

1. Définir une classe `Personne` permettant de représenter un individu avec comme attributs : un nom, un prénom, une adresse, une date de naissance. Pensez aux impacts sur les constructeurs, les affichages (`toString()`), etc. On devra assurer qu'il soit impossible de modifier le nom, le prénom et la date de naissance d'une personne (attributs privés accessibles en lecture seule).

```
Personne jby = new Personne("Jean-Baptiste", "Yunès",
                           "666 rue il Diavolo, 66600 Inferno", new Date(1922,5,1));
System.out.println(jby);
jby.setAdresse("999 avenue John Lennon, 99990 Edenpark");
System.out.println(jby);
```

2. Créer une collection de personnes et écrire une méthode de classe (statique) permettant d'afficher les adresses de toutes les personnes de la liste, une autre pour tous les prénoms :

```
void afficheAdresses(ArrayList<Personne> v);
void affichePrenoms(ArrayList<Personne> v);
```

Note, une `ArrayList` est un type de donnée prédéfini de Java permettant de stocker un nombre d'éléments variables d'objets du même type. Pour mieux comprendre comment ce type fonctionne, il faut (au minimum) se reporter à sa définition (documentation de `java.util.ArrayList` ou consulter Internet - attention Internet n'est pas toujours un bon ami) :

```
Personne jby = new Personne("JB", "Y", "A", new Date(1914,7,28));
Personne cd = new Personne("C", "D", "xxxx", new Date(1918,11,11));

ArrayList<Personne> l = new ArrayList<Personne>(); // création d'une liste vide
l.add(jby); // ajout de jby dans la liste
l.add(cd); // ajout de cd dans la liste
System.out.println(l); // affichage de la liste
Personne.afficheAdresses(l); // appel de la fonction utilitaire...
Personne.affichePrenoms(l);
```

3. Écrire une fonction permettant de lire au clavier un nom, un prénom, etc, de construire une personne avec ces caractéristiques et de la renvoyer en retour :

```
Personne lireUnePersonneAuClavier();
```

Utiliser cette fonction pour créer une liste de personnes puis obtenir l'ensemble des adresses, ou des noms, etc, des personnes de la liste.

4. Modifier la classe `Personne` de sorte qu'une personne puisse avoir plusieurs prénoms (on utilisera peut-être une `ArrayList<String>` pour stocker l'ensemble des prénoms de la personne...).
5. Créer une classe `CompteEnBanque` sur lesquels il est permis de déposer, retirer une certaine somme et déterminer combien il y a sur le compte en banque. L'opération de retrait devra vérifier que le compte en banque est suffisamment garni, et si l'opération est possible renverra `true` et `false` sinon.
6. Modifier la classe `Personne` de sorte que l'on puisse associer à une personne son compte en banque. Écrire une fonction permettant de calculer la somme détenue par l'ensemble des personnes d'une liste d'individus :

```
int getRichesse(ArrayList<Personne> population);
```

7. Modifier la classe `Personne` de sorte qu'elle puisse posséder plusieurs comptes en banque (à l'aide d'une `ArrayList<CompteEnBanque>`) et que l'affichage d'une personne conduise à afficher sa richesse totale.

```
Personne jby = new Personne("JB", "Y", "A", new Date(1914, 7, 28));
CompteEnBanque c1 = new CompteEnBanque();
CompteEnBanque c2 = new CompteEnBanque();
jby.addCompteEnBanque(c1);
jby.addCompteEnBanque(c2);
c2.deposer(1000);
c1.deposer(100);
System.out.println(jby); // affiche qqe chose comme : JB, Y, A, 28 juillet 1924, possède 1100€
```