Découvrir la programmation orientée objet avec Java

Travaux pratiques

frederic.rallo@univ-cotedazur.fr

TP3 : Héritage

L'objectif des exercices ci-dessous est de mettre en œuvre les tableaux, collections et héritage en Java.

On considère le TP 2 terminé pour réaliser les exercices suivants :

Exercice 1

Dans Eclipse, organisez projet de la façon suivante :

nom du projet ➔ TP3

+ créez un package exercice1 et placez-y vos classes et la classe qui teste Ville (TestVille.java)

Écrire une classe **Ville** suivant les spécifications suivantes :

- Une **Ville** est définie par son nom et un nombre d'habitants.
- Le nom de la ville est obligatoire lors de la construction d'une **Ville**. Ce nom ne peut pas changer.
- La connaissance du nombre d'habitants est optionnelle lors de la construction d'une **Ville** (valeur par défaut = 0). Ce nombre peut être modifié après la construction.
- La création d'une **Ville** sans paramètres entraîne l'affichage d'un message d'erreur puis l'application se termine (Vous pourrez utiliser la méthode *exit(code)* de la classe *System*).
- Une **Ville** peut retourner une chaine destinée à être affichée. Cette chaine contient la valeur le type de l'objet (Ville) et ses attributs via une méthode description().
- Créez un tableau (pas une collection) de 5 villes en testant tous les cas
- Faites en sorte que la méthode description() soit la même pour les exercices 1,2 et 3

Découvrir la programmation orientée objet avec Java

Travaux pratiques

frederic.rallo@univ-cotedazur.fr

```
* ------------------------------------------------------- *
* TP java                                                       *
*                                                         *
* @author frédéric rallo - frederic.rallo@univ-cotedazur.fr *
* @version TD3 - ex1                                     *
* ------------------------------------------------------- *
Au départ...
Les Villes sont :

création d'une ville avec nom
Les Villes sont :
- Vence est une Ville

création d'une ville avec nom et nb d'ahabitants
Les Villes sont :
- Vence est une Ville
- Cagnes sur Mer est une Ville de 453 habitants

création d'une ville sans nom
Erreur, le nom est obligatoire ! L'application doit quitter !!!

                          *
```

Exercice 2

+ créez un package exercice2 et placez-y vos classes et la classe qui teste (TestCapitale.java)

+ copiez votre classe Ville.java dans exercice2

- Écrire une classe **Capitale** suivant les spécifications suivantes :
- Une **Capitale** est une **Ville** définie par un nom de pays.
- Une **Capitale** doit être construite avec un nom de ville et un nom de pays OU un nom de ville, un nom de pays et un nombre d'habitant.
- Tentez de deviner ce que donne l'instruction : Capitale c = new Capitale(« Paris », « France »); puis vérifiez.
- Tentez de deviner ce que donne l'instruction: Capitale c = new Capitale(); puis vérifiez
- Tentez de deviner ce que donne l'instruction : Capitale c = new Ville(« Paris »); puis vérifiez
- Tentez de deviner ce que donne l'instruction : Ville v = new Capitale(« Paris », « France »); puis vérifiez

Découvrir la programmation orientée objet avec Java

Travaux pratiques

frederic.rallo@univ-cotedazur.fr

```
* ------------------------------------------------------- *
* TP java – SI3                                           *
*                                                         *
* @author frédéric rallo - frederic.rallo@univ-cotedazur.fr *
* @version TD3   - ex2                                    *
* ------------------------------------------------------- *
création de Nice...
Nice est une Ville de 1099 habitants

création de Paris...
Paris est une Capitale de 5555 habitants. C'est la capitale du pays FRANCE

création de Londres...
Londres est une Capitale. C'est la capitale du pays ANGLETERRE
```

Découvrir la programmation orientée objet avec Java

Travaux pratiques

frederic.rallo@univ-cotedazur.fr

Exercice 3

+ créez un package exercice3 et placez-y vos classes et la classe qui teste (TestPatelin.java)

