Enums in Dart

Proposed Specification

We might choose to modify the sections on inheritance and import/export to localize or repeat the restrictions on enums.

Enums

An *enumerated type*, or *enum*, is used to represent a fixed number of constant values.

enumType:
metadata **enum** *id '{'id [',' id]* [','] '}'*
;

The declaration of an enum of the form **enum** E = id_1, … id_n;
has the same effect as a class declaration

```
metadata class E {
  final int index;
  final String _name;
  const E(this.index);
   static const E id1 = const E(0, 'id1');
    …
   static const E idn = const E(n - 1, 'idn-1');

  static const List<E> values = const <E>[id1 … idn];
  String toString() => 'E.$_name';
}
```

Except for the following differences: It is a compile-time error to subclass, mix-in or implement an enum. It is also a compile-time error to explicitly instantiate an enum via **new** or **const** or to access its private fields.

Switch

The *switch statement* supports dispatching control among a large number of cases.

switchStatement:
    switch *'('* *expression* *')'* *'{'* *switchCase** *defaultCase?* *'}'*
  ;

switchCase:
    *label** *(***case** *expression* *':')* *statements*
  ;

defaultCase:
    *label** **default** *':'* *statements*
  ;

Given a switch statement of the form **switch** (*e*) { label$_{11}$ … label$_{1j1}$ **case** e_1**:** s_1 … label$_{n1}$ **..**label$_{njn}$ **case** e_n**:** s_n **default:** s_{n+1}} or the form **switch** (*e*) { label$_{11}$ … label$_{1j1}$ **case** e_1**:** s_1 … label$_{n1}$ **..**label$_{njn}$ **case** e_n**:** s_n}, it is a compile-time error if the expressions e_k are not compile-time constants, for all 1 <= *k* <= *n*. It is a compile-time error if values of the expressions e_k are not either:

- instances of the same class *C*, for all 1 <= *k* <= *n*, or
- instances of a class that implements int, for all 1 <= *k* <= *n*.