

Exercice 1 : PooEx1Object [Ne passez pas à l'exo suivant avant d'avoir fini le précédent]

- **Exercice 1.1 :** Ajouter une première classe City censée représenter une ville avec pour attributs nom de la ville, pays, nombre d'habitants, définir au moins 1 constructeur. Puis ajouter une classe pour tester (TestCity) avec un main pour instancier plusieurs villes comme par ex : `City toulouse = new City("Toulouse", "France", 450000);` répétez donc l'instanciation de votre classe avec plusieurs villes puis afficher les attributs de chaque ville dans la suite de votre programme principale. Changer l'attribut nbInhabitants de l'objet toulouse en lui ajoutant 20000 habitants puis afficher toulouse à nouveau pour vérifier que les modifications ont bien été prises en compte.
- **Exercice 1.2 :** Toujours dans votre classe City, mettez en place vos accesseurs (getter et setter) et afficher à nouveau vos villes cette fois ci via les accesseurs. A priori, l'utilisateur ne peut plus changer les attributs d'une ville sans votre approbation, dorénavant : `toulouse.nbInhabitants = -200` est impossible, cependant, il peut changer l'attribut via l'accesseur aussi veiller à trouver une solution ici. Tester cette fonctionnalité à la suite de votre programme principale (mettez entre commentaire les travaux précédents). Ajouter cette protection dans votre constructeur. Prenez la peine d'afficher des messages à l'utilisateur qui doit comprendre son erreur. Enfin, que faire si l'utilisateur veut réduire la population passant par ex la ville de Toulouse de 470000 à 370000 ?
- **Exercice 1.3 :** Ajouter un nouveau constructeur qui accepte 2 paramètres le nom de la ville et le nombre d'habitants puis tester le en affichant les 3 attributs. Ensuite, modifier l'attribut inconnu en passant par les accesseurs, afficher la ville à nouveau, soigner l'affichage pour obtenir ce
résultat :

<code>[ville : Rabat] [pays : unknown] [nombre d'habitants : 577000]</code>
<code>[ville : Rabat] [pays : Maroc] [nombre d'habitants : 577000]</code>
- **Exercice 1.4 :** crée une méthode pour afficher les attributs d'une ville tel que `toulouse.display();` aura pour résultat : « ville de Toulouse en France ayant 450000 »
- **Exercice 1.5 :** que se passe t-il ici ? `System.out.println(toulouse);` Pourquoi ?
- **Exercice 1.6 :** Ajouter une méthode toString() qui renvoi une chaîne de caractère constituée des attributs d'un objet ville, afin d'obtenir le résultat suivant, par l'appel implicite de la méthode toString de l'objet toulouse :
 - `System.out.println(toulouse);` : « ville de Toulouse en France ayant 450000 habitants »
- **Exercice 1.7 :** Trouver un moyen de compter le nombre d'instance de votre classe City, c'est à dire le nombre d'objets/villes instanciés dans votre programme.
`City paris = new City("Paris", "France", 2000000);`
`City nice = new City("Nice", "France", 340000);`

- **Exercice 1.8 :** Crée la classe Person avec pour attributs nom, prénom, âge, adresse. Puis tester la dans une classe TestPerson en créant plusieurs instances puis afficher les afin d'obtenir le résultat suivant :

```

5
6 public class TestPerson {
7     public static void main(String[] args) {
8         Person manu = new Person("Macron", "Emmanuel", 43, "Elysee à Paris");
9         Person poutine = new Person ("Poutine", "Vladimir", 68);
10        Person biden = new Person ("Biden", "joe");
11
12        System.out.println(manu);
13        System.out.println(poutine);
14        System.out.println(biden);
15    }
16 }

```

Problems @ Javadoc Declaration Console Diagrams

<terminated> TestPerson [Java Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (6 mai 2021 à 12:36:00)