+ copiez votre classe Ville.java dans exercice2

- Un **Patelin** est une **Ville** ne pouvant pas avoir plus de 1000 habitants.

```
* ------------------------------------------------------------ *
* TP java - SI3                                                *
* @author frédéric rallo - frederic.rallo@univ-cotedazur.fr *
* @version TD3   - ex3                                         *
* ------------------------------------------------------------ *
création de Nice...
Nice est une Ville de 1099 habitants


création de Séranon...
Séranon est un Patelin


création de Ovignon...
Ovignon est un Patelin de 344 habitants


création de Lyon...
Erreur, le nb d'habitants est trop gros pour un Patelin ! L'application doit quitter !!!
```

Découvrir la programmation orientée objet avec Java

Travaux pratiques

frederic.rallo@univ-cotedazur.fr

Exercice 4

+ créez un package exercice4 et placez-y vos classes et la classe qui teste (Main.java)

+ copiez vos classes dans exercice4

- Prédites ce qui ne marchera pas et expliquez pourquoi.
- Vérifiez vos prédictions

```
   //-- 1er jeu de tests
Capitale c1 = new Capitale("Paris", "France");
Ville v11 = (Ville) c1;
Ville v12 = c1;
Patelin p1 = (Patelin) c1;
System.out.println ("\t- " + p1.getNom() + " " + p1.description());
System.out.println ("\t- " + c1.getNom() + " " + c1.description());
System.out.println ("\t- " + v11.getNom() + " " + v11.description());
System.out.println ("\t- " + v12.getNom() + " " + v12.description());


//-- 2ème jeu de tests
Ville v21 = new Ville("Marseille");
Capitale c2 = (Capitale) v21;
Ville v22 = new Patelin("Marseille");
Patelin p2 = (Patelin) v22;
System.out.println ("\t- " + c2.getNom() + " " + c2.description());
System.out.println ("\t- " + v21.getNom() + " " + v21.description());
System.out.println ("\t- " + v22.getNom() + " " + v22.description());
System.out.println ("\t- " + p2.getNom() + " " + p2.description());
```

- Commentez les lignes qui posent problème et trouvez la trace suivante

```
* ------------------------------------------------------- *
* TP java                                                 *
*                                                         *
* @author frédéric rallo - frederic.rallo@univ-cotedazur.fr *
* @version TD3 - ex4                                      *
* ------------------------------------------------------- *

1er jeu de tests
-----------------------------------------------------------
- Paris est une Capitale. C'est la capitale du pays France
- Paris est une Capitale. C'est la capitale du pays France
- Paris est une Capitale. C'est la capitale du pays France

2ème jeu de tests
-----------------------------------------------------------
- Marseille est une Ville
- Marseille est un Patelin
- Marseille est un Patelin
```

Découvrir la programmation orientée objet avec Java

Travaux pratiques

frederic.rallo@univ-cotedazur.fr

Exercice 5

Les villes sont classées en quatre catégories :

- A : Celles qui ont entre un et 500 000 habitants.
- B : Celles qui ont plus de 500 000 habitants.
- ? : Celles dont le nombre d'habitant est nul ou inconnu;
- C : Celles qui sont capitales quel que soit le nombre d'habitants.

Écrire dans la classe **Ville** la méthode *char categorie()* qui renvoie la catégorie de l'objet courant. Surcharger au besoin cette méthode dans les classes **Capitale** et **Patelin** (nb : on souhaite toujours taper le moins de code possible).

Écrire une classe **TestVille** qui construise un tableau de **Ville**, remplit ce tableau avec 2 **Capitale**, 2 **Ville** et 2 **Patelin** puis affiche le nombre de ville de chaque catégorie.

Vérifiez bien que chaque nouvelle classe produite vérifie les contraintes du type de classe (contraintes sur la présence d'un pays, nombre d'habitants,...).

```
* ------------------------------------------------------------ *
* TP java - SI3                                                *
* @author frédéric rallo - frederic.rallo@univ-cotedazur.fr   *
* @version TD3   - ex4                                         *
* ------------------------------------------------------------ *
```

voici les villes :

- Paris est une Capitale de 558889 habitants. C'est la capitale du pays France. La catégorie est : C
- Londres est une Capitale. C'est la capitale du pays Angleterre. La catégorie est : C
- Nice est une Ville de 354000 habitants. La catégorie est : A
- Cagnes-sur-Mer est une Ville de 154000 habitants. La catégorie est : A
- Séranon est un Patelin de 400 habitants. La catégorie est : A
- Ovignon est un Patelin. La catégorie est : ?

