

### **Commentaires du diagramme de classe**

On a choisi de faire quatre classes dont les trois principales Compteur, Jour, et Quartier. De plus, nous avons ajouté une classe Comptage qui permet de faire le lien entre Compteur et Jour. On a ajouté une classe contenant des énumérations qui sont des attributs des autres classes.

Pour la classe Quartier, on a ajouté les méthodes addCpt(Compteur) qui ajoute à la liste de Compteurs un nouvel objet, puis dropCpt(int) qui enlève un Compteur de la liste situé à l'indice en paramètre, on a également une méthode toKm() qui renvoie la longueur des pistes cyclables initialement en mètres en kilomètres.

Dans la classe Compteur, nous avons ajouté une méthode coordonnées(double, double) qui prend en paramètres la longitude et la latitude et qui renvoie un ArrayList<double> contenant les coordonnées du compteur, nous avons aussi ajouté une méthode nomComplet(String, String) qui prend en paramètre le libellé et le sens du Compteur et qui renvoie un String contenant le nom complet.

Dans la classe Jour, nous avons ajouté estVacances(LocalDate) qui prend la date en paramètre et renvoie True si elle est en période de vacances. On a aussi ajouté la méthode estWeekEnd() qui renvoie True si le jour est en période de week-end.

Dans la classe Comptage, nous avons ajouté, en plus des get() et set(), une méthode nbVelos(int) qui prend l'heure et qui renvoie le nombre de vélos à être passés à cette heure, devant le compteur. Pour chacune des classes, nous avons également ajouté une méthode toString() qui renvoie une chaîne de caractères contenant les informations relatives à la classe.

Pour la classe enumerations, on a trois objets de type enum, le premier, JourSemaine, contient les sept jours de la semaine et est utilisé dans la classe Jour. Le deuxième, Vacances, contient les différentes périodes de vacances possibles et est utilisé dans la classe Jour. Enfin, le dernier, PresenceAnomalie, contient les deux cas possibles pour une anomalie et est utilisé dans la classe Comptage.

Ces méthodes que nous avons ajoutées nous semblaient cohérentes par rapport aux classes sélectionnées.

Nous avons regroupé toutes ces classes dans un seul et même package "velo", avec cela on peut facilement effectuer les tests en appelant le package et toutes les classes à l'intérieur de celui-ci. La classe scenario permet de tester toutes nos méthodes pour voir le bon fonctionnement du programme.

### **Commentaires du diagramme de séquence**

Pour notre diagramme de séquence, nous avons décidé de faire un diagramme de notre scénario qui représente notre application mais qui n'est pas trop répétitif.