Person [lastName=Macron, firstName=Emmanuel, age=43, address=Elysee à Paris]
 Person [lastName=Poutine, firstName=Vladimir, age=68, address=unknown]
 Person [lastName=Biden, firstName=joe, age=0, address=unknown]

- **Exercice 1.9 :** Ajouter à votre classe Person un attribut « Ville de Naissance » qui est un objet de type City de sorte d'obtenir ce qui suit :

```

6 public class TestPerson {
7     public static void main(String[] args) {
8         Person manu = new Person("Macron", "Emmanuel", 43, "Elysee à Paris", new City("Amiens", "France"));
9         Person poutine = new Person ("Poutine", "Vladimir", 68);
10        Person biden = new Person ("Biden", "joe");
11
12        System.out.println(manu);
13        System.out.println(poutine);
14        System.out.println(biden);
15    }
16 }

```

Problems @ Javadoc Declaration Console Diagrams

<terminated> TestPerson [Java Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (6 mai 2021 à 12:54:04)

Person [lastName=Macron, firstName=Emmanuel, age=43, address=Elysee à Paris]BornCity [name=Amiens, state=France, nbInhabitants=0]
 Person [lastName=Poutine, firstName=Vladimir, age=68, address=unknown]Bornnull
 Person [lastName=Biden, firstName=joe, age=0, address=unknown]Bornnull

- **Exercice 2.0 :** Voici une liste de personnalité dont nous souhaitons extraire toutes les personnes nées en France ou habitant Paris.

Notez au passage que la population d'une ville n'est affichée que si elle a été remplie aussi apportez les modifications nécessaires pour arriver au résultat suivant :

Liste des personnalités

Macron , Emmanuel , 43ans , habitant L'Elysée à Paris , Ville de naissance : Amiens , France

Sarkozy , Nicolas , 66ans , habitant Paris , Ville de naissance : Paris , France , population Paris 2000000 d'habitants

Johnson , Boris , 56ans , habitant Downing street à London , Ville de naissance : New York , Etats-unis

Depardieu , Gerard , 72ans , habitant Moscou , Ville de naissance : Châteauroux , France

Kravitz , lenny , 56ans , habitant Hotel particulier à Paris , Ville de naissance : New York , USA

Lawrence , jennifer , 30ans , habitant Louisville aux USA , Ville de naissance : Indian Hills , USA

Liste après notre filtre

Macron , Emmanuel , 43ans , habitant L'Elysée à Paris , Ville de naissance : Amiens , France