Découvrir la programmation orientée objet avec Java

Travaux pratiques

frederic.rallo@univ-cotedazur.fr

Exercice 6

Les restaurants sont caractérisés par un nom et une année de création. Les restaurants « récents » sont des restaurants de moins de 3 ans. « chez Fred » est un restaurant créé en 1970, « La cantine » est un restaurant créé en 1972, « La Jarrerie » est un restaurant créé en 2017 et « Le miam » est un restaurant créé en 2019.

1. Écrire la classe Restaurant.java avec les attributs, accesseur(s) et constructeur(s) en respectant les bonnes pratiques du java

2. Écrire la méthode toString() dans la classe Restaurant.java qui indiquera le nom et l'année de création.

3. Écrire une classe TestRestaurant.java disposant d'un point d'entrée du programme
   - Créer les personnes « chez Fred », « La cantine », « La Jarrerie » et « Le miam »
   - Afficher la liste des restaurants créés
   - Afficher un astérisque devant le nom de tous les restaurants récents

```
* ------------------------------------------------------- *
* TP java                                                 *
*                                                         *
* @author frédéric rallo - frederic.rallo@univ-cotedazur.fr *
* @version TD3 - ex6                                      *
* ------------------------------------------------------- *
"Chez Fred" (1970)
"La cantine" (1972)
*"La jarrerie" (2017)
*"Le miam" (2019)
```

Exercice 7

Les menus sont caractérisés par leur nom, un créateur (qui est un Restaurant non récent), des distributeurs (qui sont des Restaurants). Parmi les distributeurs on peut y retrouver le restaurant créateur (ou pas !). Les distributeurs de Menus seront décrits dans une collection

1. Écrire la classe Menu.java avec les attributs, accesseur(s) et constructeur(s) en respectant les bonnes pratiques du java

2. Écrire la méthode toString() dans la classe Menu.java qui indiquera son créateur, le nom du menu et l'année de création de tous les restaurants qui le distribuent.

3. Écrire une classe TestMenu.java disposant d'un point d'entrée du programme
   - Créer le menu « Rapid » inventé par chez fred
   - Créer le menu « Prestige » inventé par la jarrerie
   - Les restaurants « La cantine » et « Le miam » distribuent le menu rapid

Découvrir la programmation orientée objet avec Java

Travaux pratiques

frederic.rallo@univ-cotedazur.fr

```
* ------------------------------------------------------- *
* TP java                                                 *
*                                                         *
* @author frédéric rallo - frederic.rallo@univ-cotedazur.fr *
* @version TD3 - ex7                                      *
* ------------------------------------------------------- *
Menu "Prestige", created by "Jarrerie" (2017) is deal by []
Menu "Rapid", created by "Chez Fred" (1970) is deal by ["La cantine" (1972), "Le miam"
(2019)]
```

Exercice 8

Il existe une base de données dans laquelle sont répertoriés tous les Restaurants et tous les Menus. Cette base de données permet d'ajouter un Restaurant, de le Supprimer, d'ajouter un Menu et de le supprimer

1. Écrire la classe Database.java

2. Précisez en commentaire de la javadoc le type de relation avec les autres objets

3. Écrire la méthode getRecentRestaurants() qui retourne la liste de tous les Restaurants récents

4. Écrire la méthode getProspects(Menu menu) qui retournera la liste des Restaurants qui n'affichent pas le menu passé en paramètre

5. Tester la base de données en créant :
   - le Restaurant « chez Fred » (1970)
   - le Restaurant « la cantine» (1972)
   - le Restaurant « la jarrerie » (2017)
   - le Restaurant « le miam » (2019)
   - le Menu « rapide» distribué chez « la cantine » et « le miam »

   → Affichez les prospects pour le menu « rapide »

```
* ------------------------------------------------------- *
* TP java                                                 *
*                                                         *
* @author frédéric rallo - frederic.rallo@univ-cotedazur.fr *
* @version TD3 - ex8                                      *
* ------------------------------------------------------- *
["Chez Fred" (1970), "La jarrerie" (2017)]
```