



Modul II. Lesson 7

Constructor II

Nasibali Abdiyev. Flutter Development

Repeat the previous lesson

- Nullable list
- Nullable element
- Mutable and immutable list
- List mapping
- List filtering
- Other methods of List
- Iterable
- Other Map methods

Plan

- Constructor
- Constant constructor
- Named constructor
- Initializer list
- Redirecting constructor
- Factory constructor

Constructors

- Konstruktorlar - bu sinf instance larini hosil qiladigan yani tuzadigan method lardir. Ya'ni, konstruktor lar yangi ob'ektlarni qurishadi. Konstruktorlar class bilan bir xil nomga ega.
- Konstruktor – bu class dan object yaratayotganda birinchi bo'lib ishga tushadigan method hisoblanadi. Bu sizga class dagi boshqa methodlar ishlashidan oldin biron ish bajarishingizga imkon beradi.

Constant constructors

- Biz class field larini tashqi o'zgarishlardan himoyalashni allaqachon o'rgandik. Fieldlarni private qilish orqali.
- Bu muammoni boshqa usul bilan ham bartaraf etishimiz mumkin, class field larini immutable qilish orqali field larni keyingi o'zgarishlardan himoya qilishimiz mumkin. Endi fieldlarni private qilishimiz shart emas.
- Dart da o'zgaruvchilarni immutable qilishning ikki usuli mavjud: final va const
- Class field lari o'z qiymatini runtime da olganligi sababli fieldlarni faqat final qilishimiz mumkin.
- Agar ma'lum bir classning instance lari hech qachon o'zgarmasa, class ning barcha field lari final bo'lsa, class ning barcha instance lari compile-time constantlari bo'lishini tamillash uchun const keywordini konstruktor oldiga qo'yishingiz mumkin.

- O'zgarmas bo'lishi bilan birga, agar class field lari ham bir xil bo'lsa dart bu instance larni bir deb biladi.



```
void main() {  
    User notConstIdentifier = User(1, "Nasibali");  
    const constIdentifier = User(2, "Alisher");  
}  
  
class User {  
    final int id;  
    final String name;  
  
    const User(this.id, this.name);  
}
```

Example



main.dart

```
void main() {  
  const x = 1.0;  
  final y = 1.0;  
  
  const Point(x, 10); // OK x is 'const'  
  const Point(x, y); // Compiler error - y is not 'const'  
}  
  
class Point {  
  final double x;  
  final double y;  
  
  const Point(this.x, this.y);  
}
```


Use case



```
class Bad {}

class Good {
  const Good();
}
```

Exercise

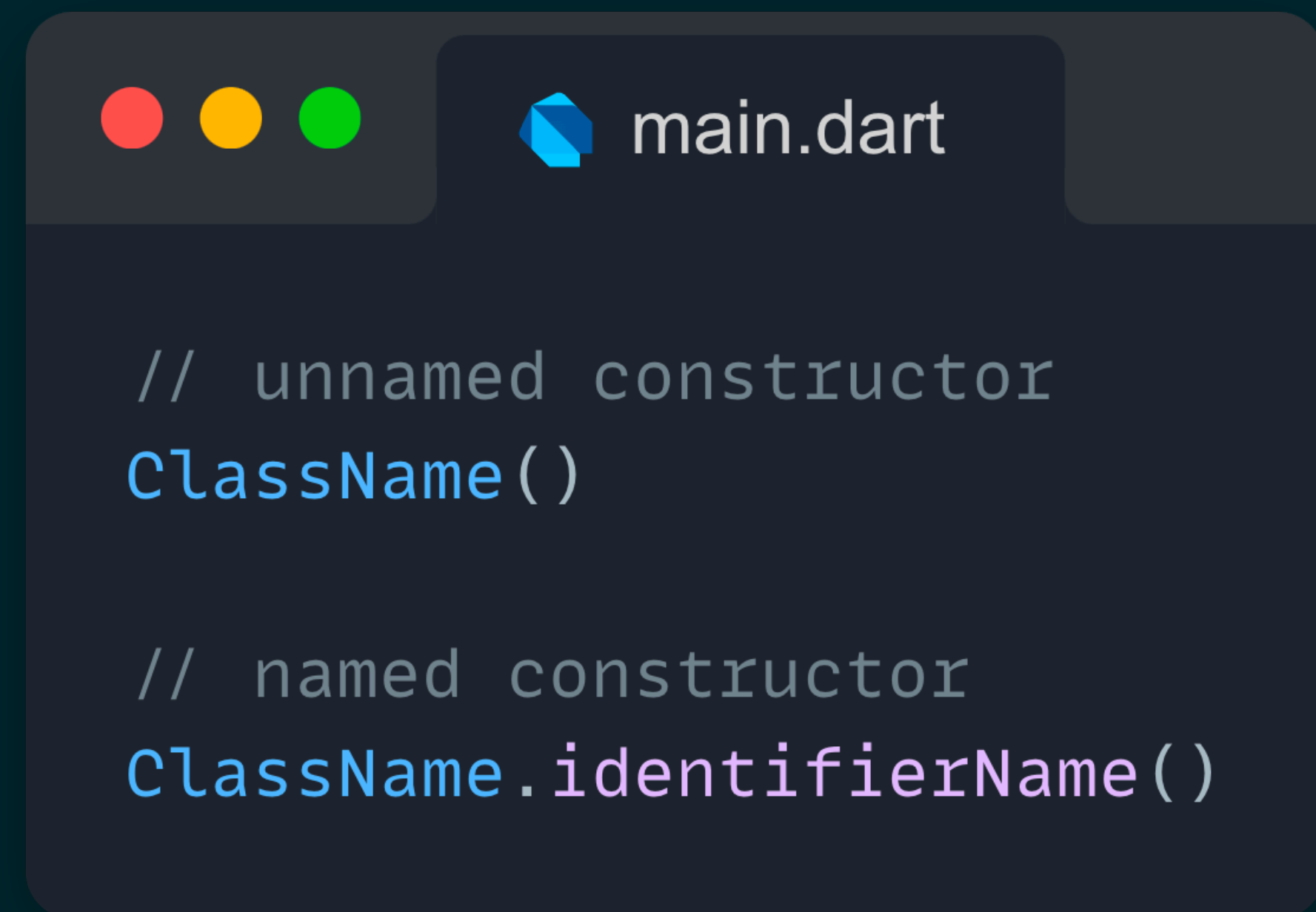
- Phone nomli class hosil qiling va classda const konstruktordan foydalaning

