EVALUATION CONTINUEE

POLYMORPHISME ET STL

Programme à écrire

Dans ce programme nous allons définir une hiérarchie de classes en héritage représentant divers véhicules de transport. Un **Vehicule** sera spécialisé en véhicule **Volant** ou **Terrestre**. Un véhicule volant sera spécialisé en **Helicoptere** ou **Avion**, un véhicule terrestre en **Velo** ou **Voiture**.

Les classes **Vehicule**, **Volant** et **Terrestre** seront abstraites, et contiendront un champ **vitesse** (en km/h) et une méthode virtuelle pure **avancer** dont le but sera d'afficher un message à l'écran.
La classe **Volant** contiendra une méthode supplémentaire **voler**, qui renverra (pas d'affichage !) juste la chaîne de caractère : « vole à *x* km/h », où x est la vitesse du véhicule. Sur le même modèle vous ajouterez une méthode **rouler** à **Terrestre**.

Les classes **Helicoptere**, **Avion**, **Velo** et **Voiture** ajouteront un champ **modele** qui contiendra le nom du véhicule (Mercedes, Airbus, etc.). Ces classes redéfiniront la méthode avancer pour qu'elle affiche à l'écran un message ressemblant à (exemple donné pour une voiture de modèle Mercedes et de vitesse 130km/h) : « Mercedes roule à 130 km/h ». La méthode avancer utilisera la méthode voler ou rouler selon le cas de figure.

Dans la fonction main :

1/ Créer un **std::vector** de véhicules, et remplissez le d'au moins deux instances de **Helicoptere**, **Avion**, **Velo** et **Voiture**. Choisissez vous-même des vitesses et noms de modèle différents.

2/ Utiliser l'algorithme de la STL **std::random_shuffle** pour mélanger l'ordre des éléments.

3/ Parcourir le vector mélangé et appeler sur chaque élément la méthode avancer.
Parcourir une seconde fois le vector, retrouver et afficher pour chaque élément si il vole ou roule.
Pour cela vous devrez utiliser le mécanisme de cast adapté en c++.

4/ Regrouper maintenant les éléments de la liste dans l'ordre. On choisira comme ordre : tous les volants puis tous les terrestres. Il vous est demandé d'utiliser pour cela un foncteur de comparaison, à définir, et l'algorithme de tri de la STL **std::sort**.