

Réflexion sur le Principe de Substitution de Liskov (LSP)

substitution de comportement

1/ Réaliser une classe Rectangle très simple comprenant :

- deux attributs width et height,
- leurs accesseurs,
- un constructeur,
- une méthode toString.

2/ Soit la méthode statique suivante :

```
public static void mystery(Rectangle r)
{
    r.setWidth(5.0);
    r.setHeight(4.0);
}
```

ajoutez-lui (à la fin) une assertion vérifiant que la hauteur multipliée par la largeur donnent 20.0

3/ Créer une sous-classe Square :

- sans nouvel attribut
- un constructeur qui prend en paramètre le côté
- redéfinissant l'écriture de width et height pour assurer qu'elles sont égales
- introduisez une assertion vérifiant que lors de toute modification width reste égale à height

4/ Que se passe-t-il lors de l'exécution de ce programme

```
Rectangle r = new Rectangle(8.0, 7.0);
mystery(r);

r = new Square(3.0);
mystery(r);
```

Que doit-on en penser ?