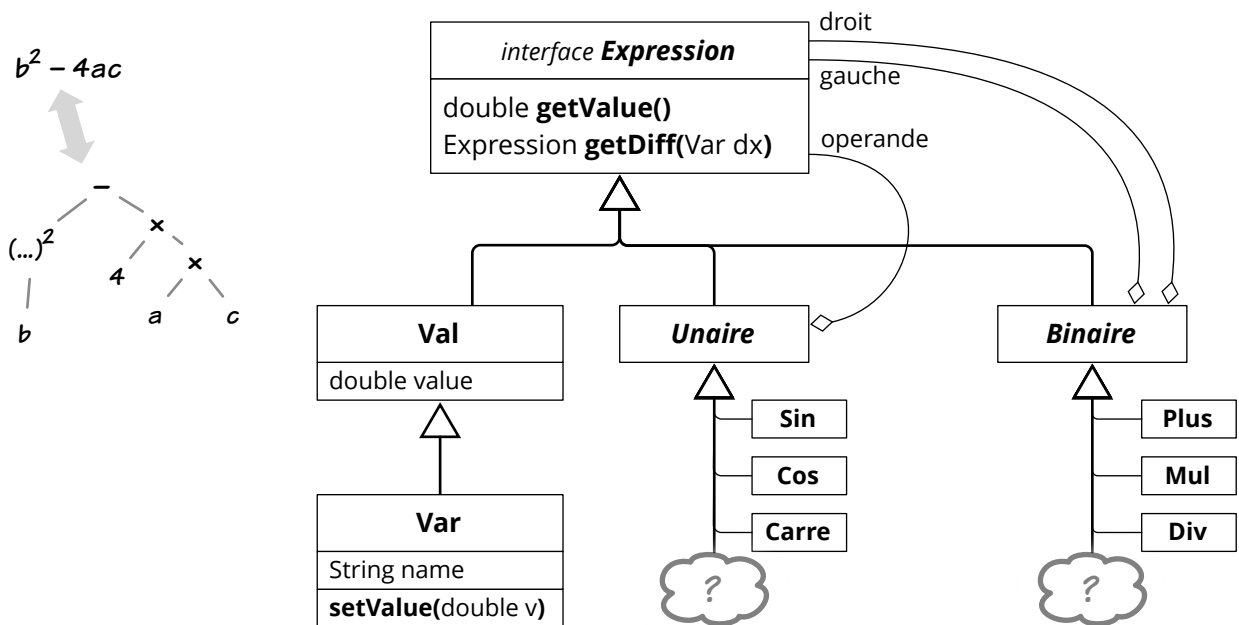


Aucun document autorisé; les rappels nécessaires sont inclus.

Répondre directement sur l'énoncé. Soin, lisibilité et orthographe seront pris en compte.

Dans cet examen, les questions se basent sur les interfaces et classes du système d'expressions mathématiques vu en TD. Quelques rappels :

- chaque (sous-)expression est représentée par un (sous-)arbre, tel qu'illustré à gauche;
- les classes du système modélisent les constantes (Val) et variables mathématiques (Var) nécessaires, ainsi que les fonctions et opérateurs de base.



On rappelle également la déclaration de l'interface Expression :

```
interface Expression {
    double getValue();           // évaluation numérique
    Expression getDiff(Var dx); // dérivation formelle, selon dx
}
```

Questions de cours : multiples choix possibles

Question 1. Dans l'expression à gauche du diagramme, les trois variables `a`, `b`, et `c` seraient représentées en Java par trois...

- variables Java
- classes différentes
- instances d'une même classe

Question 2. On considère l'élément dénoté `name` dans le diagramme. En Java, c'est...

- un attribut
- une méthode
- une classe
- de l'héritage
- une référence

Question 3. On considère l'élément dénoté `value` dans le diagramme. En Java, c'est...

- un attribut
- une méthode
- une classe
- de l'héritage
- une référence

Question 4. On considère l'élément dénoté `operande` dans le diagramme. En Java, c'est...

