UNIVERSITE DE CORSE - PASQUALE PAOLI

Faculté des Sciences et Techniques

Diplôme : **Licence SPI 3ème année**	2021-2022

UE : Ateliers de programmation

Programmation Orientée Objet

Atelier 3(bis) : classes abstraites - polymorphisme

- classes attributs méthodes
- attributs et méthodes statiques
- hiérarchie d'héritage
- classes abstraites, polymorphisme

Enseignants : Paul-Antoine BISGAMBIGLIA, Marie-Laure NIVET, Evelyne VITTORI

Agence immobilière

On souhaite développer une application très simplifiée de gestion des appartements d'une agence immobilière. L'agence gère une liste d'appartements qui peuvent être des studios ou des appartements comportant plusieurs pièces (multiPièces). Les loyers des appartements sont établis de manière forfaitaire. Les studios ont tous un loyer égal à 300 euros et les multiPièces ont un loyer égal à 200 euros multiplié par leur nombre de pièces. Le logiciel doit permettre d'enregistrer des appartements dans l'agence, d'afficher la liste des appartements gérés par l'agence caractéristiques et d'afficher le total des loyers à percevoir par l'agence.

Pour chacune des questions, vous prendrez soin d'écrire un programme de test de chaque classe sans attendre la question 6.

Question 1 : Classe Proprietaire Définissez la programmation Java d'une classe Propriétaire possédant deux attributs nom et adresse de type String et constructeur à deux paramètres nom et adresse.

Définissez les méthodes get et set necessaires pour permettre uniquement d'accéder à l'attribut nom et de modifier l'attribut adresse. Redéfinissez la méthode toString afin qu'elle renvoie une chaîne de la forme suivante :

nom (*adresse*) (exemple : Toto(Corte))

Question 2 : Classe Appartement

Définissez la programmation Java d'une classe abstraite Appartement possédant les caractéristiques suivantes :

- Un attribut *nbAppartements* représentant le nombre total d'appartements créés.
- Un attribut *proprietaire* de type Proprietaire représentant le propriétaire de l'appartement.
- Un attribut *code* de la forme APP suivi d'un numéro correspondant au numéro d'ordre de création de l'appartement (par exemple : APP1 pour le premier appartement, APP2 , ..ect…). Il est généré automatiquement lors de la création d'un appartement.

- Un constructeur à un paramètre de type Propriétaire.
- Une méthode abstraite loyer renvoyant un double.
- Une méthode toString (redéfinition) renvoyant une chaîne de la forme suivante :
  code appartenant à *propriétaire*
  (exemple : APP1 appartenant à Toto (Corte))

Question 3 : Classe Studio

Définissez la programmation Java d'une classe Studio héritant de la classe Appartement et respectant les indications données ci-dessous :

- LOYERSTUDIO est une constante égale à 300 euros. Elle représente le loyer mensuel d'un studio. On suppose en effet que tous les studios ont le même loyer égal à 300 euros.
- Le constructeur de la classe Studio possède un seul paramètre de type Proprietaire.
- La méthode loyer() renvoie simplement la valeur de la constante LOYERSTUDIO.
- La méthode toString() de la classe Studio doit avoir pour résultat une chaîne de la forme suivante :
  Studio APP1 appartenant à Toto (Corte)

Question 4 : Classe MultiPiece

Définissez la programmation Java de la classe MultiPiece héritant de la classe Appartement et respectant les indications données ci-dessous :

- LOYERPIECE est une constante égale à 200 euros. Elle représente le loyer mensuel relatif à un appartement d'une seule pièce. Un appartement de 2 pièces aura un loyer de 2x200 soit 400 euros, un appartement de 3 pièces aura un loyer de 3x200, ….
- La méthode loyer() renvoie la valeur de la constante LOYERPIECE multipliée par le nombre de pièces de l'appartement.
- Le constructeur de la classe Studio possède un seul paramètre de type Proprietaire.
- La méthode toString() doit avoir pour résultat une chaîne de la forme suivante :
  2 pièces APP1 appartenant à Toto (Corte)

Question 5 : Classe Agence

Définissez la programmation Java d'une classe Agence contenant trois attributs définis comme suit :

- Nom de type String
- nbApp de type int représentant le nombre d'appartements gérés par l'agence et présents dans la liste appartements. Cet attribut sera initialisé à 0 dans le constructeur.
- appartements de type ArrayList<Appartement> représentant la liste des appartements de l'agence ***(cf cours POO Atelier2 p103)***.

Le constructeur de la classe comportera un seul paramètre nom. Il devra initialiser la liste appartements.

Cette classe comporte trois méthodes :

- La méthode **ajoutAppartement**(a : Appartement) ajoute l'appartement « a » à la liste des appartements de l'agence.
- La méthode **totalLoyer()** a pour résultat la somme des loyers des appartements gérés par l'agence. Cette méthode doit être private.
- La méthode **afficher()** affiche à l'écran les caractéristiques de l'agence, la liste de ses appartements et le total des loyers de l'agence sous la forme suivante :

```
AGENCE ESPADONBLEU – 3 appartements
LISTE DES APPARTEMENTS
Studio APP1 appartenant à Toto (Corte)
2 pièces APP2 appartenant à Toto (Corte)
4 pièces APP3 appartenant à Titi (Ajaccio)
Total des loyers de l'agence 1500.0 euros
```

- La méthode **afficher**() invoque la méthode totalLoyer() pour effectuer le total des loyers.

Question 6 : Classe GestionAgence

Définissez la programmation Java de la classe **GestionAgence** comportant une méthode main réalisant les actions suivantes :

- Créer un propriétaire ayant pour nom "Toto" et pour adresse « Corte »
- Créer un propriétaire ayant pour nom "Titi" et pour adresse « Ajaccio »
- Créer une agence ayant pour nom ESPADONBLEU
- Créer un studio appartenant au propriétaire Toto et l'ajouter à la liste des appartements de l'agence
- Créer un appartement de 2 pièces appartenant au propriétaire Toto et l'ajouter à la liste des appartements de l'agence
- Créer un appartement de 4 pièces appartenant au propriétaire Titi et l'ajouter à la liste des appartements de l'agence
- Afficher les caractéristiques de l'agence et la liste de ses appartements.