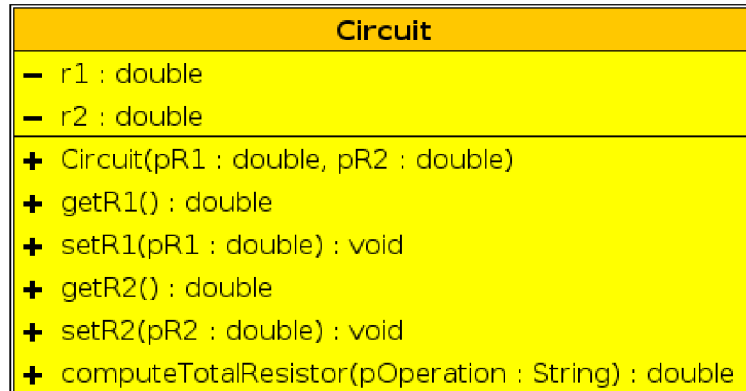


## Exercices supplémentaires – alternative

### Exercice A1

Développez une classe **Circuit** (projet : Exercice-A1) décrite par le diagramme de classes ci-dessus. Cette classe permet de manipuler des circuits électroniques – principalement des résistances (**r1** et **r2**). Programmez également une classe de test.



La méthode **computeTotalResistor** permet de calculer la résistance totale d'un circuit en parallèle ou en série. Pour cela, le paramètre **pOperation** contient le type du branchement souhaité (« s » pour branchement en série et « p » pour branchement en parallèle). Toute autre valeur (par exemple « z ») doit afficher un message d'erreur (tel que « Illegal operation 'z' ! ») et retourner la valeur -1. Attention, il faut traiter le cas où les valeurs des résistances peuvent être nulles.

$$R_p = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2} + \dots + \frac{1}{R_n}} \quad R_s = R_1 + R_2 + \dots + R_n$$

Modifiez la méthode pour également accepter les lettres majuscules « P » et « S »

### Exercice A2

Développez une classe **Dice** (projet : Exercice-A2) décrite par le diagramme de classes ci-dessus. Les attributs **player1** et **player2** contiennent le nom des deux joueurs et **points1** et **points2** le résultat du jet (*throw*, *Wurf*) de chaque joueur. La méthode **throwDices** génère aléatoirement le résultat de chaque jet avec un dé à six faces. Programmez également une classe de test.



La méthode **printResult** affiche le résultat comme ci-dessus :

```
Result of the throws:
  Tom => 5
  Jerry => 5
Deuce - there is no winner!
```

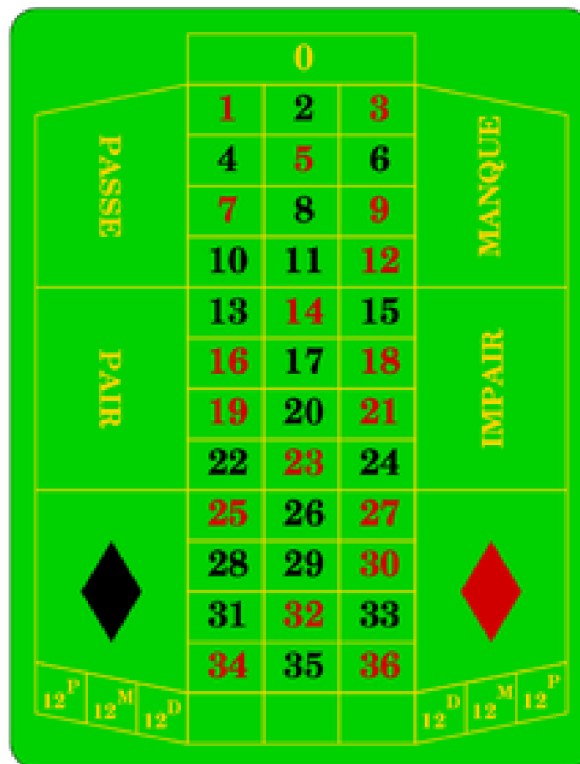
```
Result of the throws:
  Tom => 4
  Jerry => 6
The winner is: Jerry
```

**Exercice A3**

Développez une classe **Roulette** (projet : Exercice-A3) décrite par le diagramme de classes ci-dessous. Cette classe permet de simuler la roulette d'une casino. Programmez également une classe de test.

L'attribut **name** contient le « nom » de la roulette, alors que l'attribut **res** contient le résultat d'un tour de roulette. Le résultat est généré aléatoirement par la méthode **spin**.

Roulette	
-	name : String
-	res : int
+	Roulette(pName : String)
+	spin() : void
+	printResult() : void



Dans un casino, au jeu de roulette, le croupier doit toujours annoncer les résultats aux joueurs d'une manière définie.

Le résultat est toujours un nombre entier entre 0 et 36. Les nombres sont groupés en plusieurs catégories (voir image ci-contre) :

- le zéro (0) – le casino gagne ;
- un nombre inférieur ou égal à 18 est annoncé comme *manque* ; au dessus de 18, un nombre est annoncé comme *passee* ;
- certains nombres sont marqués en *rouge* (sur la figure blanc), d'autres sont marqués en *noir* (sur la figure gris) sur le tapis de jeu.

Cette « annonce » se fait avec la méthode **printResult** – exemples d'annonces de résultats :

- |                         |   |
|-------------------------|---|
| si le nombre est le 5,  | l'annonce est « R22 - le 5, rouge, impair et manque » |
| si le nombre est le 23, | l'annonce est « R22 - le 23, rouge, impair et passe » |
| si le nombre est le 30, | l'annonce est « R22 - le 28, noir, pair et passe »    |
| si le nombre est le 0,  | l'annonce est « R22 - le 0, le casino gagne! »        |

Le nom de la roulette est également à afficher lors de l'annonce (R22 dans les exemples ci-dessus).