

## UE Conception Orientée Objet

---

### TD Le facteur sonne toujours deux fois

On se propose de modéliser la distribution du courrier : chaque jour, des personnes envoient des courriers à d'autres personnes, habitant la même ville ou une autre ville.

Elles mettent ces courriers dans des boîtes aux lettres qui sont relevées, et le courrier est distribué le lendemain.

## 1 Les villes et les habitants

Dans cet exercice, les **villes** sont considérées comme des bureaux distributeurs : elles réunissent toutes les lettres postées le jour même dans la ville, et les distribuent le lendemain. Il convient donc, comme dans la réalité, de distinguer la boîte aux lettres, où sont postées les courriers, de la sacoche du facteur, qui contient les courriers à distribuer.

Une ville possède un nom, des rues et des **habitants**.

- On peut ajouter un habitant. On précise alors son adresse (on ne fera pas de vérification sur la validité de l'adresse ou les éventuels conflits d'adresse).
- La méthode `posteLettre(Courrier c)` ajoute un courrier à l'ensemble des courriers à distribuer (dans la boîte aux lettres).
- La méthode `distribueCourrier()` s'occupe de la distribution du courrier (posté la veille et qui est, au moment de la distribution, dans la sacoche du facteur).

Les habitants sont des personnes. Une personne est caractérisée par un nom, une adresse (une rue, un numéro, une ville) et un compte en banque.

On doit pouvoir créditer ou débiter ce compte d'un montant donné.

Des diagrammes UML indicatifs et partiels sont proposés en annexe.

## 2 Le courrier

Un **courrier** comporte un expéditeur et un destinataire (deux personnes) et un contenu (a priori, n'importe quoi).

A tout courrier est associée une action, qui sera accomplie lorsque ce courrier aura été distribué.

De plus tout courrier a un coût.

Un courrier peut être :

- Une **lettre simple** : son contenu est un texte. Le coût d'une lettre simple est fixe (0,5 par exemple).
- Une **lettre de change** : elle contient une certaine somme d'argent, qui provient du compte de l'expéditeur, et sera versée au compte du destinataire. Le destinataire, reconnaissant, envoie alors une lettre de remerciement à son bienfaiteur. Le coût d'une lettre de change est de  $1 + 1\%$  de la somme transférée.

De plus il est possible d'envoyer tout courrier **en recommandé**. Dans ce cas, en plus de l'effet lié au courrier, un accusé de réception est alors envoyé l'expéditeur en retour. Un recommandé a un surcoût de 15% par rapport au courrier initial.

**Q 1 .** Modélisez les différents éléments décrits ci-dessus sous la forme de diagramme de classes UML. Donnez en parallèle le code des méthodes `distribueCourrier` de `Ville` et `recoitCourrier` et `envoieCourrier` de `Habitant`

**Q 2 .** On ajoute maintenant la prise en compte de **courriers urgents**. Tout courrier peut être "transformé" en courrier urgent (y compris les recommandés). L'action est inchangée et le coût est doublé. Comment prendre en compte cette modification ? Faites une proposition qui s'intègre à la précédente.

**Q 3 .** Chacun connaît les chaînes de courrier censées vous rendre millionnaire. Le fonctionnement est le suivant : vous recevez une lettre contenant l'adresse de quatre personnes (les adresses sont ordonnées) et la lettre vous demande de :

- Envoyer 1 euro à chacun des noms de la liste.
- Supprimer le premier nom et mettre votre nom en quatrième position.
- Envoyer la lettre (avec la nouvelle liste de noms) à 10 personnes, en leur expliquant le processus.

En principe, vous devez tôt ou tard devenir riche.

Faites une proposition de conception pour prendre en compte ces nouveaux courriers. Intégrez là dans les propositions précédentes.

**Q 4 .** *Simulation (tests)* : Créez une ville  $V$  et ses habitants (vous gèrerez aléatoirement les adresses, pour simplifier on considèrera que tous les habitants de la simulation habitent la ville  $V$ ). Chaque jour, pendant  $k$  jours,  $n_k$  habitants envoient un courrier à un autre habitant de la ville, choisi aléatoirement. Vous ferez varier les types des courriers envoyés. Eventuellement, ces courriers peuvent engendrer des réponses (accusés de réception, lettres de remerciement, etc.).

Simulez la collecte et la distribution du courrier, comme sur l'exemple donné en annexe (les deux dernières distributions de courrier se passent aux jours 4 et 5, et se contentent d'écouler les lettres non encore distribuées).

NB : Le caractère "urgent" des courriers n'est pas pris en compte dans la simulation, urgent ou non un courrier arrive à destination le lendemain du jour où il est posté, simplement un courrier urgent coûte le double...

## Annexes

Diagrammes donnés à titre **indicatif** :

Habitant	Adresse	Ville
- soldeCompte : float - adresse : Adresse ...	-numero :int -rue :Rue -ville :Ville	- boiteALettres : List<Courrier> ...
... +getAdresse() :Adresse +recoitCourrier(courrier :Courrier) +envoieCourrier(courrier :Courrier) +debiter(montant :float) +crediter(montant :float)	... +getNumero() :int +getRue() :Rue +getVille() :Ville	... +posteLettre(c :Courrier) +distribueCourrier() +courrierADistribuer() :boolean +randomHabitant() :Habitant*

\* : méthode proposée uniquement pour faciliter la mise en place de la simulation.

