

## switch

The `switch` statement allows you to choose between arbitrary number of code sections to be executed depending on the value of an integer expression. The general syntax is:

```
switch( <integer expression> ) {  
  case <integer constant 1>:  
    <statements to be executed in case 1>  
    break;  
  case <integer constant 2>:  
    <statements to be executed in case 2>  
    break;  
  ...  
  default:  
    <statements to be executed if no cases apply>  
    break;  
}
```

The `break` statements at the end of each `case` section tell Java that the `switch` statement execution is finished. If you do not put `break`, Java will continue executing the next section, no matter that it is marked as a different case. The `default` section is executed when the integer expression does not match any of the cases. It is optional.

The integer values that correspond to different cases of the switch are usually pre-defined as integer constants. Imagine you are developing a model of an overhead bridge crane where the crane is an agent controlled by a set of commands. The response of the crane to the commands can be programmed in the form of a `switch` statement:

```
switch( command ) {  
  case MOVE_RIGHT:  
    speed = 10;  
    break;  
  case MOVE_LEFT:  
    speed = -10;  
    break;  
  case STOP:  
    speed = 0;  
    break;  
  case RAISE:  
    ...  
    break;  
  case LOWER:  
    ...  
    break;  
  default:  
    error( "Invalid command: " + command );  
}
```