

# MOBİL PROGRAMLAMA PROJESİ KODLARI VE ÇIKTILARI

## PROVİDER

```
Provider.dart X
lib > Provider.dart > ...
1
2 import 'package:flutter/material.dart';
3 import 'package:provider/provider.dart';
4
5 void main() {
6   runApp(
7     // Uygulama açılırken Counter modelini sağlayan yapı
8     ChangeNotifierProvider(
9       create: (_) => Counter(),
10      child: const MyApp(),
11    ), // ChangeNotifierProvider
12  );
13 }
14
15 class Counter extends ChangeNotifier {
16   int value = 0;
17   void inc() {
18     value++;
19     notifyListeners(); // Dinleyen widget'lara haber verir
20   }
21 }
22
23 class MyApp extends StatelessWidget {
24   const MyApp({super.key});
25
26   @override
27   Widget build(BuildContext context) {
28     // Counter sınıfındaki güncel değeri dinler
29     final counter = context.watch<Counter>();
30
31     return MaterialApp(
32       home: Scaffold(
33         body: Center(child: Text("${counter.value}", style: TextStyle(fontSize: 40))),
34         // Butona basıldığında sayacı artırır
35         floatingActionButton: FloatingActionButton(
36           onPressed: () => context.read<Counter>().inc(),
37           child: Icon(Icons.add),
38         ), // FloatingActionButton
39       ), // Scaffold
40     ); // MaterialApp
41   }
42 }
43
```

14



# RİVERPOD

```
Riverpod.dart X
lib > Riverpod.dart > ...
1
2 import 'package:flutter/material.dart';
3 import 'package:flutter_riverpod/flutter_riverpod.dart';
4
5 // PROVIDER
6 final counterProvider = StateProvider<int>((ref) => 0);
7
8 Run | Debug | Profile
9 void main() {
10   runApp(
11     // Riverpod'un çalışması için ProviderScope gerekir
12     const ProviderScope(child: MyApp()),
13   );
14 }
15 class MyApp extends ConsumerWidget {
16   const MyApp({super.key});
17
18   @override
19   Widget build(BuildContext context, WidgetRef ref) {
20     final count = ref.watch(counterProvider);
21
22     return MaterialApp(
23       home: Scaffold(
24         body: Center(
25           child: Text("$count", style: const TextStyle(fontSize: 40)),
26         ), // Center
27         // Butona basınca değeri 1 artırır
```

```
26       ), // Center
27       // Butona basınca değeri 1 artırır
28       floatingActionButton: FloatingActionButton(
29         onPressed: () => ref.read(counterProvider.notifier).state++,
30         child: const Icon(Icons.add),
31       ), // FloatingActionButton
32     ), // Scaffold
33   ); // MaterialApp
34 }
35 }
36
```

## BLOC

```
BloC.dart X
lib > BloC.dart > ...
1
2 import 'package:flutter/material.dart';
3 import 'package:flutter_bloc/flutter_bloc.dart';
4
5 // CUBIT
6 // BLoC yapısı = Event + State dönüşümü yapan sınıf
7 class CounterCubit extends Cubit<int> {
8   CounterCubit() : super(0);
9
10   void inc() => emit(state + 1);
11 }
12
13 Run | Debug | Profile
14 void main() {
15   runApp(
16     // CounterBloc'u uygulamaya sağlayan yapı
17     BlocProvider(
18       create: (_) => CounterCubit(),
19       child: const MyApp(),
20     ), // BlocProvider
21   );
22 }
23
24 class MyApp extends StatelessWidget {
25   const MyApp({super.key});
26 }
```

```
22
23 class MyApp extends StatelessWidget {
24   const MyApp({super.key});
25
26   @override
27   Widget build(BuildContext context) {
28     return MaterialApp(
29       home: Scaffold(
30         body: Center(
31           child: BlocBuilder<CounterCubit, int>(
32             builder: (_, count) => Text("$count", style: const TextStyle(fontSize: 40)),
33           ), // BlocBuilder
34         ), // Center
35         // Butona basınca IncrementEvent gönderir
36         floatingActionButton: FloatingActionButton(
37           onPressed: () => context.read<CounterCubit>().inc(),
38           child: const Icon(Icons.add),
39         ), // FloatingActionButton
40       ), // Scaffold
41     ); // MaterialApp
42   }
43 }
```