

L'objectif de ce TP est de créer des classes permettant de représenter des maisons et des villas (vues comme des maisons avec jardin), et de calculer leur prix de vente, en supposant que ceux-ci sont calculés en fonction d'un prix « de base », par unité de surface. De façon générale, on considérera que le prix de base est celui par unité de surface intérieure, tandis que chaque unité de surface extérieure (de jardin) vaut la moitié de ce prix.

Toutes les classes sont à écrire dans un *package* nommé **houses**. Un main complet de test est donné par le code suivant :

```
import housestests.HouseTests;
import housestests.SimulatorTests;
import housestests.VillaTests;
[...]
boolean ok = true;
HouseTests houseTester = new HouseTests();
ok = ok && houseTester.testGetAddress();
ok = ok && houseTester.testPrice();
VillaTests villaTester = new VillaTests();
ok = ok && villaTester.testGetAddress();
ok = ok && villaTester.testPrice();
SimulatorTests simulatorTester = new SimulatorTests();
ok = ok && simulatorTester.testPriceOneHouse();
ok = ok && simulatorTester.testPriceSet();
System.out.println(ok ? "All tests OK" : "At least one test KO");
```

1 Maisons et villas

Exercice 1. *Écrire une classe nommée **House** permettant de représenter des maisons, avec un constructeur prenant en arguments, dans cet ordre, une adresse (chaîne de caractères) et une surface intérieure (int); les maisons ainsi représentées seront des maisons sans jardin.*

Exercice 2. *Ajouter à la classe un accesseur **getAddress**.*

Exercice 3. *Ajouter à la classe une méthode **price** prenant un argument de type float, représentant le prix par unité de surface intérieure, et retournant le prix de la maison (comme une valeur de type float).*

Exercice 4. *Redéfinir la méthode **String toString()** de la classe **Object** dans la classe **House**.*

Exercice 5. *Écrire une classe exécutable, dont le main instancie une maison et un prix de base, et affiche la maison ainsi que son prix.*

Exercice 6. *Écrire une classe nommée **Villa**, héritant de la classe **House**, et dont le constructeur prend en arguments, dans cet ordre, une adresse, une surface intérieure, et une surface extérieure (de types respectifs **String**, **int**, **int**).*

Exercice 7. Redéfinir les méthodes `price` et `toString` dans cette sous-classe.

Exercice 8. Compléter la classe exécutable de l'exercice 5 avec l'instanciation d'une villa, son affichage ainsi que celui de son prix.

2 Simulateur de prix

On souhaite désormais écrire un simulateur, permettant de calculer le prix d'un ou plusieurs biens, avec ou sans jardin (par exemple, le prix total d'un lotissement ou d'une rue). Pour la classe qui suit, on change donc de perspective :

- dans les classes `House` et `Villa`, on prenait le point de vue du propriétaire d'un bien, qui pouvait estimer le prix dans différents scénarios (c'est-à-dire différents prix au mètre carré) en utilisant la méthode `price`,
- dans la classe qui suit, on représente au contraire le point de vue d'une agence immobilière, qui connaît les prix au mètre carré, et qui les utilise pour estimer différents biens ou ensembles de biens.

Exercice 9. Écrire une classe nommée `Simulator`, dont le constructeur prend un argument de type `float`, représentant le prix de base (par unité de surface) à prendre en compte pour les simulations.

Exercice 10. Ajouter à la classe une méthode `price` prenant en argument un objet de type `House`, et retournant son prix.

Exercice 11. Vérifier, en complétant la classe exécutable de l'exercice 5, que la méthode `price` peut prendre en argument une instance directe de la classe `House` ou une instance de la classe `Villa`, et qu'elle retourne la bonne valeur dans tous les cas.

Exercice 12. Ajouter à la classe une méthode `price` prenant cette fois en paramètre un ensemble d'objets de type `House` (via un paramètre de type `HashSet<House>`), et retournant la somme des prix de ces maisons/propriétés (il s'agit donc d'une surcharge de la méthode `price` précédente).

Exercice 13. Compléter la classe exécutable de l'exercice 5 avec la création d'un tableau contenant plusieurs maisons et propriétés, et vérifier le prix calculer.

Exercice 14. Tester l'ensemble des méthodes écrites en utilisant le code fourni au début de cet énoncé.