

Colour

A fairly classic way to set colours in a computer program consists of splitting the colour in three components: red, green and blue. Each of these components is an integer and takes a value between 0 and 255 (boundaries included). Within this system, the greyscales correspond to the colours in which the three components have the same value.

If a photo must be printed as a greyscale picture, every colour must be converted to a specific greyscale. You could use the mean value, for example, but this gives a wrong image, since the human eye is not equally sensitive to every colour. This is why the following conversion is often used:

$$\text{greyscale} = 30\% \text{ of red} + 59\% \text{ of green} + 11\% \text{ of blue}$$

If a picture should be made unrecognizable, one sometimes inverts parts of the picture. Here, every colour is replaced by its inverse colour. In order to obtain this colour, you should replace every component by 255 minus that component.

Assignment

Your assignment consists of making a class `Colour` with the following methods:

1. The initializing method `__init__` gets the three colour components as parameters, where each component gets the value 0 as a standard. Make sure that only components between 0 and 255 are used: values smaller than 0 are replaced with 0, values larger than 255 are replaced with 255.
2. The method `__str__` represents this colour as a string in the format (R, G, B) , where R, G and B indicate the different components.
3. The method `inverse` prints a new `Colour` that represents the inverse colour.
4. The method `greyscale` prints a new `Colour` that represents the greyscale of this colour according to the calculation above.

Example

```
>>> c = Colour(128, 62, -50)
>>> print(k)
(128, 62, 0)
>>> c = Colour(128, 62, 350)
>>> c
(128, 62, 255)
>>> g = c.greyscale()
>>> g
(103, 103, 103)
>>> i = c.inverse()
>>> i
(127, 193, 0)
>>> i.greyscale()
(151, 151, 151)
```

Een vrij klassieke manier om kleuren voor te stellen in computerprogramma's bestaat erin een kleur op te splitsen in drie componenten: rood, groen en blauw. Elk van deze componenten is

een geheel getal en neemt een waarde tussen 0 en 255 (inclusief de grenzen) aan. Binnen dit systeem komen de grijswaarden overeen met de kleuren waarin de drie componenten dezelfde waarde hebben.

Als een foto als een grijswaarde-afbeelding moet afgedrukt worden, moet elke kleur omgezet worden naar een specifieke grijswaarde. Je zou hiervoor bijvoorbeeld de gemiddelde waarde kunnen gebruiken, maar dit geeft een verkeerde beeld, doordat het menselijke oog niet even gevoelig is voor elke kleur. Daarom wordt vaak gebruik gemaakt van de volgende omzetting:

$$\text{grijswaarde} = 30\% \text{ van rood} + 59\% \text{ van groen} + 11\% \text{ van blauw}$$

Als een foto in beperkte mate onherkenbaar moet gemaakt worden, gaat men soms bepaalde stukken van de foto inverteren. Hierbij wordt elke kleur vervangen door zijn inverse kleur. Om deze kleur te verkrijgen, moet je elke component vervangen door 255 min die component.

Opgave

Je opdracht bestaat erin een klasse `Kleur` aan te maken met volgende methoden:

1. De initialisatiemethode `__init__` krijgt als parameters de drie kleurcomponenten mee, waarbij elke kleurcomponent standaard de waarde 0 krijgt. Zorg ervoor dat enkel componenten tussen 0 en 255 gebruikt worden: waarden kleiner dan 0 vervang je door 0, waarden groter dan 255 vervang je door 255.
2. De methode `__str__` stelt deze kleur als string voor in de vorm (R, G, B) , waarbij R, G en B voor de verschillende componenten staan.
3. De methode `invers` geeft een nieuwe `Kleur` terug die de inverse kleur voorstelt.
4. De methode `grijswaarde` geeft een nieuwe `Kleur` terug die de grijswaarde van deze kleur voorstelt volgens bovenstaande berekening.

Voorbeeld

```
>>> k = Kleur(128, 62, -50)
>>> print(k)
(128, 62, 0)
>>> k = Kleur(128, 62, 350)
>>> k
(128, 62, 255)
>>> g = k.grijswaarde()
>>> g
(103, 103, 103)
>>> i = k.invers()
>>> i
(127, 193, 0)
>>> i.grijswaarde()
(151, 151, 151)
```