

TP Noté - sujet 2

BUT2 SD R3.VCOD.10 - Programmation objet

27 octobre

Consignes générales

- Durée : **2h**
- Pensez bien à vérifier que votre implémentation respecte les principes de la Programmation Orientée Objet et à relire les consignes.
- vous avez le droit au cours et à Internet.
- n'oubliez pas de déposer votre code sur Chamilo (BUT2 SD R3.VCOD.10 Programmation objet → Travaux d'étudiants → Rendu TP noté - sujet 2 (27/10/2023))

Le Chat de Schrödinger

Erwin Schrödinger est un physicien célèbre pour avoir posé les bases de la mécanique quantique. L'expérience dite du '*Chat de Schrödinger*' propose une illustration du principe de superposition quantique, où un atome peut être dans une superposition de deux états équiprobables : intact et désintégré.

Schrödinger a ainsi imaginé une expérience de pensée dans laquelle un chat est enfermé dans une boîte avec un dispositif qui tue le chat dès qu'il détecte la désintégration d'un atome d'un corps radioactif. Si l'atome a une durée de demi-vie de n minutes, alors il a 50% de chances de s'être désintégré au bout de n minutes. La mécanique quantique indique que, tant que l'observation n'est pas faite, l'atome est dans une superposition de deux états équiprobables : intact et désintégré. Or, le mécanisme imaginé par Erwin Schrödinger lie l'état du chat (mort ou vivant) à l'état des particules radioactives, de sorte que le chat serait aussi dans une superposition d'états (l'état mort et l'état vivant), jusqu'à ce que l'ouverture de la boîte (l'observation) déclenche le choix entre les deux états. Par conséquent, il est impossible de dire si le chat est mort ou non au bout de n minutes.



Classes

/5pt

Boîte

1. Créez une classe **Boîte** avec son constructeur.
2. les boîtes ont des dimensions (cm) : *largeur*, *profondeur* et *hauteur*.
3. Chaque boîte peut contenir un *chat* et un ou plusieurs *atome*.

Chat

1. Créez une classe **Chat** avec son constructeur.
2. Chaque chat a un *nom*, un *statut* ('vivant' par défaut) et une *taille* (cm).

Atome

1. Créez une classe **Atomes**, ses sous-classes **Uranium238**, **Potassium40** et **Thorium232** et leurs constructeurs.
2. Chaque atome a une *demi-vie* :
 - l'uranium 238 a une demi-vie de 4.51 milliards d'années.
 - le potassium 40 a une demi-vie de 1.248 milliard d'années.
 - le thorium 232 a une demi-vie de 14.00 milliards d'années.

Méthodes

/6pt

Implémentez les méthodes suivantes. Soyez vigilant quant à la classe dans laquelle ajouter chaque méthode.

1. **empaqueter_chat** : pour mettre un chat dans une boîte. Son état devient alors 'quantique'.
2. **sortir_chat** : pour sortir un chat d'une boîte.
3. **insérer_atome** : pour mettre un atome dans une boîte.
4. **extraire_atome** : pour sortir un atome d'une boîte.
5. **miauler** : afin qu'un chat renvoie la chaîne de caractère '*miaou*'.
6. **proba_desintegre** : recevant en paramètre un nombre d'années (en milliards) écoulées, cette méthode retourne la probabilité qu'un atome se soit désintégré : $\frac{1}{2}$ chances si le nombre d'années écoulées dépasse sa demi-vie. Cette probabilité est cumulable par tranche complète d'années écoulées égales à sa demi-vie (e.g. le potassium 40 a $\frac{1}{2}$ après 1.5 milliards d'années, $\frac{3}{4}$ après 3.0 milliards d'années, etc.). Δ on ne supprime pas réellement l'atome.
7. **ouvrir_boite** : pour regarder dans la boîte et renvoyer ce qui s'y trouve. Cela détermine aussi le statut du chat en utilisant la méthode précédente (tips : utilisez le module `random`).

Afin de déterminer dans quelle classe ajouter ces méthodes, considérez ce que vous désirez représenter (abstraction) et la manière dont ces méthodes interagissent avec les instances d'objets présentes (envois de messages).

Exécution

/7

Créez les instances d'objets d'une boîte, trois chats et cinq atomes (choisis parmi les trois sous-types d'atomes existants). Programmez et exécutez les trois cas suivants :

Cas 1 :

1. mettez un chat et un atome dans une boîte.
2. créez une boucle avec un pas de 500 millions d'années.
3. la boucle se termine au bout de 10 milliards d'années. On regarde alors dans la boîte et on affiche ce que l'on y trouve.
4. lorsque l'expérience s'arrête, videz la boîte et ressuscitez le chat si nécessaire.

Cas 2 : reproduisez (copiez-collez) le **Cas 1**, avec les changements suivants :

1. on ne met plus 1, mais 3 atomes dans la boîte (Δ cela influencera le destin du chat).
2. à chaque pas de la boucle, un chat dans la boîte a 10% de chances de miauler. Lorsque cela arrive, on ouvre la boîte. Si le chat est vivant, l'expérience continue. Sinon on stop l'expérience.

Cas 3 : reproduisez (copiez-collez) le **Cas 1**, avec les changements suivants :

1. à chaque itération, tous les chats 'vivant' ou 'quantique' (dans et hors de la boîte) ont 10% de chances de miauler.
2. quand un chat dans la boîte miaule, on ouvre la boîte. Si il est vivant, on rajoute un atome.
3. lorsque un chat miaule en dehors de la boîte, on ouvre la boîte et on échange ce chat avec celui de la boîte.
4. cette fois l'expérimentation se termine lorsque aucuns chats n'a miaulé pendant 2.5 milliards d'années.

Pensez à mettre chaque cas dans une fonction (afin de pouvoir les exécuter indépendamment) et à afficher des messages permettant de suivre les étapes importantes de l'exécution.

Ajustements

/4pt

Modifiez votre code pour :

- vérifier que la taille d'un chat ne dépasse pas l'une des dimensions de la boîte avant de l'y mettre (dans ce cas, on essaie de mettre un autre chat).
- un chat qui miaule dans une boîte ne fait pas le même son qu'un chat qui miaule à l'extérieur.
- créez une seconde version de la méthode permettant de retourner la probabilité qu'un atome se soit désintégré selon la formule suivante :

$$\lambda = \frac{\ln(2)}{t_{1/2} \div N}$$

- avec λ la probabilité qu'un atome se soit désintégré, $\ln(2)$ le logarithme népérien de 2 (tips : `import math ; math.log(2)`), $t_{1/2}$ la valeur de demi-vie d'un atome (en milliards) et N le nombre d'années écoulées (en milliards).
- commentez votre code.