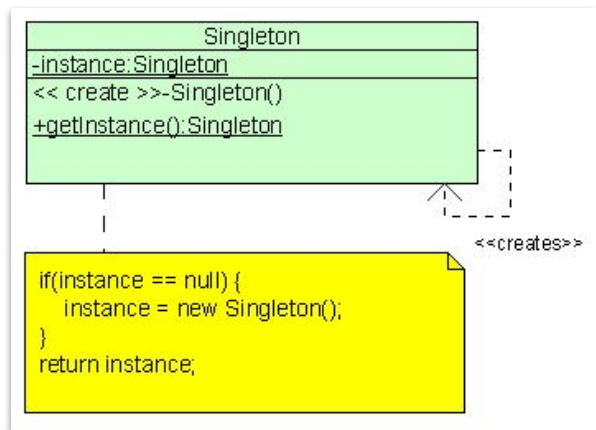


Design Patterns - Singleton Pattern

Basic Implementation:

There are a few ways in which we could implement the Singleton pattern in Dart and Flutter.



Classic Approach

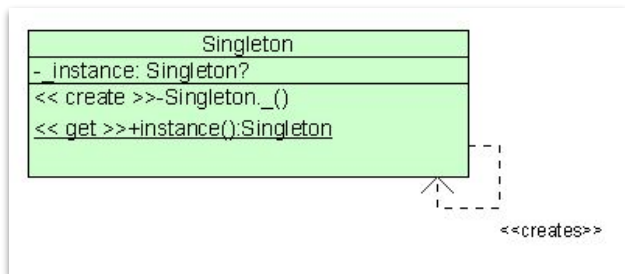
Design Patterns - Singleton Pattern

Design Patterns - Singleton Pattern

Basic Implementation:

There are a few ways in which we could implement the Singleton pattern in Dart and Flutter.

Classic Approach in Dart



```
Singleton s = Singleton.instance;
```

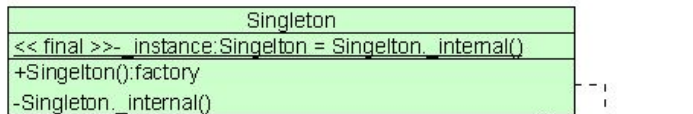
```
class Singleton {  
  /// eager instantiation  
  static final Singleton _instance = Singleton._internal();  
  
  ///  
  /// generative constructor called internally  
  Singleton._internal() {  
    // simple test for creation  
    debugPrint('Creating an instance of Singleton');  
  }  
  
  ///  
  /// factory constructor, should return same instance  
  factory Singleton() {  
    return _instance;  
  }  
}
```

Design Patterns - Singleton Pattern

Design Patterns - Singleton Pattern

Basic Implementation:

There are a few ways in which we could implement the Singleton pattern in Dart and Flutter.



factory keyword in Dart

```
Singleton s = Singleton();
```

```
class Singleton {  
  /// eager instantiation  
  static final Singleton _instance = Singleton._internal();  
  
  ///  
  /// generative constructor called internally  
  Singleton._internal() {  
    // simple test for creation  
    debugPrint('Creating an instance of Singleton');  
  }  
  
  ///  
  /// factory constructor, should return same instance  
  factory Singleton() {  
    return _instance;  
  }  
}
```

Design Patterns - Singleton Pattern

Design Patterns - Singleton Pattern

Usage Example:

```
Logger? _logger;  
final _dateFormatter = DateFormat('H:m:s.S');  
const appName = 'my_app';  
  
Logger.root.level = Level.ALL;  
Logger.root.onRecord.listen((LogRecord rec) {  
    print('${_dateFormatter.format(rec.time)}: ${rec.message}');  
});  
_logger = Logger(appName);
```

```
_logger?.info(message, error, stackTrace);
```

Design Patterns - Singleton Pattern

Design Patterns - Singleton Pattern

Usage Example:

```
final logger = DebugLogger();  
logger.log('some message');
```

```
class DebugLogger {  
    static DebugLogger? _instance;  
    static Logger? _logger;  
    static final _dateFormatter = DateFormat('H:m:s.S');  
    static const appName = 'my_app';  
  
    DebugLogger._internal() {  
        Logger.root.level = Level.ALL;  
        Logger.root.onRecord.listen(_recordHandler);  
  
        _logger = Logger(appName);  
        _instance = this;  
    }  
  
    factory DebugLogger() => _instance ?? DebugLogger._internal();  
  
    void _recordHandler(LogRecord rec) {  
        print('${_dateFormatter.format(rec.time)}: ${rec.message}');  
    }  
  
    void log(message, [Object? error, StackTrace? stackTrace]) =>  
        _logger?.info(message, error, stackTrace);  
}
```

Design Patterns - Singleton Pattern