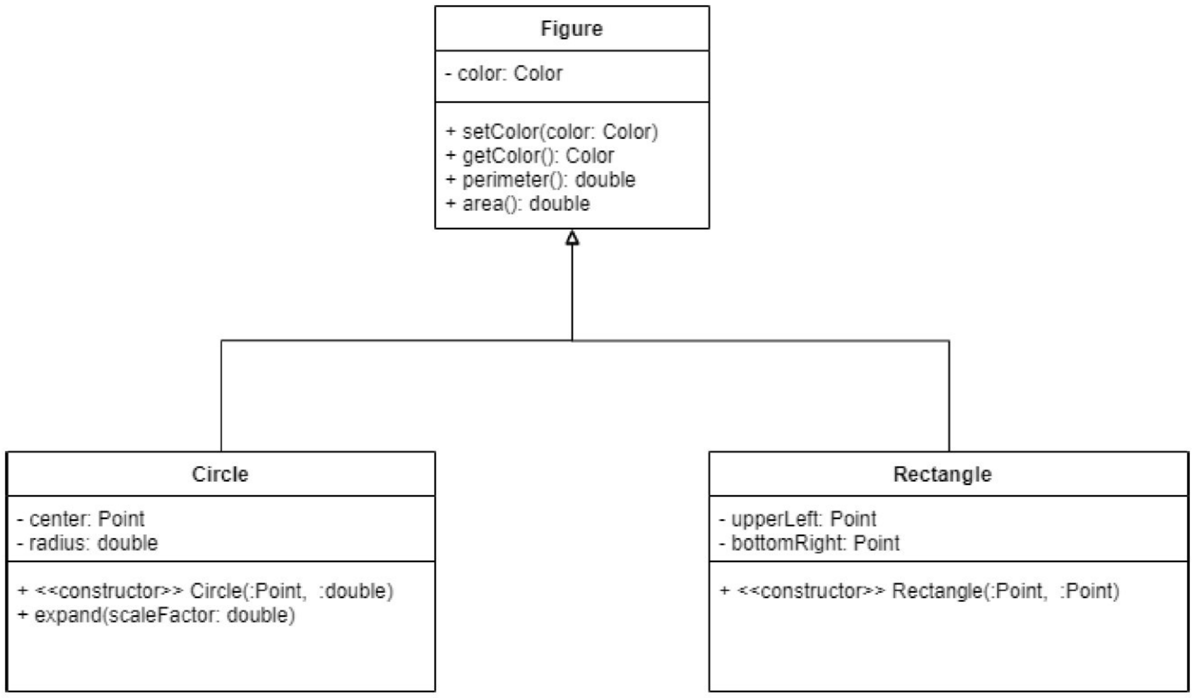
# Metodologias de Trabalho em Equipa

Suppose we want to write a java application that manages shapes and has these requirements:

- Shapes can be circles or rectangles.
- Shapes exist in some two-dimensional space
- Shapes can be colored
- We want to compute areas and perimeters

If we want to take advantages of the similarities of the 3 concepts, we must use inheritance. In UML, we can capture the similarities and differences like this:



Working in a team with 3 members (M1, M2 and M3), implement this application according to the diagram above.

Perform the following tasks:

1. [Member M1] Create a public java repository in Github. Invite team members M2 and M3 as collaborators.
2. [M2] Create an IntelliJ Java project and add the remote created in 1. Push the java project to the remote repository.
3. [M1 and M3] Clone the Github repository to their development environments
4. In parallel:
    a. [M1] Create the Figure abstract class skeleton with color attribute. Push, Commit and Pull changes.
    b. [M2] create Point Class (point coordinate attributes, constructors, getter and setter methods). Push, Commit and Pull changes frequently.
    c. [M3] add IntelliJ Idea project files to *.gitignore*.
5. Concurrently:
    a. [M2] create Circle class. Push, Commit and Pull changes frequently.
    b. [M3] create Rectangle class. Push, Commit and Pull changes frequently.
6. In parallel:
    a. [M1] Create *getColor* and *setColors* methods. Push, Commit and Pull changes frequently.
    b. [M2] Create abstract method *perimeter* in Figure class and implement perimeter methods in classes Circle and Rectangle. Push, Commit and Pull changes frequently.
    c. [M3] Create abstract method *area* in Figure class and implement area methods in classes Circle and Rectangle. Push, Commit and Pull changes frequently.