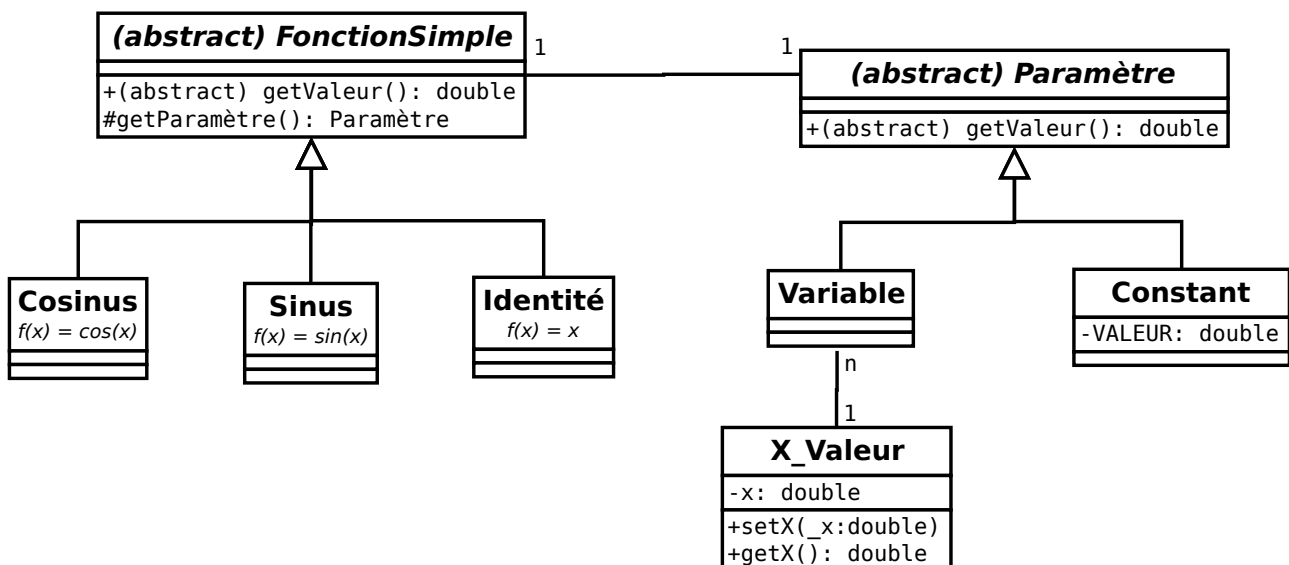


Examen de programmation orientée objet et java*Durée 2h**Documents autorisés**Le barème est donné à titre indicatif, il pourra être adapté***Partie A (4.5)/ Questions de cours**

- 1 Qu'est-ce que l'encapsulation ? Quelle est son utilité ?
- 2 Quelle différence y a t'il entre une interface et une classe abstraite ?
- 3 Donner le code permettant d'exécuter « print(« Hello world ») » dans un processus créé en utilisant une notation de classe anonyme.

Partie B / Calcul de fonctions simples (5Pts)

Rappel: un diagramme de classe ne représente pas nécessairement toutes les fonctions et tous les attributs

1 (1) Les classes Variable et Constant doivent-elles donner une définition de la fonction `getValeur()` ? Justifier.