Named constructors

- Dart shuningdek, named konstruktor deb ataladigan konstruktorga ega, siz uni sinf nomiga identifikator qo'shish orqali yaratasiz.
- Hozirgacha yaratgan konstruktor larimiz unnamed konstruktor lardir.



```
ClassName.identifierName()
```



```
// unnamed constructor  
ClassName()  
  
// named constructor  
ClassName.identifierName()
```


- Ba'zan turli funktsiyalarni bajarish uchun bizga bir nechta konstruktor kerak bo'ladi. Lekin bir xil nomli bir nechta konstruktor yarata olmaysiz.
- Ushbu muammoni bartaraf etish uchun dart da turli nomlar bilan bir nechta konstruktorlar yaratish imkoni mavjud.



main.dart

```
void main() {  
    final anonymousUser = User.anonymous();  
    print(anonymousUser);  
}  
  
class User {  
    int id = 0;  
    String name = '';  
  
    User(this.id, this.name);  
  
    User.anonymous() {  
        id = 0;  
        name = 'anonymous';  
    }  
}
```

- Agar class da konstruktor e'lon qilingan bo'lsa, u named yoki unnamed bo'lishidan qat'iy nazar Dart tarafidan default konstruktor berilmaydi. Shuning uchun quyidagi misolda xatolik yuz beradi:



```
void main() {  
    // Error: doesn't have an unnamed constructor.  
    Example();  
}  
  
class Example {  
    const Example.named();  
}
```

- Named konstruktorlar orqali library-private konstruktorlarni yaratishingiz mumkin:



```
class Example {  
  const Example._();  
}
```

- Json dan obyektga named konstruktor orqali soddaroq o'tishingiz mumkin. Ammo bu misol yakuniy yechim emas!

```
void main() {  
  final json = {  
    "userId": 1,  
    "id": 1,  
    "title": "Yettinchi dars tugadimi?",  
    "completed": false  
  };  
  
  final todo = Todo.fromJson(json);  
  print(todo.title);  
}
```

```
class Todo {  
  late int userId;  
  late int id;  
  late String title;  
  late bool completed;  
  
  Todo({  
    required this.userId,  
    required this.id,  
    required this.title,  
    required this.completed,  
  });  
  
  Todo.fromJson(Map<String, Object?> json) {  
    userId = json['userId'] as int;  
    id = json['id'] as int;  
    title = json['title'] as String;  
    completed = json['completed'] as bool;  
  }  
}
```

Exercise

- Biror tizim azosini ifodalovchi Member class hosil qiling, class da premiumGold va premiumSilver toifali memberlarni hosil qiluvchi named konstruktorlarni hosil qiling.

