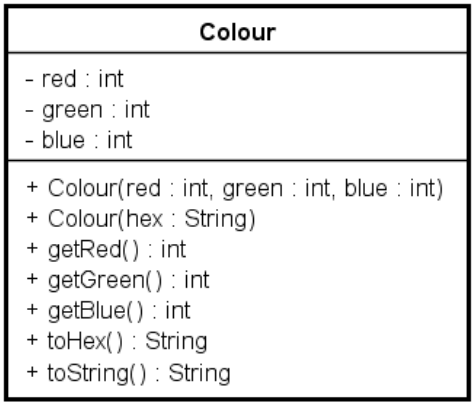
Exercise: Colour, version 1



Create a new module in IntelliJ and name it `Colour_v1`. Add a new class called `Colour` representing a colour as the three colour values `red`, `green` and `blue`. A legal colour value is given as an integer in the interval [0; 255]. You have to follow the UML class diagram as shown (it is ok to add private methods not shown in the class diagram).

A hexadecimal representation of the colour is `"#"` and six characters (digits or letters A-F). The first two representing a hexadecimal of the `red` value, the next two a hexadecimal of the `green` value and the last two a hexadecimal of the `blue` value.

A hexadecimal "digit" is one of the following: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F} and to convert a value (in base 10) in the interval [0; 255] do the following:

The 1st "digit": divide the number by 16. If the result is {10, 11, 12, 13, 14, 15} then use the equivalent letter from the list {A, B, C, D, E, F} instead. The 2nd "digit": calculate the value modulus 16. If the result is {10, 11, 12, 13, 14, 15} then use the equivalent letter from the list {A, B, C, D, E, F} instead.

Example: The colour medium turquoise has the values `red`=72, `green`=209 and `blue`=204

Red: 72 to hexadecimal (base 16): *1st "digit" = 72 / 16 = **4***
*2nd "digit" = 72 % 16 = **8***

Green: 209 to hexadecimal (base 16): *1st "digit" = 209 / 16 = 13 -> **D***
*2nd "digit" = 209 % 16 = **1***

Blue: 204 to hexadecimal (base 16): *1st "digit" = 204 / 16 = 12 -> **C***
*2nd "digit" = 204 % 16 = 12 -> **C***

*Therefore, the hexadecimal representation of this colour is **"#48D1CC"***

Medium turquoise
{R,G,B} = {72, 209, 204}
#48D1CC

To convert a hexadecimal value (base 16) into a decimal number (base 10) do the following:
Multiply the 1st "digit" hexadecimal number by 16 and add the 2nd "digit" hexadecimal number. Before the calculation {A, B, C, D, E, F} is translated to {10, 11, 12, 13, 14, 15}.

Converting the hexadecimal string ***"#48D1CC"*** back to red, green and blue values:

*Red: **48** hexadecimal to base 10:* *From 1st "digit" = **4** * 16 = 64*
*From 2nd "digit" = **8***
*Total: 64+8 = **72***

*Green: **D1** hexadecimal to base 10:* *From 1st "digit" = **13** * 16 = 208* (Note: D=13)
From 2nd "digit" = 1
*Total: 208+1 = **209***

*Blue: **CC** hexadecimal to base 10:* *From 1st "digit" = **12** * 16 = 192* (Note: C=12)
*From 2nd "digit" = **12*** (Note: C=12)
*Total: 192+12 = **204***

The class `Colour` has:

a) Three instance variables
b) A constructor taking values for each instance variable. If values are not integers from 0 to 255 then change the values accordingly.
c) A constructor taking a hexadecimal string in the format "`#RRGGBB`" with RR representing hexadecimal value for red, GG representing hexadecimal value for green and BB representing hexadecimal value for blue. If the format is wrong then set the values to proper default values.
d) Getters for all instance variables
e) A method `toHex` returning the hexadecimal string for the colour in the format "`#RRGGBB`" with RR representing hexadecimal value for red, GG representing hexadecimal value for green and BB representing hexadecimal value for blue.
f) A method `toString` returning a string with the information of the colour in a proper format.

Implement a test class with a main method for your class `Colour` (name the class `ColourTest`).