t use that the compiler is throwing an error as shown in the second image. That we have to implement the method written in our interface. Therefore, using ‘implements’ (for interface) tells the class to forcefully implement the methods used in the first class.

Text

Description automatically generatedText

Description automatically generated

* To solve the above error, we simply implement (override) the method in our Samsung class. In the example shown below, describes that implementation was done on Samsung class also as dart interface forces us to implement the method. Main method below creates the object of the class and runs the method we wrote in Samsung class.



We should write @override notation above the powerOff method. As Dart Pad is also giving same suggestion. New keyword on line 17 is also not mandatory which can be removed which was also suggested by Dart Pad compiler.

* As we saw that we can inherit other classes with the help of implements which make the other class an interface. If we compare this topic with inheritance in oops where is use extends keyword to inherit the properties of the other class where extends (inheritance) allows only one class to inherit and doesn’t allow more than one. But interfaces give us the advantage to inherit more than one class. For example, if we have a class which inherits from 3 classes that’s only possible through interfaces using the implements keyword. Let’s understand with the help of the example I wrote on Dart pad.A screenshot of a computer

  Description automatically generated with medium confidence
* Above in the image, we have three classes named as Mobile, Tablet and Watch with respective methods. We have other class named as Samsung. So here as we can see we can inherit more than one classes separated by comas with the help of implements keyword. So, interface, gives us an advantage to inherit more than one class.
* We can also use abstract class and make that interface scenario. Let’s discuss in depth as shown in below screenshot.

A screenshot of a computer

Description automatically generated with medium confidence

* It is useful as an abstraction in a Dart Programming language. As above, we created an abstract interface class meaning we can only define the class which made a forceful implementation in Samsung class because we are using implements keyword to inherit the Mobile abstract class. We must use override notation above the methods in this scenarios.
* Interfaces in Dart Programming language is useful as we seen how we can use and why to use in above paragraphs.

# Part 2 – Basic Widgets in Flutter.

* Flutter is the cross-platform meaning we can use the same code/tech for android which will also work in IOS. Before, we have to code or do it differently, but Flutter gives us the advantage to use it as a cross-platform and is free to use. In flutter, we have widgets which are simply the things that will be viewed by the user on the application. For example, text. Text is a widget type which shows the user the description in a text view on the application. Every element or the things which are seen in the application view is a widget in Flutter.
* In flutter, there are some basic but strong features which help to make the application view better. They are text, row, Column, Container, Scaffold, Image, Placeholder, Elevated Button, Icon, and Flutter Logo. These are the basic widgets according to flutter's Official documents.
* First Basic Widget is Text. It is itself a class meaning we have to declare that as a Text class. This widget simply shows the string of desired text on the application. If we want “Hello Mobile Development” on the screen we have to use the text widget which will show this string on the application screen. It will not contain any design as depending on where we place it, will be seen accordingly. For example. It could be seen in two different lines meaning mode on one and hello on one. Or only Hello can be seen, and Mode is ignored depending on where we place it on our layout. But this is one of the basic widgets used in a flutter.
* The second is the Row Widget. This is the widget which allows aligning the element in a row or column. These are the widgets which are commonly used in the development process to align the elements. This widget is used when the developer wants to show the element’s children horizontally.
* The third Basic Widget is Column. This is also a common widget used by developers to set the children in a vertical row. For example, if we want to align the box vertically we use this column widget.
* The fourth Basic Widget is the Image. If on the application screen we want to show any photos then Image Widget is the best option to view it on the screen. There are various properties meaning constructors which help to source the image. For example, if any developer needs to source the image from any website or has a like then simply developer can use -:

Image.Network(URL);

If any developer wants to use the image from the file then the easy way in flutter is to use this property which is -:

Image.file(file location on local machine);

These are some of the properties which can be useful for placing an image on the application. More than that are Image.asset, Image.new and Image.memory.

* The fifth Basic widget is the App bar. On the Websites, we see the navigation bars on the top of the website with icons, menu actions, titles, etc. Similarly to that, in Flutter we have an App Bar widget. As discussed above. This widget gives the ability to add titles, icons, buttons, and a menu bar at the top of the app that the user can easily be able to see. We can also change the theme of the app bar by setting different colours. We can easily code or set the widget defining/setting the properties discussed previously.

* Another Basic Widget is the container. This is the common and basic widget of the flutter. This works like when we use a container it specifies the thing will be surrounded with padding (spaces around). It is used to position the widget with the appropriate margin and padding surrounding it.

* Another Basic Widget that flutters offer is Icons. This widget is used to show the icon on the screen of the application. As a part of coding, we use Icons.nameOfTheAppropriateIcon to use the icons and display them on the screen.

* Flutter also offers another basic widget which is PlaceHolder. It’s a basic widget which generally use to draw the boxes around the elements which specify that the application is not ready to be deployed or for any further use.

* Sometimes we as a developer needs the flutter logo to show it on the screen for whatever reasons. One can be to give the credit to the flutter but there can be many reasons. To answer this problem, flutter gives the widget which is called flutter logo which is also defined in the code as FlutterLogo.

* ElevatedButton is also one of the basic features that flutter provides. This is used when the developer wants to show some slight evaluation when the button is pressed.
* So, these are the basic Widgets that flutter provides to the developers to use towards the applications. As we have gone through each basic widget with some example code blocks to understand the widget.

# References

*Language samples*. Dart. (n.d.). Retrieved November 11, 2022, from <https://dart.dev/samples>

*Basic widgets*. Flutter. (n.d.). Retrieved November 11, 2022, from <https://docs.flutter.dev/development/ui/widgets/basics>

*Dart programming - interfaces*. Tutorials Point. (n.d.). Retrieved November 11, 2022, from <https://www.tutorialspoint.com/dart_programming/dart_programming_interfaces.htm#:~:text=Dart%20does%20not%20have%20a,functions%20of%20the%20implemented%20interface>.