Ростелеком
Информационные технологии

Введение в Dart

Flutter

Школа мобильной разработки EPIC

Немного истории

Dart это мультипарадигмальный язык программирования с возможностью компиляции под популярные платформы.

Made by Google

Типы компиляции кода

- JIT (Just-in-Time)

  Динамическая компиляция «точно в нужное время» основанная на байт-коде.

- AOT (Ahead-of-Time)

  Статическая компиляция перед исполнением сразу в машинный код без посредников.

Паттерны компиляции Dart

- **Script**

  Самый распространенный режим JIT. Код Dart выполняется напрямую с помощью инструмента командной строки Dart VM.

- **Script snapshot**

  Режим JIT. В отличие от обычного Script, происходит упаковка кода в токениризированный код.

- **Application snapshot**

  Режим JIT. Похож на своего рода дамп из среды выполнения. Он включает в себя проанализированные классы и функции из исходного кода для среды выполнения, но этот вид снимка зависит от архитектуры.

- **AOT**

  В этом режиме исходный код Dart будет переведен в файлы сборки, затем файлы сборки будут скомпилированы ассемблером в двоичный код для различных архитектур.

Окружение разработки

Браузер

+

dartpad.dev

dart.dev/get-dart

+

VS code || Android studio

Переменные в Dart

- Константы (const)

  Неизменяемые переменная, которая определяется на уровне компиляции программы.

- Изменяемые переменные (var)

  Определяются и вычисляются при выполнении программы и могут быть изменены в будущем.

- Неизменяемые переменные (final)

  Определяются и вычисляются при выполнении программы, но не могут быть изменены.

Определение переменных

```
const name = value;
```

```
~~const type name = value;~~
```

```
var name = value;
```

```
~~type name = value;~~
type name;
```

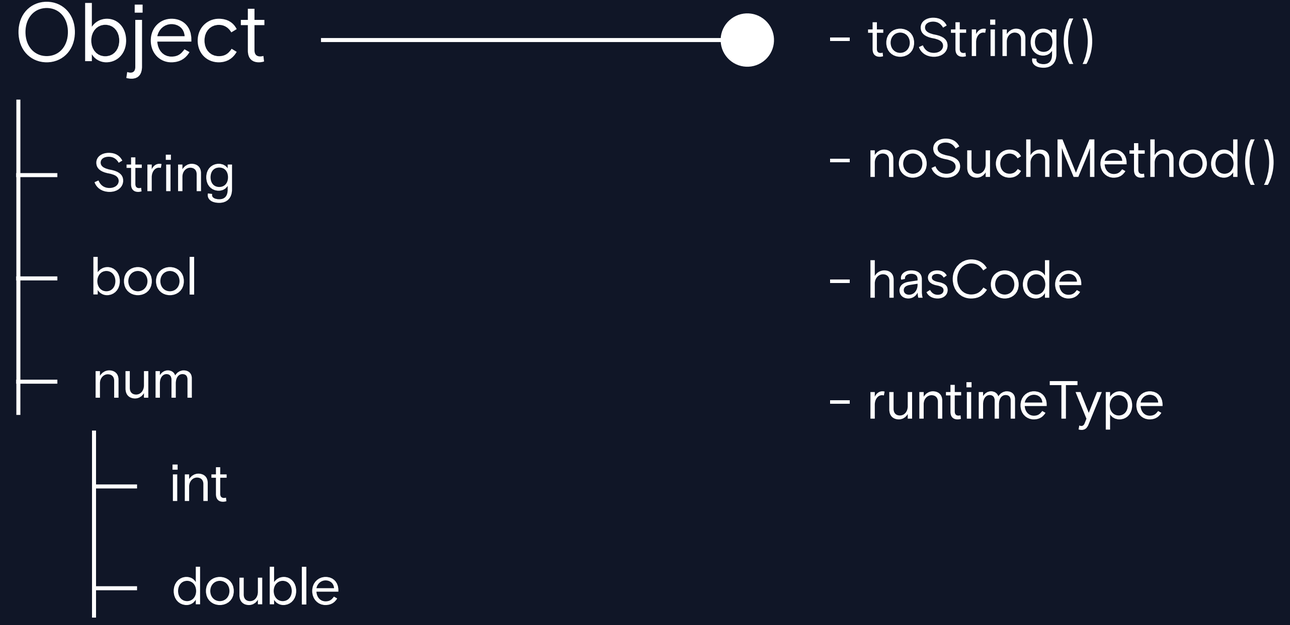
```
final name = value;
```

```
~~final type name = value;~~
final type name;
```

Типы данных

Типы данных в Dart это классы и их производные.
И все типы наследуются от супер-класса Object.

*о классах и ООП мы будем подробнее говорить позже.



Null safety

По умолчанию если мы объявляем переменную с явным указанием типа, то мы обязаны определить ее значение перед использованием, так как она не может равна null

т.е. все переменные non-nullable

Если мы допускаем возможность значения null для переменной, то должны использовать опционные типы, проверка которых лежит всегда на программисте.

Операторы для работы с null

```
final String? name;
```
Объявления опционный переменной

```
final String name2 = name!;
```
Принудительное извлечение значения из опциона

```
final String name3 = name ?? "default";
```
Проверка на null

Основные операторы

=, !=	Идентичность
>, >=, <, <=	Больше/меньше
is, is!	Сравнение типов
&&	Логическое И
||	Логическое ИЛИ

+, -, *, /, %	Арифметические операторы
+=, -=, *=, /=	Операторы присвоения
??	Проверка на null
exp1 ? exp2 : exp3	Тернарный оператор
++, —	Убранный плюс/минус

Контроль выполнения программы

```
if
else if
else
```

```
switch
    case
        break
    default
```

```
for (var; conditional; action)
for (el in collection) {}
while (conditional) {}
do {} while(conditional)
```