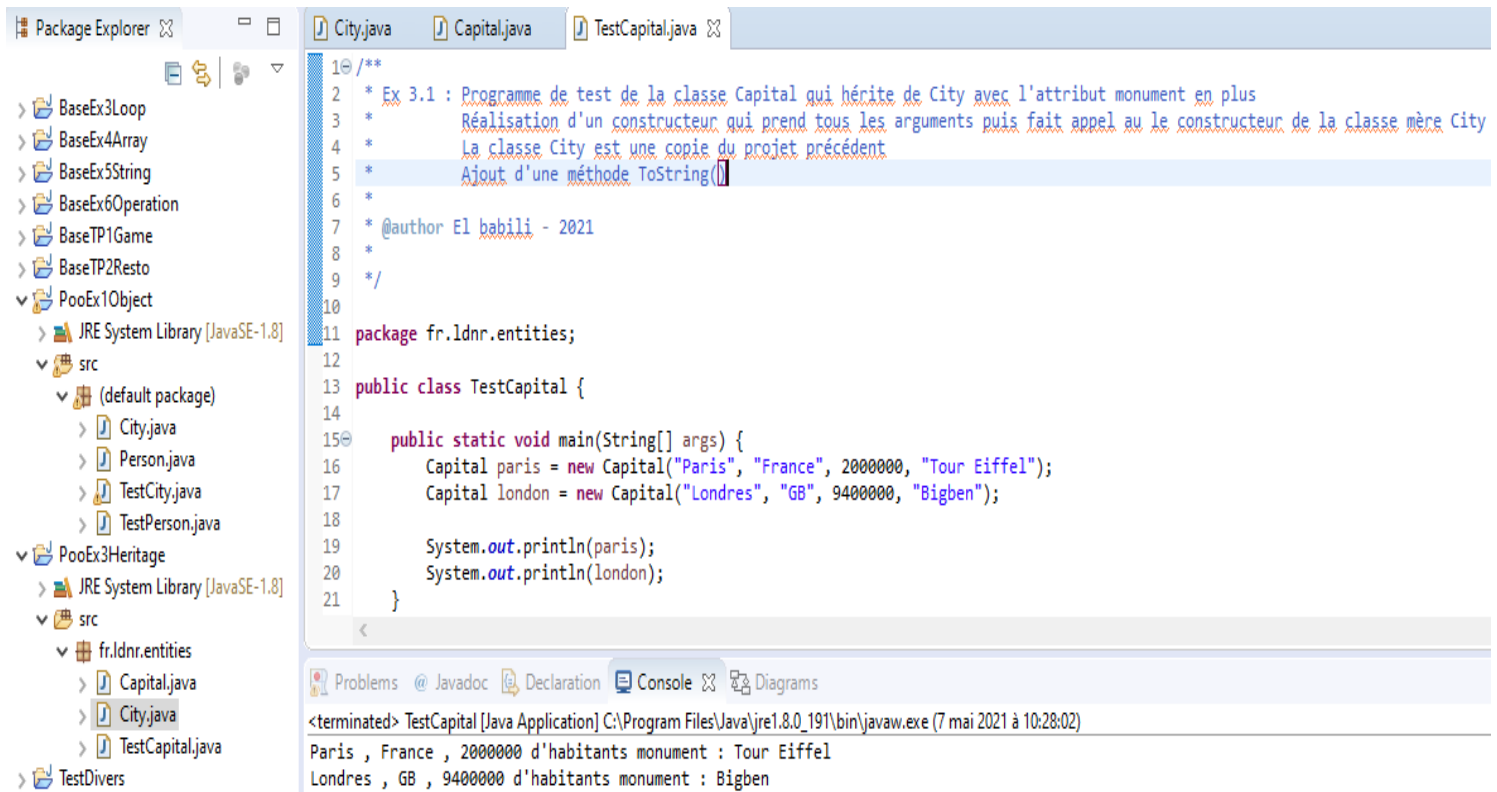
Sarkozy , Nicolas , 66ans , habitant Paris , Ville de naissance : Paris , France , population Paris 2000000 d'habitants

Depardieu , Gerard , 72ans , habitant Moscou , Ville de naissance : Châteauroux , France

Kravitz , lenny , 56ans , habitant Hotel particulier à Paris , Ville de naissance : New York , USA

Exercice 3 : PooEx3Heritage

- **Exercice 3.1 :** Commencer cette fois ci par créer un package "fr.fms.entities" puis dans celui-ci une classe Capitale qui hérite de la classe Ville avec comme particularité d'avoir un monument. En effet, Paris est une Ville avec un monument la tour Eiffel ! (N'oublier pas d'ajouter la classe Ville dans votre package) puis ajouter une classe de Test avec un main dans votre package afin d'instancier plusieurs capitale, les modifier via les accesseurs et les afficher in fine via la méthode toString(). Nous devrions obtenir ceci :



```
1 /**
2  * Ex 3.1 : Programme de test de la classe Capital qui hérite de City avec l'attribut monument en plus
3  * Réalisation d'un constructeur qui prend tous les arguments puis fait appel au le constructeur de la classe mère City
4  * La classe City est une copie du projet précédent
5  * Ajout d'une méthode ToString()
6  *
7  * @author El babili - 2021
8  *
9  */
10
11 package fr.fms.entities;
12
13 public class TestCapital {
14
15     public static void main(String[] args) {
16         Capital paris = new Capital("Paris", "France", 2000000, "Tour Eiffel");
17         Capital london = new Capital("Londres", "GB", 9400000, "Bigben");
18
19         System.out.println(paris);
20         System.out.println(london);
21     }
22 }
```

