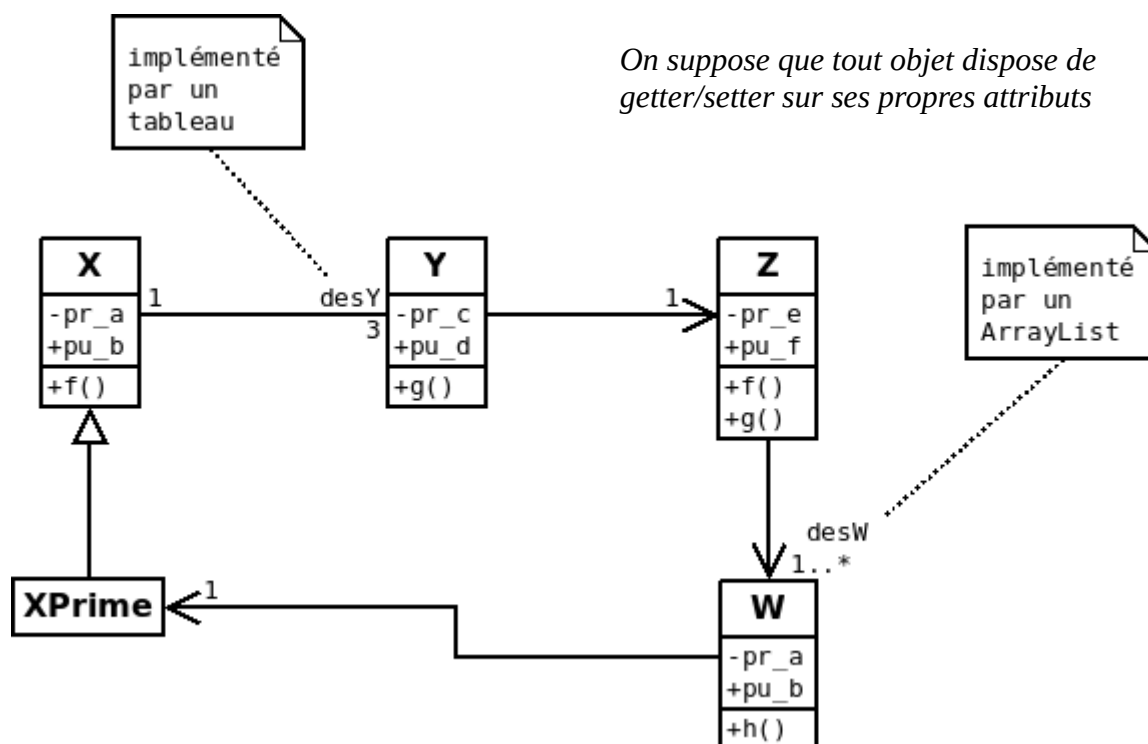


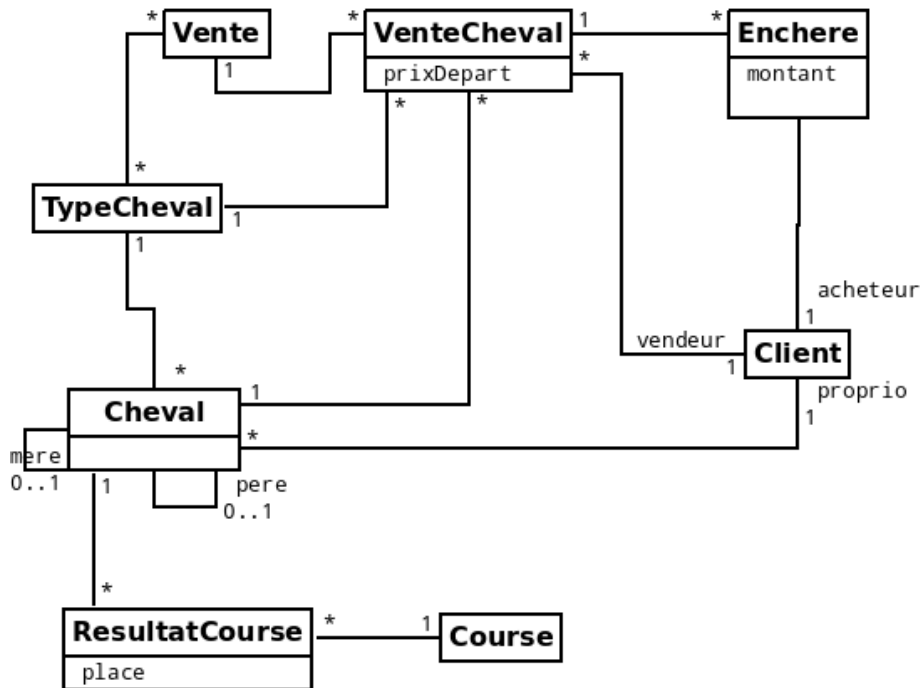
PARTIE I (8 points)

Compréhension du principe d'encapsulation et de navigation dans un graphe d'objets



Question	Réponse	Justification
Q2/ Une instance de X peut-elle accéder à la fonction g() du troisième élément de type Y d'un objet de type X ? (1 point)	oui	<code>this.getDesY()[2].g()</code>
Q3/ Une instance de W peut-elle accéder à la fonction f() d'un objet de type Z ? (1 point)		
Q4/ Une instance de W peut-elle savoir combien d'instance de type W sont en relation (par transitivité) avec l'instance de XPrime liée à elle ? (2 points)		
Q5/ Une instance de Y peut-elle accéder à la fonction f() d'un objet de type X ? (2 points)		
Q6/ Une instance de Y peut-elle accéder à la fonction f() d'un objet de type XPrime sans passer par Z ? (2 points)		

PARTIE II (16 points)



Q7/ À partir d'une instance de Cheval peut-on accéder à son propriétaire ? (2 points)

Q8/ Peut-on connaître le nombre de ventes d'un cheval ? (2 points)

Q9/ Peut-on savoir, à partir d'une enchère, si l'acheteur est ou non le vendeur ? (2 points)

Q10/ On souhaite connaître le potentiel d'un cheval, basé sur le résultat de ses courses : on considère qu'un cheval *a un bon potentiel* s'il est arrivé au moins une fois dans les 3 premiers d'une course.

Q10.1/ (2 points) Concevoir l'interface d'une méthode qui réalise le travail demandé

Q10.2/ (2 points) Donnez le nom de classe dans laquelle vous envisagez déclarer cette méthode, en justifiant votre proposition

Q10.3/ (2 points) Proposez une implémentation de cette méthode

Q11/ (4 points) On souhaite ajouter une la logique plus forte que la précédente, sous la forme d'une nouvelle méthode, afin d'évaluer le *très fort potentiel* d'un cheval, qui sera calculé sur le fait qu'il est arrivé dans les 3 premiers dans, au moins, la moitié des courses auxquelles il a participé, à condition qu'il ait participé à au moins 3 courses. Proposez une implémentation de cette nouvelle méthode.

TODO