## Exemple de trace d'exécution d'une simulation

---

Jour 0

```
>>> hab96 envoie courrier0 de change recommande (cout :1.84) a hab33
>>> hab13 envoie courrier2 (cout :1.0) a hab78
>>> hab56 envoie courrier3 recommande (cout :1.15) a hab2
>>> hab43 envoie courrier5 URGENT (cout :2.0) a hab2
```

---

Jour 1

```
courrier0 de change (cout :1.6) de [valeur = 10.0 euros] envoye par hab96 reçu par hab33
>>> hab33 envoie courrier7 de remerciement (cout :1.0) a hab96
>>> hab33 envoie courrier8 accuse de reception (cout :1.0) a hab96
courrier2 (cout :1.0)[bla bla] envoye par hab13 reçu par hab78
courrier3 (cout :1.0)[bla bla] envoye par hab56 reçu par hab2
>>> hab2 envoie courrier9 accuse de reception (cout :1.0) a hab56
courrier5 (cout :1.0)[bla bla] envoye par hab43 reçu par hab2
>>> hab1 envoie courrier10 URGENT (cout :3.2) a hab13
>>> hab35 envoie courrier12 (cout :1.0) a hab69
>>> hab85 envoie courrier13 de change recommande URGENT (cout :3.68) a hab83
>>> hab3 envoie courrier15 de change (cout :1.6) a hab2
```

---

Jour 2

```
courrier7 de remerciement (cout :1.0)[merci pour courrier0 de change] envoye par hab33 reçu par hab96
courrier8 accuse de reception (cout :1.0)[accuse du courrier0 de change recommande] envoye par hab33 reçu par hab96
courrier9 accuse de reception (cout :1.0)[accuse du courrier3 recommande] envoye par hab2 reçu par hab56
courrier10 de change (cout :1.6) de [valeur = 10.0 euros] envoye par hab1 reçu par hab13
>>> hab13 envoie courrier16 de remerciement (cout :1.0) a hab1
courrier12 (cout :1.0)[bla bla] envoye par hab35 reçu par hab69
courrier13 de change (cout :1.6) de [valeur = 10.0 euros] envoye par hab85 reçu par hab83
>>> hab83 envoie courrier17 de remerciement (cout :1.0) a hab85
>>> hab83 envoie courrier18 accuse de reception (cout :1.0) a hab85
courrier15 de change (cout :1.6) de [valeur = 10.0 euros] envoye par hab3 reçu par hab2
>>> hab2 envoie courrier19 de remerciement (cout :1.0) a hab3
>>> hab95 envoie courrier20 de change (cout :1.6) a hab6
>>> hab52 envoie courrier23 URGENT (cout :2.0) a hab1
>>> hab46 envoie courrier25 URGENT (cout :2.0) a hab91
>>> hab60 envoie courrier26 (cout :1.0) a hab24
```

---

Jour 3

```
courrier16 de remerciement (cout :1.0)[merci pour courrier10 de change] envoye par hab13 reçu par hab1
courrier17 de remerciement (cout :1.0)[merci pour courrier13 de change] envoye par hab83 reçu par hab85
courrier18 accuse de reception (cout :1.0)[accuse du courrier13 de change recommande] envoye par hab83 reçu par hab85
courrier19 de remerciement (cout :1.0)[merci pour courrier15 de change] envoye par hab2 reçu par hab3
courrier20 de change (cout :1.6) de [valeur = 10.0 euros] envoye par hab95 reçu par hab6
>>> hab6 envoie courrier27 de remerciement (cout :1.0) a hab95
courrier21 (cout :1.0)[bla bla] envoye par hab52 reçu par hab1
>>> hab1 envoie courrier28 accuse de reception (cout :1.0) a hab52
courrier23 (cout :1.0)[bla bla] envoye par hab46 reçu par hab91
courrier25 (cout :1.0)[bla bla] envoye par hab60 reçu par hab24
>>> hab23 envoie courrier29 (cout :1.0) a hab25
>>> hab55 envoie courrier30 de change recommande (cout :1.84) a hab44
>>> hab98 envoie courrier32 (cout :1.0) a hab33
>>> hab14 envoie courrier33 URGENT recommande (cout :2.30) a hab53
```

---

Jour 4

```
courrier27 de remerciement (cout :1.0)[merci pour courrier20 de change] envoye par hab6 reçu par hab95
courrier28 accuse de reception (cout :1.0)[accuse du courrier21 recommande] envoye par hab1 reçu par hab52
courrier29 (cout :1.0)[bla bla] envoye par hab23 reçu par hab25
courrier30 de change (cout :1.6) de [valeur = 10.0 euros] envoye par hab55 reçu par hab44
>>> hab44 envoie courrier35 de remerciement (cout :1.0) a hab55
>>> hab44 envoie courrier36 accuse de reception (cout :1.0) a hab55
courrier32 (cout :1.0)[bla bla] envoye par hab98 reçu par hab33
courrier33 URGENT (cout :2.0)[bla bla] envoye par hab14 reçu par hab53
>>> hab53 envoie courrier37 accuse de reception (cout :1.0) a hab14
```

---

Jour 5

```
courrier35 de remerciement (cout :1.0)[merci pour courrier30 de change] envoye par hab44 reçu par hab55
courrier36 accuse de reception (cout :1.0)[accuse du courrier30 de change recommande] envoye par hab44 reçu par hab55
courrier37 accuse de reception (cout :1.0)[accuse du courrier33 recommande] envoye par hab53 reçu par hab14
```