

# Exercices - Les objets

---

## Exercice 1 : Feux tricolores

---

Créer un objet permettant de gérer un carrefour. Notre objet possèdera quatre attributs Feux 1, Feux 2, Feux 3 et Feux 4.

Les feux 1 et 3 sont liés et les feux 2 et 4 aussi.

Les feux liés sont soit vert soit rouge en même temps, sauf que les feux 1 et 2 ont toujours un temps d'avance.

Une méthode **etapeSuivante** vous permettra de passer à l'étape suivante et d'afficher l'état de chaque feu dans la console. Voici les différentes étapes possibles :

1. feux1 = "rouge", feux2="vert", feux3 = "rouge", feux4="vert"
2. feux1 = "rouge", feux2="orange", feux3 = "rouge", feux4="vert"
3. feux1 = "rouge", feux2="rouge", feux3 = "rouge", feux4="orange"
4. feux1 = "vert", feux2="rouge", feux3 = "vert", feux4="rouge"
5. feux1 = "orange", feux2="rouge", feux3 = "vert", feux4="rouge"
6. feux1 = "rouge", feux2="rouge", feux3 = "orange", feux4="rouge"

## Exercice 2 : Enigme du berger, loup, chèvre et chou

---

Créer un objet qui permet de résoudre l'énigme du berger qui doit faire traverser le loup, la chèvre et le chou.

L'objet contiendra deux tableaux (riveGauche et riveDroite).

Les quatre protagonistes commencent sur la rive gauche à l'initialisation. Tout le monde doit traverser la rivière et atterrir sur la rive droite.

Seulement la barque permettant de traverser la rivière ne peut contenir que le berger et un animal/légume.

Attention le loup ne peut pas rester sur une rive avec la chèvre sans que le berger soit présent et il en va de même pour la chèvre et le loup.

Vous aurez besoin des méthodes suivantes pour résoudre le jeu :

- **traverser** : prendra en argument un animal/légume si ce dernier n'est pas sur la même rive que le berger il ne se passe rien (car le berger est sur l'autre rive).

Sinon on le déplacera sur l'autre rive (d'un tableau à l'autre) avec le berger.

Ensuite, cette méthode appellera la méthode **verifierRives**.

- **verifierRives** : Cette méthode vérifie si le loup et la chèvre/ la chèvre et le choux ne sont pas seuls sur une rive. Si c'est le cas on réinitialise la partie avec la méthode **initialisation**. Si tout le monde est du même côté on avertit le joueur que le jeu est terminé puis on utilise la méthode **initialisation** pour relancer le jeu.
- **afficherRives** : Affiche le contenu des deux rives.
- **initialisation** : Remet les quatre protagonistes sur la rive gauche.

## Exercice 3 : Le distributeur automatique

Notre distributeur automatique contiendra deux catégories d'éléments (les éléments liquide et les éléments solides), le distributeur peut contenir maximum 8 éléments différents et une quantité de 10 maximum par élément.

Ex: 10 mars, 10 snickers, 10 chips, 10 cookies, 10 coca, 10 fanta, 10 orangina et 10 bières MAX.

Les produits liquides coûtent 3€.

Les produits solides coûtent 2.50€.

Créer l'objet distributeur qui permettra de réaliser les actions suivantes (méthodes, getter ou setter) :

- **remplir** : Prendra en paramètre un tableau (à deux dimensions) contenant les différents éléments qui rempliront le distributeur au maximum.

Ex : le tableau `[["coca", "badois", "fanta"], ["snickers", "chips", "mars"]]` remplira le distributeur de 10 coca, 10 fanta, 10 badois, 10 chips 10 mars et 10 snickers.

- **alumerLeDistributeur** : Pour chaque élément différents, le distributeur attribue un **code produit** entre 100 et 150 (aléatoire mais unique) qui permettra plus tard à l'utilisateur de choisir son produit. Puis un attribut dira que le distributeur est allumé.
- **afficherLesProduits** : dans la console, on affichera chaque produit ainsi que le code qui permettra à l'utilisateur de **commander**.

- **commander** : en prenant en argument un code produit, le distributeur affichera à l'utilisateur le prix du produit désiré.
- **ajouterUnePiece** : on pourra ajouter des pièces dans le distributeur une par une (2€, 1€, 50cts, 20cts, 10cts, 5cts, 2cts, 1cts). Quand le montant de pièces atteint le prix de l'élément commandé, le distributeur lance la méthode **preparerCommande**.
- **preparerCommande** : On affiche dans la console "Préparation en cours". Puis un compte à rebours se lancera, allant de 5 à 0 (décrément de 1 par secondes). Arrivé à 0 on affichera "commande prête" le produit sera donc disponible dans la **trappe** et cette dernière sera **ouverte**. On déduira alors 1 du stock de notre élément.
- **seServir** : L'utilisateur récupère le contenu de la trappe si cette dernière est ouverte. Puis on ferme la trappe. Si la trappe est fermée, au bout de trois tentatives, une alerte se déclenche "distributeur verrouillée" et empêche toutes interactions tant que ce dernier est bloqué.
- **deverouiller** : permet de déverrouiller le distributeur.
- **eteindreDistributeur** : cette action supprimera les codes liés aux produits. Puis on affichera un décompte de 3s et enfin on affichera distributeur éteint.
- **check** : Permet d'afficher à tout moment le contenu du distributeur

produits et codes liés, Statut et contenu de la trappe, nombre de produits restant pour chaque élément, s'il est verrouillé ou non, la quantité d'argent que le distributeur a reçu au total.