<terminated> TestCapital [Java Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (7 mai 2021 à 10:28:02)

Paris , France , 2000000 d'habitants monument : Tour Eiffel
Londres , GB , 9400000 d'habitants monument : Bigben

- **Exercice 3.2 :** Dans le même package, ajouter la classe Employé qui hérite de Personne tel qu'un Employé est une Personne avec comme particularité d'être dans une entreprise avec un salaire. Puis créer une classe Test dans votre main pour instancier des Employés, afficher les en ajoutant une méthode toString() ! Faites en sorte qu'il y ai des employés ayant pour ville de naissance de type Capital ou City. Nous devrions avoir ce résultat sur la console :

Gate , Bill , 65ans , habitant aux USA , Ville de naissance : Washington , USA ,
7600000 d'habitants monument : W-M , Entreprise : Fondation B&M , salaire : 100000.0
Musk , Elon , 49ans , habitant à palo alto , Ville de naissance : Pretoria , Afrique du
Sud , 800000 d'habitants , Entreprise : Tesla , salaire : 150000.0

- **Exercice 3.3 :** Ajouter une classe Commerciale qui est une personne qui travaille pour une entreprise avec une rémunération sur la base d'un pourcentage du chiffre d'affaires par mois puis vous connaissez dorénavant le mode opératoire : tester là !

Dupont , robert , 50ans , habitant rue des rosiers à Toulouse , Ville de naissance :
Limoges , France , 133000 habitants Entreprise : brico , % CA : 5.0

- **Exercice 3.4 :** Veiller à empêcher l'utilisateur de faire des mauvaises saisies (salaire ou remise négative par ex), dans ce cas, mettez des valeurs minimums par défaut que l'on pourra aisément modifier par la suite, tester pour vérifier si ça répond aux besoins.

% du chiffre d'affaire inférieur à 0 impossible

Dupont , robert , 50ans , habitant rue des rosiers à Toulouse , Ville de naissance : Limoges , France , 133000 habitants Entreprise : brico , % CA : 3.5

- **Exercice 3.5 :** Faites en sorte que la population d'une ville n'apparaisse plus dorénavant

Gate , Bill , 65ans , habitant aux USA , Ville de naissance : Washington , USA monument : W-M , Entreprise : Fondation B&M , salaire : 100000.0

- **Exercice 3.6 :** Dans une classe de Test « TestEntities » par ex, Utiliser un tableau avec plusieurs instances d'Employee et Commercial à parcourir pour afficher les informations de chaque objet la aussi à l'aide de la méthode toString(). Reproduisez le même test avec une liste d'objets pouvant contenir des employés ou des commerciaux.

```
Gate , Bill , 65ans , habitant aux USA , Ville de naissance : Washington , USA monument : W-M , Entreprise : Fondation B&M , salaire : 100000.0
Musk , Elon , 49ans , habitant à palo alto , Ville de naissance : Pretoria , Afrique du Sud , Entreprise : Tesla , salaire : 150000.0
Dupont , robert , 50ans , habitant rue des rosiers à Toulouse , Ville de naissance : Limoges , France Entreprise : brico , % CA : 5.0
-----
```

- **Exercice 3.7 :** Ajouter la méthode rémunération d'un côté pour empêcher des instanciations de Person et de l'autre pour permettre aux classes filles de calculer respectivement la rémunération d'un employé (salaire - charges) ou d'un commercial (% du ca). Utiliser des données fictives et(ou) moyennes pour les charges (20%) et chiffre d'affaire (moyen : 50000) . Parcourez à nouveau le tableau de Person et appelez cette nouvelle méthode avec un affichage réduit :

```
-----
salaire mensuel de Bill Gate : 80000.0
salaire mensuel de Elon Musk : 120000.0
salaire mensuel de robert Dupont : 2500.0
```