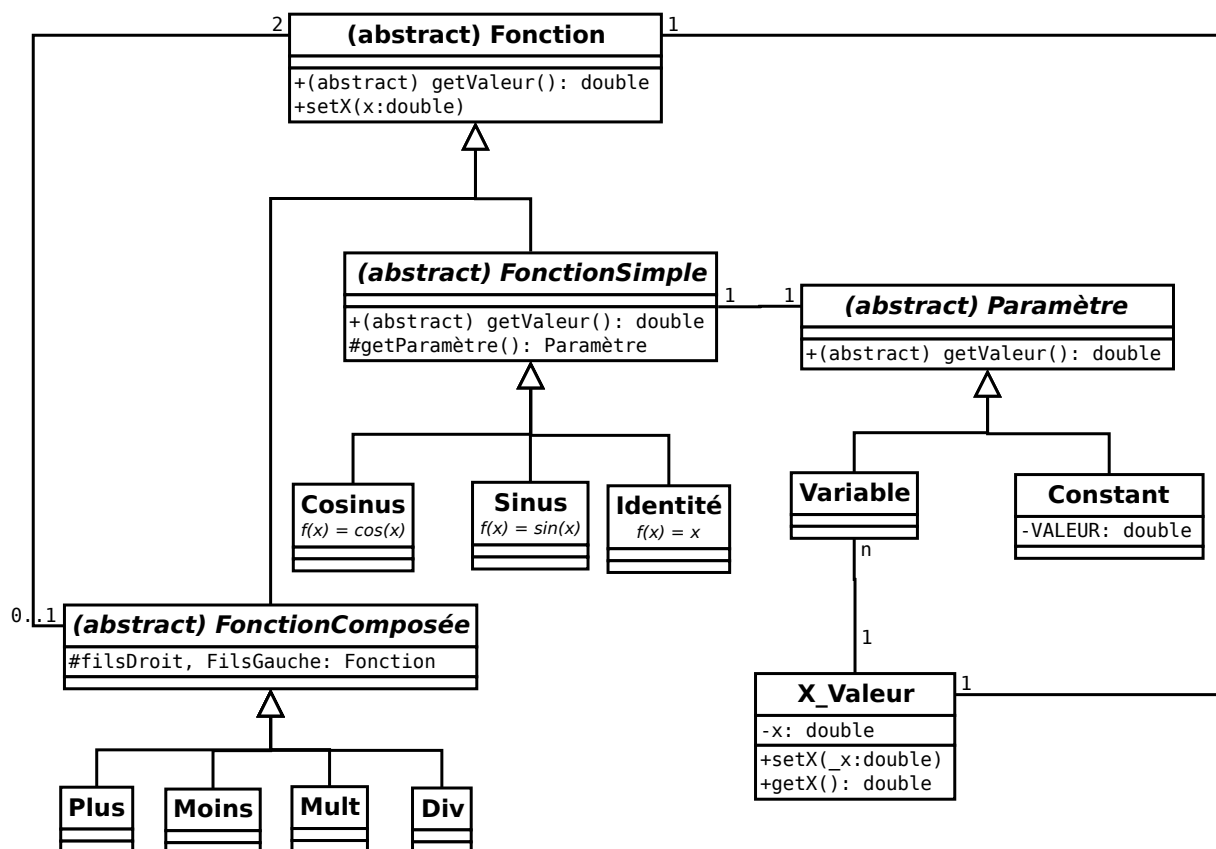
2 (2). Donner les définitions de la fonction `getValeur()` pour les classes Sinus et Identité.

Examen de programmation orientée objet et java*Durée 2h**Documents autorisés**Le barème est donné à titre indicatif, il pourra être adapté*

3 (2). Donner le diagramme objet du scénario : « soit un objet sinus(X), utilisant un Paramètre variable X = 5 »

Partie C / Calcul de fonctions (9.5 Pts)

On ajoute au diagramme précédent la classe FonctionComposée (et ses sous-classes). Chaque Fonction composée est composée de deux fonctions filles (nommées filsDroit et filsGauche). Ceci permet d'exprimer des fonctions sous la forme d'un arbre binaire (les nœuds non-feuilles de l'arbre étant de type FonctionComposée, les nœuds feuille étant de type FonctionSimple).



La fonction composée « plus » retourne la somme des valeurs de ses fils.

Examen de programmation orientée objet et java*Durée 2h**Documents autorisés**Le barème est donné à titre indicatif, il pourra être adapté***1 (1.5 pts) / Donner une définition de la fonction `getValeur():double` pour la classe `Plus`.**

La fonction `toString():String` de la classe `Objet` a été redéfinie pour les différentes classes du diagramme. Ceci permet de représenter toute fonction sous la forme d'une chaîne de la forme suivante : « $((\cos(x) + \sin(4)) * \tan(x))$ »

2 (1.5) Donner la définition de la fonction `toString()` de la classe `Plus`.

On ajoute à la classe `Fonction`, la fonction statique suivante qui respecte la priorité des opérateurs (+, *, etc.) et qui permet de générer une fonction en mémoire (un ensemble d'objets connectés) : *(static) `constructionFonction(expression:String):Fonction`*

3 (3) Soit le code suivant :

```
Fonction f = Fonction.constructionFonction(«  $\cos(x) + \sin(4) * \tan(x)$  ») ;  
f.setX(Math.Pi) ;
```

Donner le diagramme d'objet résultant du code précédant.

4 (3) À la suite du traitement de la question 3, on exécute l'instruction : `f.calculer()` ;

Donner le diagramme de séquence associé au calcul de la fonction.

5 (1.5) L'objet `X_Valeur` est partagé pour toutes les fonctions de paramètre variable d'une fonction donnée (représentée sous la forme d'un arbre).

Serait-il judicieux de déclarer dans la classe `Fonction` cette variable de façon statique ? Justifier.