Initializer list

- Konstruktor tanasi ishga tushgunga qadar instance o'zgaruvchilarini ishga tushirishingiz ham mumkin. Initsializatorlarni vergul bilan ajrating.
- Initializer list konstruktor tanasi ishga tushishidan oldin instance o'zgaruvchilarini o'rnatadi.



 main.dart

```
Point.fromJson(Map<String, double> json)
  : x = json['x']!,
    y = json['y']! {
  print('In Point.fromJson(): ($x, $y)');
}
```

- Initializer listlari final maydonlarni o'rnatishda qulaydir. Quyidagi misol boshlang'ich ro'yxatidagi uchta final fieldni ishga tushiradi.

```
import 'dart:math';

class Point {
  final double x;
  final double y;
  final double distanceFromorigin;

  Point(double x, double y)
    : x = x,
      y = y,
      distanceFromorigin = sqrt(x * x + y * y);
}

void main() {
  var p = Point(2, 3);
  print(p.distanceFromorigin);
}
```

Exercise

- Rasmda keltirilgan json ni obyektga aylantiring, initializer list orqali. Album nomli class orqali

```
[  
  {  
    "userId": 1,  
    "id": 1,  
    "title": "pictures"  
  },  
  {  
    "userId": 1,  
    "id": 2,  
    "title": "natures"  
  }  
]
```

Redirecting constructors

- Oldingi misolda ko'rganingizdek ikkala konstruktor ham o'zi fieldlarga qiymat biriktirmoqda.
- Bu codening qayta yozilayotganini bildiradi.
- Ushbu muammoni hal qilishning bir usuli - asosiy konstruktori nomlangan konstruktorda chaqirishdir. Bu yo'naltirish yoki qayta yo'naltirish deb ataladi. Buning uchun yana this kalit so'zidan foydalanasiz



main.dart

```
User.anonymous() : this(0, 'anonymous');
```

- Quyida ko'rib turganingizdek this keyword i orqali qiymat biriktirishni asosiy konstruktorga yo'nartirib yubormoqdamiz va endi bizga field lardagi default qiymatlar kerak emas.



```
class User {  
  int id;  
  String name;  
  
  User(this.id, this.name);  
  
  User.anonymous() : this(0, 'anonymous');  
}
```



 main.dart

```
class Fraction {  
  final int numerator;  
  final int denominator;  
  
  const Fraction(this.numerator, this.denominator);  
  
  // Initializer list:  
  const Fraction.whole(this.numerator) : denominator = 0;  
  
  // Redirecting constructors:  
  const Fraction.oneHalf() : this(1, 2);  
  const Fraction.one() : this.whole(1);  
}
```

Exercise

- Biror tizim azosini ifodalovchi Member class hosil qiling, class da “free”, “premiumGold” va “premiumSilver” toifali memberlarni hosil qiluvchi “redirecting” konstruktorlarni hosil qiling.

Factory constructors

- Siz hozirgacha ko'rgan barcha konstruktorlar generativ konstruktorlar hisoblanadi. Dart, shuningdek, factory(zavod) konstruktori deb ataladigan boshqa turdagi konstruktorni ham taqdim etadi.
- Factory constructor ob'ektlarni yaratishda ko'proq moslashuvchanlikni ta'minlaydi. Generativ konstruktor faqat class ning yangi nusxasini yaratishi mumkin, yani yangi obyekt. Biroq, factory konstruktorlari class ning mavjud nusxalarini yoki hatto uning pastki class larini qaytarishi mumkin. Bu class ning amalga oshirish tafsilotlarini uni ishlatadigan koddan yashirishni xohlaganingizda foydalidir.
- Factory constructor asosan factory kalit so'zi bilan boshlanadigan va class turidagi ob'ektni qaytaradigan maxsus method dir.

- Factory methodi user class ning yangi nusxasini yaratish va qaytarish uchun generativ konstruktordan foydalanadi.



```
class User {  
  final int id;  
  final String name;  
  
  const User(this.id, this.name);  
  
  factory User.premium(String name) {  
    return User(777, name);  
  }  
}
```

Example

- Real loyihalarda asosan misolda keltirilgan ko'rinishda Jsondan obyektga factory nomlangan fromJson konstruktori orqali o'tiladi.

```
class Todo {
  final int userId;
  final int id;
  final String title;
  final bool completed;

  Todo({
    required this.userId,
    required this.id,
    required this.title,
    required this.completed,
  });

  factory Todo.fromJson(Map<String, Object?> json) {
    final userId = json['userId'] as int;
    final id = json['id'] as int;
    final title = json['title'] as String;
    final completed = json['completed'] as bool;

    return Todo(userId: userId, id: id, title: title, completed: completed);
  }
}
```

Exercise

- `final Map<String, dynamic> map = {'id': 10, 'name': 'Manda'};`
- `final manda = User.fromJson(map);`
- Yuqorida keltirilgan map ni qabul qiladigan va `User.fromJson()` named constructori orqali yangi instace yaratib qaytaradigan factory constructori ni yarating.

