

D. Zosimovych, Bachelor student
A. Morozov, PhD in Engr., As. Prof., research advisor
S. Kobzar, Senior lecturer, language advisor
Zhytomyr State Technological University

HEAD SDART¹ FOR BEGINNING MOBILE DEVELOPERS

Mobile development is a very promising direction in programming. People use smartphones for different ordinary affairs. And somebody is even really addicted to their pocket gadgets. That is why most IT solutions should be primarily targeted to be mobile friendly.

What is the language you love coding on? Java? C-Sharp? Maybe Python? Dart is about a mix of C# code style, Java's streams and embedded big arithmetic. Object oriented language with great flexibility that is enriched with powerful frameworks is the one worth trying. You have an opportunity to write client code using AngularDart (it is still processed into JavaScript to be executed by modern browsers), server code on DartVM or mobile one using Flutter [1].

We have been developing an absolutely new app using Flutter. It is a relatively new cross-platform framework developed by Google as a response to React Native [2]. Its Beta version has been released recently. That means it is still being developed furtherly and it will be enriched with other features.

There are a lot of pros of native mobile development on Dart. One major advantage is that you have a single code base for different platforms. For now, Flutter supports building apps for only 2 major mobile OS: Android and IOS. However, these two cover almost 95% of world's mobile users [3]. Another great feature is possibility to write platform specific code that can interact with your Dart one. Apart from that, Dart has a bunch of "out of box" functional solutions. There are many libraries distributed with language SDK. Networking, storage and media support are among them. Dart is described to provide features like `async/await`, generator functions, string interpolation and much more [1]. And debugging applications on devices both emulated and real makes this process more exciting because hot and hard app reloads occur within seconds.

Unlike usual Android app written on Java, Flutter has a single entry point. All further operations start from the "main" function. One major disadvantage is the reflection absence due to the fact that Dart is processed into the native code and the reflection will cause dozens of code to be generated.

Turning to the interface scaffolding you have the possibility to build adaptive widgets that are based on the native components or created from scratch. You can either create immutable ones, that save their constant state during the view lifecycle, or simply draw them on Canvas. There are also "stateful" widgets that are re-rendered each time the data changes. Besides, placeholder for the asynchronous data can be displayed depending on its execution state. Flutter also supports building flex box layouts, so that widget positioning and sizing are much easier to be built up by the developer.

In conclusion, we want to notice that our work showed that Dart is neither Kotlin, nor Swift, though, it is different and the possibility to code on it must not be skipped.

¹ Head sDart stands for "head start with Dart"

REFERENCES

1. Dart language, [Электронный ресурс] – Режим доступа:
<https://www.dartlang.org/>
2. Matwrites, [Электронный ресурс] – Режим доступа:
<https://matwrites.com/flutter-react-native-competitor/>
3. GlobalStats, statcounter, [Электронный ресурс] – Режим доступа:
<https://gs.statcounter.com/os-market-share/mobile/worldwide/>