

Exercice : La gestion de Saint Nicolas et des cadeaux

Saint Nicolas doit non seulement gérer les cadeaux, mais aussi les enfants auxquels ils sont destinés, leurs comportements et les contraintes liées à la distribution. Vous allez implémenter un programme pour répondre à ces besoins.

1. Classes principales

1. **Classe** Cadeau :

- **Attributs :**

- `nom` : le nom du cadeau (string).
- `valeur` : une valeur numérique entre 1 et 10, représentant la rareté ou la qualité du cadeau.
- `stock` : un nombre entier, représentant combien d'unités de ce cadeau sont disponibles.

- **Constructeur** : initialise ces attributs.

- **Propriété et setter :**

- Contrôlez que `valeur` est toujours entre 1 et 10.
- Contrôlez que `stock` ne peut jamais être négatif.

- **Méthodes :**

- `enlever_du_stock(quantite)` : réduit le stock d'un certain nombre. Si le stock est insuffisant, déclenche une erreur.
- `__str__()` : affiche les détails du cadeau (nom, valeur, stock).

2. **Classe** CadeauSpecial (**hérite de** Cadeau) :

- Ajoutez un attribut magique (booléen).
- Redéfinissez `__str__()` pour indiquer si le cadeau est magique.

3. **Classe** Enfant :

- **Attributs** :
 - `nom` : le nom de l'enfant.
 - `age` : l'âge de l'enfant (doit être supérieur à 0 et inférieur à 14).
 - `comportement` : une chaîne parmi gentil, moyen, ou méchant.
- **Constructeur** : initialise ces attributs avec des contrôles sur les valeurs.
- **Propriété et setter** :
 - Le comportement ne peut être que gentil, moyen ou méchant.
- **Méthodes** :
 - `__str__()` : affiche les détails de l'enfant.
 - `est_eligible()` : retourne un booléen. Un enfant est éligible à recevoir des cadeaux uniquement s'il est gentil ou moyen.

4. **Classe** SaintNicolas :

- **Attributs :**
 - liste_cadeaux : une liste de tous les cadeaux disponibles.
 - enfants : une liste d'enfants.
- **Méthodes :**
 - ajouter_cadeau(cadeau) : ajoute un cadeau à la liste de cadeaux.
 - ajouter_enfant(enfant) : ajoute un enfant à la liste.
 - distribuer_cadeaux() : distribue les cadeaux aux enfants selon les règles suivantes :
 - Les enfants méchants ne reçoivent rien.
 - Les enfants gentils reçoivent les cadeaux les plus rares en priorité.
 - Les enfants moyens reçoivent des cadeaux aléatoires parmi ceux restant.
 - Réduit le stock des cadeaux distribués.
 - Affiche une liste des enfants et des cadeaux qu'ils ont reçus.
- **Méthode bonus :**
 - afficher_stock() : affiche tous les cadeaux avec leurs stocks restants.

2. Règles et contraintes

1. Les cadeaux doivent avoir un stock suffisant pour être distribués.
2. Aucun enfant ne peut recevoir plus de 2 cadeaux.
3. Si un enfant est méchant, il doit apparaître dans la liste des enfants non servis.
4. Saint Nicolas doit prioriser les cadeaux rares pour les enfants gentils.

3. Exemple de rendu attendu

Code principal :

```
# Création des cadeaux
ours = Cadeau("Ours en peluche", 8,
stock=5)
train = Cadeau("Train électrique", 9,
stock=3)
livre = CadeauSpecial("Livre magique",
10, stock=2, magique=True)
chocolat = Cadeau("Chocolat", 5,
stock=10)

# Création des enfants
alice = Enfant("Alice", 7, "gentil")
bob = Enfant("Bob", 5, "moyen")
charlie = Enfant("Charlie", 10,
"méchant")
diane = Enfant("Diane", 6, "gentil")
```

```
# Gestion de Saint Nicolas
st_nicolas = SaintNicolas()
st_nicolas.ajouter_cadeau(ours)
st_nicolas.ajouter_cadeau(train)
st_nicolas.ajouter_cadeau(livre)
st_nicolas.ajouter_cadeau(chocolat)

st_nicolas.ajouter_enfant(alice)
st_nicolas.ajouter_enfant(bob)
st_nicolas.ajouter_enfant(charlie)
st_nicolas.ajouter_enfant(diane)

# Distribution des cadeaux
st_nicolas.distribuer_cadeaux()

# Affichage du stock restant
st_nicolas.afficher_stock()
```

Résultat attendu :

Distribution des cadeaux :

- Alice (gentil) : Livre magique, Ours en peluche
- Bob (moyen) : Chocolat, Chocolat
- Charlie (méchant) : Aucun cadeau
- Diane (gentil) : Train électrique, Livre magique

Stock restant :

- Ours en peluche : 4
- Train électrique : 2

- Livre magique : 0
- Chocolat : 8

4. Difficultés ajoutées

1. Les contraintes sur les stocks et les comportements nécessitent des contrôles.
2. La méthode distribuer_cadeaux() introduit des priorités et de la logique conditionnelle.
3. Le système des propriétés et setters force une gestion rigoureuse des données.
4. Le concept d'héritage est appliqué avec CadeauSpecial.

Amusez-vous bien !