Summary

Interview Questions

- Constructor nima?
- Const constructor qanday konstruktor? U qanday hosil qilinadi?
- Named constructor qanday konstruktor? U qanday hosil qilinadi?
- Agar dasturchi klass ichida konstruktor e'lon qilinsa, dart tarafidan berilgan oddiy default konstruktorni ishlata olamizmi?
- Konstuktorlarda ham library-private qilib e'lon qilish mumkinmi?
- Initializer list nima?
- Redirecting constructor qanday konstruktor? U qanday hosil qilinadi?
- Factory constructor qanday konstruktor? U qanday hosil qilinadi?
- fromJson nomli konstruktori nima vazifani bajaradi?

Homework

1. Vehicle nomli klass tuzilsin. Bu klassni imkon qadar mukammal modellashtiring, yani fieldlariga yaxshi e'tibor berilishi kerak va shu klassda quyidagilar bo'lishi kerak:
 1. generative parameterize constructor tuzilsin
 2. truck nomli named constructor tuzilsin
 3. bus nomli named constructor tuzilsin
 4. sport nomli named constructor tuzilsin
 5. car nomli named redirect constructor tuzilsin
 6. balonlar soni va o'rindiqlar soni uchun getter/setter yozilsin
 7. main funksiyani ichida yuqoridagi constructorlardan foydalanib bir nechta object lar hosil qiling

2. Employee nomli klass tuzilsin. Bu klassni imkon qadar mukammal modellashtiring, yani fieldlariga yaxshi e'tibor berilishi kerak va shu klassda quyidagilar bo'lishi kerak:
 1. generative parameterize constructor tuzilsin
 2. intern nomli named constructor tuzilsin
 3. const constructor tuzilsin
 4. factory constructor tuzilsin
 5. getter/setter yozilsin
 6. main funksiyani ichida yuqoridagi constructorlardan foydalanib bir nechta object lar hosil qiling

3. Product nomli klass tuzilsin. Bu klassni imkon qadar mukammal modellashtiring, yani fieldlariga yaxshi e'tibor berilishi kerak va shu klassda quyidagilar bo'lishi kerak:
 1. generative parameterize constructor tuzilsin
 2. fruit nomli named constructor tuzilsin
 3. drink nomli named constructor tuzilsin
 4. const constructor tuzilsin
 5. factory constructor tuzilsin
 6. getter/setter yozilsin
 7. main funksiyani ichida yuqoridagi constructorlardan foydalanib bir nechta object lar hosil qiling

4. Computer nomli klass tuzilsin. Bu klassni imkon qadar mukammal modellashtiring, yani fieldlariga yaxshi e'tibor berilishi kerak va shu klassda quyidagilar bo'lishi kerak:
 1. generative parameterize constructor tuzilsin
 2. laptop nomli named constructor tuzilsin
 3. desktop nomli named constructor tuzilsin
 4. const constructor tuzilsin
 5. factory constructor tuzilsin
 6. getter/setter yozilsin
 7. main funksiyani ichida yuqoridagi constructorlardan foydalanib bir nechta object lar hosil qiling

5. Rasmda keltirilgan jsonni obyektga aylantiring.

```
final photos = [  
    {  
        "albumId": 1,  
        "id": 1,  
        "title": "accusamus beatae ad facilis cum similique qui sunt",  
        "url": "https://via.placeholder.com/600/92c952",  
        "thumbnailUrl": "https://via.placeholder.com/150/92c952"  
    },  
    {  
        "albumId": 1,  
        "id": 2,  
        "title": "reprehenderit est deserunt velit ipsam",  
        "url": "https://via.placeholder.com/600/771796",  
        "thumbnailUrl": "https://via.placeholder.com/150/771796"  
    },  
    {  
        "albumId": 1,  
        "id": 3,  
        "title": "officia porro iure quia iusto qui ipsa ut modi",  
        "url": "https://via.placeholder.com/600/24f355",  
        "thumbnailUrl": "https://via.placeholder.com/150/24f355"  
    }  
];
```

Q&A

Thank you for your time!