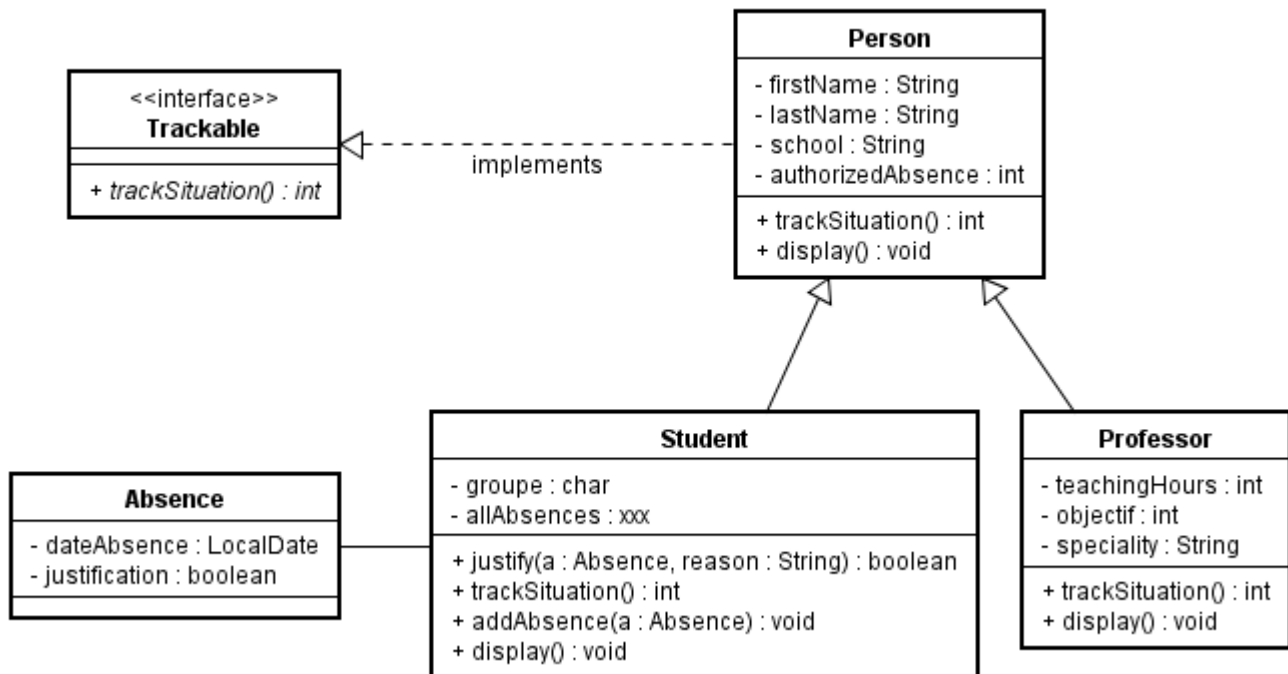


Etude de Cas : Java

L'école « **DataTruc** » souhaite développer une application pour le suivi de ses étudiants et ses professeurs, elle a donc opté pour une approche orienté objet.

Soit l'extrait de diagramme de classe suivant :



L'école « DataTruc » a défini un seuil de nombre d'absence autorisé égal à **10**.

Une personne est caractérisée par son nom et son prénom, on doit pouvoir suivre sa situation et afficher ses informations. Il existe deux types de personnes dans l'école, les étudiants et les professeurs :

- Les étudiants sont caractérisés par leur groupe (par exemple 'A' ou 'B'). L'application permet de gérer pour chaque étudiant l'ensemble des absences qu'il a réalisé.
- Les professeurs possèdent une spécialité et un objectif d'heure d'enseignement. L'application permet de gérer pour chaque professeur son état d'avancement dans les heures d'enseignement.

Une information concernant l'absence comporte la date de l'absence et un attribut booléen qui spécifie si l'absence est justifiée ou pas. Un étudiant peut justifier une absence en spécifiant un motif, elle devient donc justifiée et n'est plus comptabilisée.

Le suivi de situation est comme suit :

- Pour les étudiants il s'agit de retourner le nombre d'absence autorisé restant, seul les absences non justifiées sont comptés
- Pour les professeurs il s'agit de retourner le nombre d'heure restant.

L'école désire connaître l'étudiant qui a le plus d'absence.

Finalement, il est obligatoire de renseigner un motif lors de la justification d'absence.

Questions :**Partie 1**

- Créer les classes **Person**, **Student**, **Professor** et **Absence** (classe / constructeurs / méthodes / getters / setters)
- Créer une classe **TestSchool** qui effectue le traitement suivant :
 - Instancier les objets suivants

Student			
lastName	firstName	group	allAbsence
A	Youssef	'A'	null
A	Mohammed	'B'	null
A	Mouna	'A'	null

Professor				
lastName	firstName	teachingHours	objectif	speciality
B	Ahmed	12	60	computer science
A	Imane	32	80	communication

La méthode `addAbsence` permet d'ajouter une absence donnée à `allAbsence`

- Ajouter deux absences pour Youssef
 - Le 03/12/18 – Non justifiée
 - Le 28/11/18 – Non justifiée
- Ajouter une absence pour Mouna
 - Le 26/11/18 – Non justifiée

Partie 2

- Créer l'interface **Trackable**
- Dans les classes **Student** et **Professor**
 - Implémenter la méthode `trackSituation`
- Créer une classe **AbsenceException** qui retourne le message suivant : « Motif non renseigné » lorsqu'on essaie de justifier l'absence avec un chaîne de caractère vide ().
- Modifier la classe **Student** en conséquence

Partie 3

- Dans la classe **Student**
 - Implémenter la méthode `justifyAbsence`
- Dans la classe **TestEntreprise**
 - Justifier l'absence de Mouna du 26/11/18 pour motif de « Participation à une journée sportive »
 - Afficher l'étudiant qui a le plus d'absence