**HAI715I – Ordres, treillis, induction (Examen)**

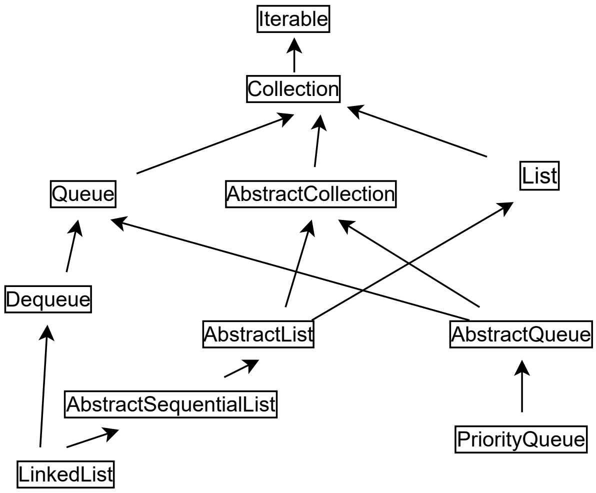
*12 janvier 2021 - 16h-18h*

Université de Montpellier – Faculté Des Sciences - Master informatique (Génie Logiciel)

Tous les supports de cours et vos notes personnelles sont autorisés. Les ordinateurs portables, tablettes, calculatrices, etc. sont interdits. L'examen dure 2h. Le sujet comporte 2 parties.

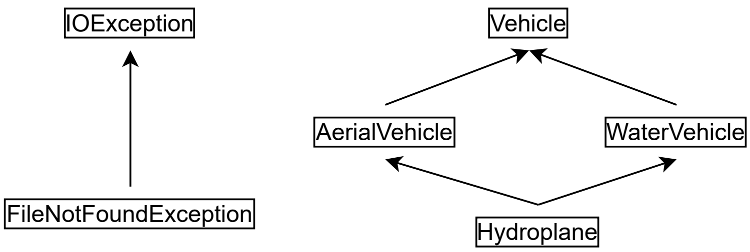
## Partie sur les ordres partiels et les treillis

**Q1.** La fermeture réflexive et transitive de la hiérarchie des collections ci-dessous forme un ordre partiel. Cet ordre partiel n'a pas une structure de treillis. Indiquez pourquoi, puis proposez une complétion de cet ordre partiel afin d'obtenir un treillis qui le contienne.




**Q2.** Formaliser la relation d'inclusion entre packages de Java :
- Proposer une notation pour l'ensemble des packages et cette relation ; indiquez dans quel produit cartésien cette relation est incluse.
- Est-elle : réflexive, irréflexive, symétrique, anti-symétrique, transitive ? Donnez des arguments.

**Q3.** On donne les hiérarchies suivantes.




**Q3.1** Dessinez les ordres partiels correspondant à leur fermeture réflexive et transitive respective. Notons (E,<=E) et (C,<=C) ces deux ordres partiels.

**Q3.2** Dessinez le diagramme de Hasse de leur produit direct $(P, <=_P) = (E,<=_E) \times (C,<=_C)$.

**Q3.3** Utilisez cet ordre $(P, <=_P)$ pour déterminer, parmi ces quatre signatures de méthodes, quelle signature peut redéfinir quelle autre signature (au sens de l'annotation *Override*) :

*Vehicle read() throws IOException*
*Vehicle read() throws FileNotFoundException*
*WaterVehicle read() throws FileNotFoundException*
*AerialVehicle read() throws IOException*

**Partie sur l'induction**

**Exercice 1**

On considère le type inductif N des entiers naturels vu en cours.

**Q1.** Écrire la fonction *mult*, qui étant donnés deux entiers naturels $n_1$ et $n_2$, réalise la multiplication de $n_1$ par $n_2$. Pour ce faire, on utilisera la fonction d'addition *plus*, vue en cours et la définition inductive de cette fonction devra se faire par rapport au premier argument $n_1$.

**Q2.** Démontrer que : *(mult 2 n) = (plus n n)*, pour tout entier naturel *n*.
A-t-on besoin de réaliser une induction structurelle pour faire cette preuve ?

**Q3.** Démontrer que : *(mult n 2) = (plus n n)*, pour tout entier naturel *n*.
A-t-on besoin de réaliser une induction structurelle pour faire cette preuve ?

**Exercice 2**

On considère le type inductif L des listes (polymorphes) vu en cours.

**Q1.** Définir la relation inductive *is_rev*, qui teste si une liste est l'inversion d'une autre liste.
Exemple : [1; 2; 3] est une inversion de [3; 2; 1] (et vice-versa).

**Q2.** Démontrer que [1; 2; 3] est une inversion de [3; 2; 1].

**Q3.** Écrire la fonction *rev*, qui étant donnée une liste *l*, rend la liste *l* inversée.
Exemple : l'évaluation de (*rev* [1; 2; 3]) donne [3; 2; 1].

**Q4.** Écrire le schéma d'induction fonctionnelle correspondant à la fonction *rev*.

**Q5.** Démontrer la correction de la fonction *rev* vis-à-vis de la relation *is_rev*.