

Lab 1. LÀM QUEN VỚI FLUTTER

A. MỤC TIÊU:

- ✓ Giúp sinh viên làm quen với lập trình flutter.
- ✓ Sinh viên có khả năng tự viết chương trình mở rộng dựa trên các bài thực hành có hướng dẫn cụ thể.

B. CÔNG TÁC CHUẨN BỊ

- ✓ Tải và cài đặt Flutter trên IDE Android Studio.
- ✓ Tải và cài đặt ngôn ngữ Dart.

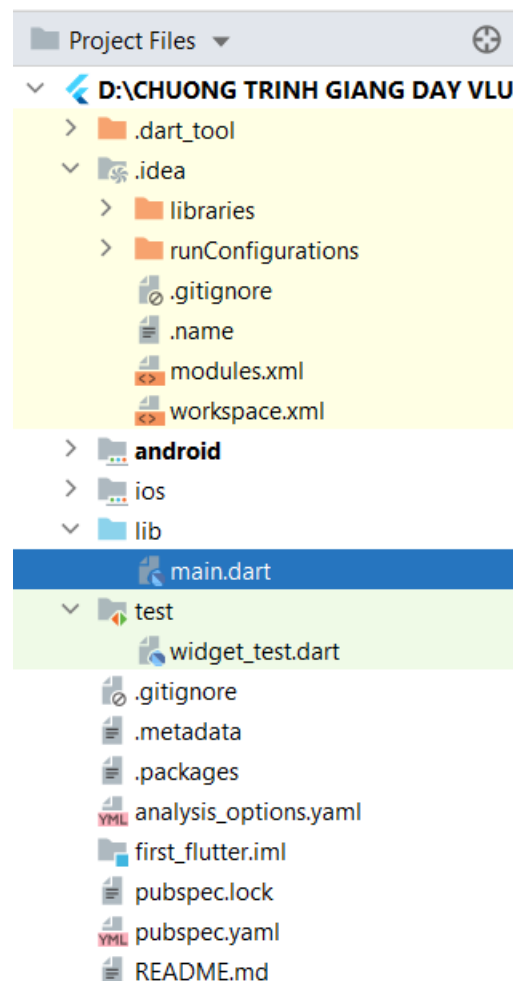
C. MÔI TRƯỜNG LẬP TRÌNH

Có nhiều môi trường và công cụ để lập trình cho một hoặc nhiều ngôn ngữ. Tuy nhiên để dễ dàng làm quen, sinh viên có thể lập trình với IDE đã được cài đặt trên máy tính.

D. NỘI DUNG THỰC HÀNH

❖ YÊU CẦU

- ✓ **Tạo mới một Project Flutter.**
- ✓ **Cấu trúc của Project sau khi được tạo thành công**



Ý nghĩa các thành phần trong Project:

- ✓ **android:** Thư mục code sinh tự động cho ứng dụng Android
- ✓ **ios:** Thư mục code sinh tự động cho ứng dụng iOS
- ✓ **lib:** Main folder chứa Dart code được viết khi sử dụng flutter framework
- ✓ **lib/main.dart:** File đầu tiên là điểm khởi đầu của ứng dụng Flutter application
- ✓ **test:** Folder chứa Dart code để test flutter application
- ✓ **test/widget_test.dart:** Sample code
- ✓ **gitignore:** Git version control file - File này chứa cấu hình cho project git
- ✓ **metadata:** sinh tự động bởi flutter tools
- ✓ **packages:** sinh tự động để theo dõi flutter packages
- ✓ **iml:** project file của Android studio
- ✓ **pubspec.yaml:** Được sử dụng Pub, Flutter package manager
- ✓ **pubspec.lock:** Sinh tự động bởi Flutter package manager, Pub
- ✓ **README.md:** Project description được viết theo cấu trúc Markdown

Mặc định Android Studio đã tạo sẵn code ở lib/main.dart, nên xóa nội dung trong file đi và viết code lại.

Bài tập 1: Viết chương trình xuất ra một chuỗi “ Đây là chương trình Flutter đầu tiên của tôi!”. Không sử dụng Widget.

Hướng dẫn:

```
import 'package:flutter/material.dart';

void main(){
  runApp(MaterialApp(
    home: Text("Đây là chương trình Flutter đầu tiên của tôi!"),
  )); // MaterialApp
}
```

Run chương trình để xem kết quả của chương trình không sử dụng Widget.

Bài tập 2: Tương tự bài tập 1, nhưng sử dụng Widget là một Scaffold cho chương trình.

Hướng dẫn:

```
import 'package:flutter/material.dart';

void main(){
  runApp(MaterialApp(
    home: Scaffold(
      body: Text("Đây là chương trình Flutter đầu tiên của tôi!")
    ), // Scaffold
  )); // MaterialApp
}
```

Run chương trình xem kết quả.

Bài tập 3. Tiếp tục cải tiến chương trình từ bài tập 2 và thiết lập khu vực hiển thị dữ liệu an toàn.

Hướng dẫn:

```
import 'package:flutter/material.dart';

void main(){
  runApp(MaterialApp(
    home: SafeArea(
      child: Scaffold(
        body: Text("Đây là chương trình Flutter đầu tiên của tôi!")
      ), // Scaffold
    ), // SafeArea
  )); // MaterialApp
}
```

Run chương trình xem kết quả.

Bài tập 4. Viết chương trình xuất chuỗi "Hello Flutter, this is my first Application!" lên màn hình di động. Yêu cầu xây dựng class riêng để quản lý chương trình.

Hướng dẫn:

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MyApp());
5 }
6
```

```

7   class MyApp extends StatelessWidget {
8     const MyApp({Key? key}) : super(key: key);
9     // This widget is the root of your application.
10    @override
11    Widget build(BuildContext context) {
12      return MaterialApp(
13        title: 'Flutter Demo',
14        theme: ThemeData(
15          primarySwatch: Colors.blue,
16        ), // ThemeData
17        home: const MyHomePage(title: 'My Home Page'),
18      ); // MaterialApp
19    }
20  }
21
22  class MyHomePage extends StatelessWidget {
23    const MyHomePage({Key? key, required this.title}) : super(key: key);
24    final String title;
25    Widget build(BuildContext context){
26      return Scaffold(
27        appBar: AppBar(
28          title: Text(this.title),
29        ), // AppBar
30        body: Center(
31          child: Text("Hello Flutter, this is my first Application!")
32        ), // Center
33      ); // Scaffold
34    }
35  }

```

Run chương trình xem kết quả.

-----Hết-----