آشنایی با

Class Modifiers

در زبان دارت

Class Modifier چیه؟

Class Modifier نحوه استفاده از کلاس رو مدیریت می‌کنه.

کلمه‌های کلیدی Modifier قبل از اعلان کلاس قرار می‌گیرن.

Class Modifier

Raheleh Shahpasand

انواع Class Modifier

- abstract
- base
- final
- interface
- sealed
- mixin

Raheleh Shahpasand

abstract

وقتی نیازی به پیاده‌سازی concrete کلاس نباشه استفاده می‌شه.

نمی‌تونیم از کلاس abstract نمونه بسازیم.

می‌تونیم با استفاده ازش یک subtype جدید تعریف کنیم.

می‌تونیم واسطی که کلاس abstract ایجاد کرده رو پیاده کنیم.

a.dart

```dart
abstract class Vehicle {
  void moveForward(int meters);
}
```

b.dart

```dart
import 'a.dart';

// Error: Can't be constructed.
Vehicle myVehicle = Vehicle();

// Can be extended.
class Car extends Vehicle {
  int passengers = 4;
  // ...
}

// Can be implemented.
class MockVehicle implements Vehicle {
  @override
  void moveForward(int meters){
    //...
  }
}
```

Raheleh Shahpasand

base

از این نوع کلاس فقط میشه ارث‌بری کرد.
این نوع کلاس، اجازه نمیده پیاده‌سازی خارج از لایبرری خودش انجام بشه.

می‌تونیم از کلاس base نمونه بسازیم.

می‌تونیم با استفاده از کلاس base یک subtype جدید بسازیم که باید base، final یا sealed باشه.

نمی‌تونیم کلاس base رو پیاده کنیم.

a.dart

```dart
base class Vehicle {
  void moveForward(int meters) {
    // ...
  }
}
```

b.dart

```dart
import 'a.dart';

// Can be constructed.
Vehicle myVehicle = Vehicle();

// Can be extended.
base class Car extends Vehicle {
  int passengers = 4;
  // ...
}

// ERROR: Can't be implemented.
base class MockVehicle implements Vehicle {
  @override
  void moveForward() {
    // ...
  }
}
```

Raheleh Shahpasand

interface

برای تعریف واسط استفاده می‌شه. پیاده‌سازی می‌تونه در لایبرری غیر از لایبرری فعلی کلاس انجام بشه اما extend ممکن نیست.

می‌تونیم نمونه بسازیم.

نمی‌تونیم ارث‌بری کنیم.

می‌تونیم پیاده‌سازی کنیم.

a.dart

```dart
interface class Vehicle {
  void moveForward(int meters) {
    // ...
  }
}
```

b.dart

```dart
import 'a.dart';

// Can be constructed.
Vehicle myVehicle = Vehicle();

// ERROR: Can't be inherited.
class Car extends Vehicle {
  int passengers = 4;
  // ...
}

// Can be implemented.
class MockVehicle implements Vehicle {
  @override
  void moveForward(int meters) {
    // ...
  }
}
```

Raheleh Shahpasand

final

از کلاسی که final تعریف شده، خارج از لایبرری جاری نمی‌تونیم subtype ایجاد کنیم.

می‌تونیم نمونه بسازیم.

نمی‌تونیم ارث‌بری کنیم.

نمی‌تونیم پیاده‌سازی کنیم.

a.dart

```dart
final class Vehicle {
  void moveForward(int meters) {
    // ...
  }
}
```

b.dart

```dart
import 'a.dart';

// Can be constructed.
Vehicle myVehicle = Vehicle();

// ERROR: Can't be inherited.
class Car extends Vehicle {
  int passengers = 4;
  // ...
}

class MockVehicle implements Vehicle {
  // ERROR: Can't be implemented.
  @override
  void moveForward(int meters) {
    // ...
  }
}
```

Raheleh Shahpasand

sealed

می‌تونیم یک مجموعه از enumerable از subtypeها رو با استفاده از کلاس نوع sealed ایجاد کنیم. این کلاس‌ها به صورت ضمنی abstract هستن.

از کلاس نوع sealed نمی‌تونیم نمونه بسازیم.

از subtype می‌تونیم نمونه بسازیم.

خطا داریم چون یک subtype رو جا انداختیم.

```dart
// a.dart
sealed class Vehicle {}

class Car extends Vehicle {}

class Truck implements Vehicle {}

class Bicycle extends Vehicle {}

// ERROR: Can't be instantiated.
Vehicle myVehicle = Vehicle();

// Subclasses can be instantiated.
Vehicle myCar = Car();

String getVehicleSound(Vehicle vehicle) {
  // ERROR: The switch is missing the Bicycle
  // subtype or a default case.
  return switch (vehicle) {
    Car() => 'vroom',
    Truck() => 'VROOOOMM',
  };
}
```

Raheleh Shahpasand

mixin

کلمه کلیدی mixin برای تعریف mixin استفاده می‌شه. کلمه کلیدی class هم برای تعریف کلاس. یک mixin class هم کلاسه هم mixin.
این نوع کلاس محدودیت‌های class و mixin رو همزمان داره پس:

نمی‌تونیم از extends یا with برای mixin استفاده کنیم پس برای mixim class هم نمی‌تونیم.

نمی‌تونیم از on برای کلاس استفاده کنیم پس برای mixim class هم نمی‌تونیم.

main.dart

```dart
mixin class Musician {
  // ...
}

//Use Musucian as a mixin
class Novice with Musician{
  // ...
}

//Use Musucian as a class
class Novice extends Musician{
  // ...
}
```

Raheleh Shahpasand

No modifier

a.dart

```dart
class Vehicle {
  void moveForward(int meters) {
    // ...
  }
}
```

یک حالت دیگه هم داریم که بیشتر از بقیه برامون آشناست:
از modifier استفاده نکنیم.
با این کار به همه لایبرری‌هایی که از کلاس ما استفاده می‌کنن مجوز بدون محدودیت برای نمونه‌سازی، پیاده‌سازی یا ارث‌بری از کلاس‌مون رو دادیم.

حالا که با Class Modifier در زبان دارت آشنا شدیم، بهتره به کلاس‌هایی که می‌نویسیم بیشتر دقت کنیم.
می‌تونیم دسترسی محدودتری برای کلاس در نظر بگیریم؟
چه دسترسی برای چه سناریویی مناسبه؟