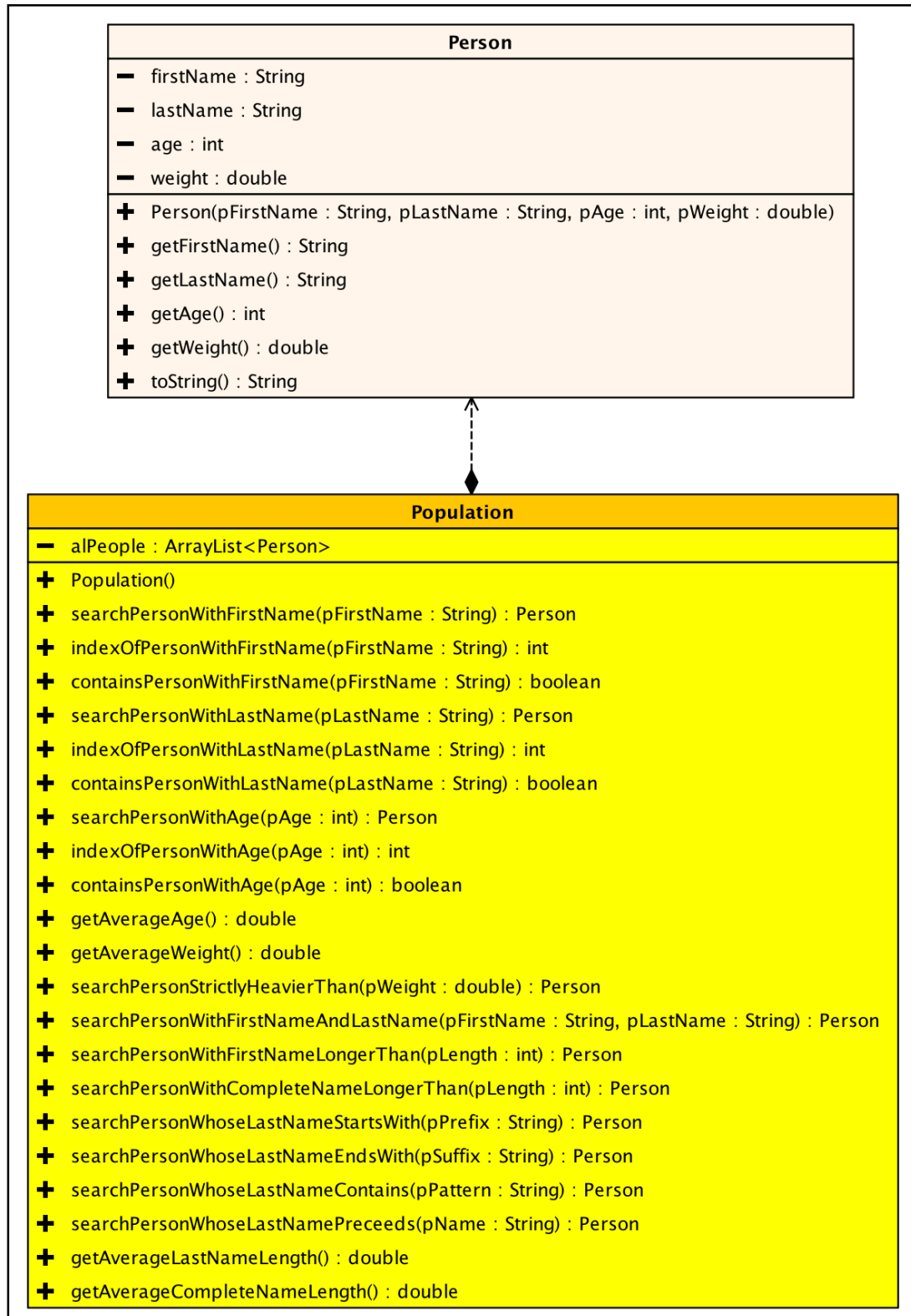


Exercice M2 - Recherche linéaire

Le but de cet exercice est de compléter une classe `Population` qui permet de gérer une liste de personnes représentées elles par des objets de la classe `Person`.