- un attribut
- une méthode
- une classe
- de l'héritage
- une référence

Question 5. Selon le diagramme, `gauche`, et `droit` sont...

- déclarés dans `Expression`
- de type `Expression`
- déclarés dans `Binaire`
- de type `Binaire`

Question 6. Les expressions sont représentées par une structure arborescente; quels éléments graphiques du diagramme UML représentent les liens de parenté entre les nœuds de cet arbre?

Question 7. Dans l'exercice vu en cours, les expressions implémentent la méthode `toString()`, cependant celle-ci est absente de l'interface `Expression`. Est-ce une erreur? Justifiez.

Question 8. Identifiez un exemple d'extension parmi les définitions du diagramme.

Question 9. Identifiez un exemple de redéfinition parmi les définitions du diagramme.

Polynômes d'une variable

Question 10. Initialisez la variable p ci-dessous avec la représentation du polynôme $x^2 + 2x + 3$, par combinaison des expressions déjà vues.

Expression x = new Var("x");

Expression p =

Cette représentation par produits et sommes étant peu pratique, on souhaite représenter de façon optimisée les polynômes d'une variable. La représentation d'un polynôme $a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ se fera à l'aide d'une nouvelle classe `Polynome`, qui associe à une variable la liste des coefficients $a_0 \dots a_n$.

Question 11. Écrivez la déclaration Java de la classe `Polynome`, de ses attributs, et de son constructeur. Pour simplifier, les polynômes peuvent être construits avec une liste `coefs` vide, la méthode `setCoef(n, a)` permettant ensuite d'initialiser individuellement chaque coefficient a_n .

Polynome
Var variable List<Double> coefs
setCoef(int rang, double coef) int degre()

Question 12. Écrivez la méthode `int degre()`.

- Le degré d'un polynôme est le rang n maximal pour lequel a_n est différent de zéro.
- Dans la liste `coefs`, l'indice de chaque coefficient est son rang.
- On considère que les coefficients de rang supérieur à la taille de la liste valent zéro.

Question 13. Écrivez la méthode `void setCoef(int rang, double coef)`. Le rang spécifié peut être supérieur à la taille de la liste, auquel cas il vous faut insérer autant de coefficients zéro que nécessaire.

Question 14. Écrivez la méthode `double getValue()` de la classe `Polynome`.
Rappel : l'élevation à la puissance x^n est réalisée par la méthode `Math.pow(x, n)`.

Question 15. Écrivez la méthode `String toString()` de la classe `Polynome`.
Formattage attendu pour $3x^3 + 2x + 1$: `"(1 + 2x + 3x^3)"` (NB. le terme de rang 2 est omis car $a_2 = 0$).

Question 16. Écrivez la méthode Expression `getDiff(Var dx)` de la classe Polynome.
Rappel : la dérivée d'un polynôme $P = a_0 + a_1x + \dots + a_nX^n$ est $P' = a_1 + \dots + n a_n X^{n-1}$.

Annexe — Documentation sur les listes Java

```
List<E> maListe; // déclaration d'une référence vers une liste
                // dont les éléments sont d'un certain type E
maListe = new LinkedList<E>(); // construction d'une liste vide

for (E e : maListe) { // itération sur les éléments d'une liste
    // utiliser l'élément e
}
```

Quelques méthodes utiles (pour une `List<E>`):

boolean	isEmpty()	Renvoie true si la liste est vide, false sinon.
int	size()	Renvoie le nombre d'éléments de la liste.
void	add(E e)	Ajoute l'élément e à la fin de la liste.
E	get(int i)	Renvoie l'élément d'indice i. Les éléments sont indicés à partir de zéro. Lève une erreur à l'exécution si l'indice est hors bornes.
E	set(int i, E e)	Remplace l'élément à l'indice i par e, et renvoie l'élément remplacé. Lève une erreur à l'exécution si l'indice est hors bornes.
E	remove(int i)	Retire puis renvoie l'objet à l'indice i de la liste. Les éléments suivants descendent d'un indice. Lève une erreur à l'exécution si l'indice est hors bornes.