"switch" statements should have "default" clauses (squid:SwitchLastCaseIsDefaultCheck)

Code smell Critical

The requirement for a final `default` clause is defensive programming. The clause should either take appropriate action, or contain a suitable comment as to why no action is taken.

Noncompliant Code Example

```
switch (param) {  //missing default clause
  case 0:
    doSomething();
    break;
  case 1:
    doSomethingElse();
    break;
}

switch (param) {
  default: // default clause should be the last one
    error();
    break;
  case 0:
    doSomething();
    break;
  case 1:
    doSomethingElse();
    break;
}
```

Compliant Solution

```
switch (param) {
  case 0:
    doSomething();
    break;
  case 1:
    doSomethingElse();
    break;
  default:
    error();
    break;
}
```

Exceptions

If the `switch` parameter is an `Enum` and if all the constants of this enum are used in the `case` statements, then no `default` clause is expected.

Example:

```
public enum Day {
    SUNDAY, MONDAY
}
...
switch(day) {
  case SUNDAY:
    doSomething();
    break;
  case MONDAY:
    doSomethingElse();
    break;
}
```

See

- MISRA C:2004, 15.0 - The MISRA C *switch* syntax shall be used.
- MISRA C:2004, 15.3 - The final clause of a switch statement shall be the default clause
- MISRA C++:2008, 6-4-3 - A switch statement shall be a well-formed switch statement.
- MISRA C++:2008, 6-4-6 - The final clause of a switch statement shall be the default-clause
- MISRA C:2012, 16.1 - All switch statements shall be well-formed
- MISRA C:2012, 16.4 - Every *switch* statement shall have a *default* label
- MISRA C:2012, 16.5 - A *default* label shall appear as either the first or the last *switch label* of a *switch* statement
- MITRE, CWE-478 - Missing Default Case in Switch Statement
- CERT, MSC01-C. - Strive for logical completeness