Un modèle vous est fourni. Il contient déjà la classe `Person` toute entière ainsi que la classe `Population` avec son constructeur. Le constructeur crée une liste contenant quelques personnes avec des valeurs d'attributs aléatoires, pour vous permettre de tester plus facilement vos méthodes.

Partie 1 : Répétition

0. Copiez le modèle **ExerciceM2_M** du répertoire de classe vers votre répertoire personnel.
1. Ajoutez la méthode `searchPersonWithFirstName` qui retourne la première personne portant le prénom indiqué en paramètre.
2. Ajoutez la méthode `indexOfPersonWithFirstName` qui retourne l'index de la première personne portant le prénom indiqué en paramètre.
3. Ajoutez la méthode `containsPersonWithFirstName` qui vérifie s'il y a une personne portant le prénom indiqué en paramètre.
4. Ajoutez la méthode `searchPersonWithLastName` qui retourne la première personne portant le nom de famille indiqué en paramètre.
5. Ajoutez la méthode `indexOfPersonWithLastName` qui retourne l'index de la première personne portant le nom de famille indiqué en paramètre.
6. Ajoutez la méthode `containsPersonWithLastName` qui vérifie s'il y a une personne avec le nom de famille indiqué en paramètre.
7. Ajoutez la méthode `searchPersonWithAge` qui retourne la première personne ayant l'âge indiqué en paramètre.
8. Ajoutez la méthode `indexOfPersonWithAge` qui retourne l'index de la première personne ayant l'âge indiqué en paramètre.
9. Ajoutez la méthode `containsPersonWithAge` qui vérifie s'il y a une personne avec l'âge indiqué en paramètre.
10. Ajoutez la méthode `getAverageAge` qui retourne l'âge moyen de toutes les personnes.
11. Ajoutez la méthode `getAverageWeight` qui retourne le poids moyen de toutes les personnes.

Partie 2 : Méthodes pour élèves rapides

Pour la résolution des points 3 à 10, il vous faut des méthodes de la classe `String` que vous ne connaissez pas encore. Sur le site suivant, il y a de brèves descriptions de toutes les méthodes de la classe `String`. Utilisez celles que vous jugez utiles.

<http://docs.oracle.com/javase/7/docs/api/java/lang/String.html>

1. Ajoutez la méthode `searchPersonStrictlyHeavierThan` qui retourne la première personne ayant un poids strictement supérieur (>) au poids indiqué en paramètre.
2. Ajoutez la méthode `searchPersonWithFirstNameAndLastName` qui retourne la première personne portant le nom et prénom indiqués en paramètre.
3. Ajoutez la méthode `searchPersonWithFirstNameLongerThan` qui retourne la première personne portant un prénom composé de strictement plus de caractères que le nombre indiqué en paramètre.
4. Ajoutez la méthode `searchPersonWithCompleteNameLongerThan` qui retourne la première personne portant un nom complet (prénom et nom de famille combinés) composé de strictement plus de caractères que le nombre indiqué en paramètre.
5. Ajoutez la méthode `searchPersonWhoseLastNameStartsWith` qui retourne la première personne dont le nom de famille commence par le préfixe indiqué en paramètre.
6. Ajoutez la méthode `searchPersonWhoseLastNameEndsWith` qui retourne la première personne dont le nom de famille se termine par le suffixe indiqué en paramètre.
7. Ajoutez la méthode `searchPersonWhoseLastNameContains` qui retourne la première personne dont le nom de famille contient la séquence de caractères indiquée en paramètre.
8. Ajoutez la méthode `searchPersonWhoseLastNamePreceeds` qui retourne la première personne dont le nom de famille précède par ordre alphabétique le nom indiqué en paramètre.
9. Ajoutez la méthode `getAverageLastNameLength` qui retourne la longueur moyenne (en nombre de caractères) de tous les noms de famille.
10. Ajoutez la méthode `getAverageCompleteNameLength` qui retourne la longueur moyenne (en nombre de caractères) de tous les noms complets (prénom et nom de famille combinés).