Implementing 3 times

- ChangeNotifierProvider + ChangeNotifierProxyProvider ⇦ **DONE**
- ChangeNotifierProvider + ProxyProvider ⇦ **DONE**

- StateNotifierProvider
  - A package created by Remi, the author of the provider package.
  - No need to use ProxyProvider
  - Widely used in Riverpod, another state management solution by Remi.
  - If you are going to use Riverpod in the future, knowing the StateNotifierProvider is very helpful to understand.
  - A detailed description of StateNotifierProvider will be provided in the corresponding Chapter.

State pattern

- Creating state class
  - class TodoListState {}

- Creating ChangeNotifier and initializing state
  - Class TodoList with ChangeNotifier {
    TodoListState state = TodoListState(todos: []);
    }

- Changing state and notifying that change to the listeners
  - state = state.copyWith(...); notifyListners();

State

```
class TodoSearchState extends Equatable {
  final String searchTerm;
  const TodoSearchState({required this.searchTerm});

  factory TodoSearchState.initial() {
    return const TodoSearchState(searchTerm: '');
  }

  @override
  List<Object> get props => [searchTerm];

  @override
  bool get stringify => true;

  TodoSearchState copyWith({String? searchTerm}) {
    return TodoSearchState(searchTerm: searchTerm ?? this.searchTerm);
  }
}
```

ChangeNotifier

```
class TodoSearch with ChangeNotifier {
  TodoSearchState _state = TodoSearchState.initial();
  TodoSearchState get state => _state;

  void setSearchTerm(String newSearchTerm) {
    _state = _state.copyWith(searchTerm: newSearchTerm);
    notifyListeners();
  }
}
```

Access

```
ChangeNotifierProvider<TodoSearch>(create: (context) => TodoSearch()),

Provider.of<TodoSearch>(context, listen: false).setSearchTerm('abc');

        - OR -

context.read<TodoSearch>().setSearchTerm('abc');


Provider.of<TodoSearch>(context).state

        - OR -

context.watch<TodoSearch>().state
```

ProxyProvider

```
ChangeNotifierProxyProvider3<TodoFilter, TodoSearch, TodoList, FilteredTodos>(
            create: (context) => FilteredTodos(),
            update: (
              BuildContext context,
              TodoFilter todoFilter,
              TodoSearch todoSearch,
              TodoList todoList,
              FilteredTodos? filteredTodos,
            ) =>
                filteredTodos!..update(todoFilter, todoSearch, todoList),
          ),
```