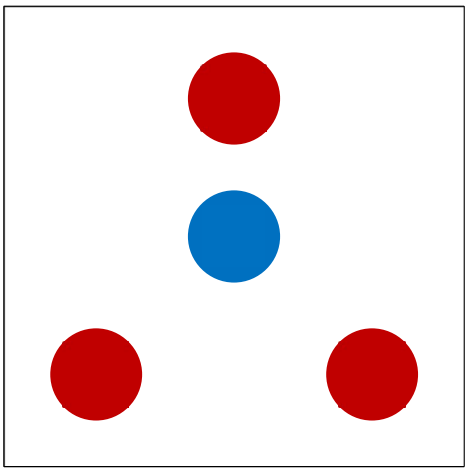
Programming 1 – live exercises

Class declaration and use

Task 1: Declaration of the `Circle` class



A drawing sheet measures 500 x 500, with the origin of the coordinates in the upper left corner of the drawing sheet.

a) Declare a `Circle` class which represents circles that are drawn on this sheet.

b) You should focus on the attributes of the circles. (We'll look at the behaviour later.) In principle, circles are characterised by a radius, an x coordinate and a y coordinate (together representing the centre point) as well as a colour. Declare all object variables of the `Circle` class.

Solution:

Circle
radius : int xPos : int yPos : int colour : Colour

```
public class Circle {
  private int radius;
  private int xPos;
 private int yPos;
 private Colour colour;   or String colour;
```

Task 2: Using the `Circle` class

a) Write an application for the `Circle` class in which a circle is created. Name the corresponding application class `MainSingleStandardCircle`, and within it declare a reference variable `circle1` that references the circle object created.

b) Assign the following values to the object variables of the object created in task (a): radius (50), x coordinate (200), y coordinate (200) and colour (Colour.MAGENTA).

c) Then call up the given method `draw()`.

Solution:

```
public class MainSingleStandardCircle {

    public static void main(String[] args) {

        // Create and draw new circle
        Circle circle1 = new circle();

        Circle1.radius = 50;
        Circle1.xPos = 200;
        Circle1.yPos = 200;
        Circle1.colour = Colour.MAGENTA;

        circle1.draw();

    }

}
```

Task 3: Constructors for the `Circle` class

a) Write a **default constructor** for the `Circle` class which assigns the following standard (default) values to each newly-created object: radius (50), x coordinate (200), y coordinate (200) and colour (Colour.MAGENTA).

b) Declare a **constructor with multiple parameters** for the `Circle` class. When calling up the constructor, **four values** should be passed, namely a specific radius value, an x coordinate value, a y coordinate value and a specific colour.

```
Circle() {
     radius = 50;
     xPos = 200;
     yPos = 200;
     colour = Colour.MAGENTA;
    }


*/
Circle(int r, int x, int y, Colour initialColour) {
        radius = r;
        xPos = x;
        yPos = y;
        colour = initialColour;
      }
```

Task 4: Method for the `Circle` class

Write a `relocate` method for the `Circle` class, which moves a circle by changing the coordinates of the circle. This means that an integer value should be added to both the x coordinate and the y coordinate. Two integer values can be passed to the method, and it does not return any result.

```
void relocate(int xDelta, int yDelta) {
     xPos = xPos + xDelta;
     yPos = yPos + yDelta;
    }
```

Task 5: Setter and getter for the `Circle` class

Write the appropriate setter methods and getter methods for the two object variables Radius and Colour of the `Circle` class.

```
void setRadius(int newRadius) {
        radius = newRadius;
    }

void setColour(Colour newColour) {
        colour = newColour;
    }

int getRadius() {
        return radius;
    }

Colour getColour() {
        return colour;
    }
```