

**Durée : 1h10.** Documents autorisés. Faites une archive (tar.gz ou zip) ne contenant que les sources et déposez-le sur l'espace Moodle dans le devoir « TP Noté 03/11/21 ». Il est inutile d'écrire les accesseurs qui ne sont pas demandés et dont vous n'avez pas besoin. On veut ici écrire des classes permettant de manipuler des relevés de données météorologiques (température, direction du vent, précipitations).

1. Écrire une classe `datereleve` représentant une date et une heure de relevé<sup>1</sup> et munir cette classe d'un constructeur prenant comme arguments année, mois, jour, heure<sup>2</sup>.  
Écrire une méthode `tostring` retournant une chaîne du type « 2021.11.3 14h ».
2. Munir cette classe d'une méthode `comparememejour` déterminant si deux `datereleve` concernent le même jour du même mois de la même année, ainsi qu'une méthode `compareavant` déterminant si une `datereleve` se situe avant une autre `datereleve` dans l'ordre chronologique.
3. Nous allons maintenant décrire les *relevés*. Chaque relevé a été effectué à une `datereleve`. Il y a trois types de relevés :  
temp pour un relevé de *température* exprimée en dixièmes de degrés celsius (int, par exemple 201 pour 20.1°C) ;  
prec pour un relevé de *précipitations* pour lequel on mémorisera le type de précipitations (*pluie*, *grêle* ou *neige*) ainsi que la quantité mesurée en millimètres (int) ;  
vent pour la *direction* du vent (*Nord*, *Sud*, *Est*, *Ouest*).  
Déclarer les classes pour représenter les relevés et écrire les constructeurs. Il n'y aura pas de mutateurs : les valeurs des attributs seront fixées à la construction et ne seront pas modifiées ensuite.
4. Écrire une méthode `typerereleve` retournant une chaîne du type « Température », « Précipitations » ou « Vent ».
5. Définir une méthode `tostring` pour les relevés retournant des chaînes telles que « 2021.11.3 14h Température 18.0°C » « 2021.11.3 14h Précipitations Pluie 2mm » ou « 2021.11.3 14h Vent Nord ».
6. Écrire une méthode `sortiesurflux` prenant comme argument un flux de sortie et envoyant sur ce flux la chaîne retournée par `tostring`.
7. Définir une classe `basereleves` contenant des relevés, vous utiliserez pour cela `std::list`. À la construction, la base sera vide.
8. Ajouter une fonctionnalité `ajout` permettant d'ajouter un relevé à la base avec les contraintes suivantes :  
il ne peut y avoir dans la base deux relevés de température ou de direction de vent pour la même `datereleve` ;  
il ne peut y avoir dans la base deux relevés de précipitations avec le même type de précipitations pour la même `datereleve`<sup>3</sup>.  
La méthode retournera un booléen valant vrai si l'ajout a été effectué et faux sinon.
9. Écrire une méthode `sortirrecents` prenant comme argument un flux de sortie et une `datereleve` et sortant sur le flux les relevés postérieurs à la `datereleve` passée.
10. Écrire une méthode `tempmoyenne` prenant comme argument une `datereleve` et retournant la température moyenne relevée à cette date-là (par date, on entend jour et mois), en prenant en compte tous les relevés de température de cette date (l'heure de la `datereleve` passée en argument sera donc ignorée).
11. Écrire une méthode `tempsdangereux` retournant les `datereleve` pour lesquels il y a un relevé de vent, une température inférieure à 2, et des précipitations de neige ou grêle.

1 Dans le cadre de l'exercice, on s'interdit d'utiliser les classes fournies par la bibliothèque standard pour représenter le temps. On ne mémorisera pas les minutes, car on fera au plus un relevé par heure.

2 On ne fera aucune vérification sur la validité des valeurs passées au constructeur.

3 Il est donc possible d'avoir dans la base pour une `datereleve` plusieurs relevés de précipitations, à condition que ces précipitations soient de types différents. Par exemple, il est possible d'avoir pour une `datereleve` un relevé de précipitations de neige et un autre de pluie pour représenter une chute de pluie et neige mêlées.