

EXERCICE JAVA : INTERFACES ET CLASSES ABSTRAITES

EXERCICE + CORRIGE EN JAVA: EXPRESSIONS ARITHMÉTIQUES

On considère maintenant des expressions arithmétiques formées à partir de constantes réelles (nombres flottants) et utilisant les quatre opérations arithmétiques usuelles (addition, soustraction, multiplication et division). Une telle expression est par exemple $3 + (4.7 + 2.3) * 5$.

Ces expressions arithmétiques sont représentées par des arbres binaires. Les nœuds internes de l'arbre contiennent les opérateurs alors que les feuilles de l'arbre contiennent les constantes. Les fils gauche et droit d'un nœud interne représentent les deux sous-expressions gauche et droite. L'expression $3 + (4.7 + 2.3) * 5$ par l'arbre de la figure ci-dessous.

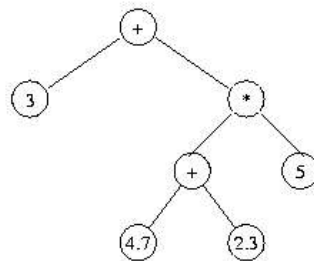


FIG 1. L'arbre de l'expression $3 + (4.7 + 2.3) * 5$

L'ORGANISATION EN INTERFACES ET CLASSES

1) -----

Définir une interface `ArithExpr` qui déclare les fonctionnalités d'une expression arithmétique. La seule fonctionnalité requise est de pouvoir être évaluée. Déclarer en conséquence une méthode `eval` du bon type.

Pour représenter l'expression, on va créer des objets pour chacun des nœuds. Ces objets vont bien sûr être des instances de classes différentes suivant qu'il s'agisse de nœuds internes ou de feuilles.

2) -----

Définir une classe `Constant` pour représenter les constantes des expressions.

3) -----

Définir quatre classes `Addition`, `Soustraction`, `Multiplcation` et `Division` pour les opérations arithmétique. On remarque que ces quatre classes partagent beaucoup de propriétés. Comment faut-il faire pour partager le code ?

4) -----

Redéfinir la méthode `toString` afin d'afficher les expressions en notation infixe. Écrire des méthodes `prefix` et `suffix` permettant d'afficher les expressions en notation préfixe et suffixe.

5) -----

Écrire une méthodes `prefix` et `suffix` permettnat d'afficher les expressions arithmétique en notation prefix et suffix.

SOLUTION

```
// Interface for arithmetic expressions
interface ArithExpr {
    double eval();
    String toString();
    String prefix();
    String suffix();
}

// Class representing constants
class Constant implements ArithExpr {
    private double value;

    public Constant(double value) {
        this.value = value;
    }

    @Override
    public double eval() {
        return value;
    }

    @Override
    public String toString() {
        return Double.toString(value);
    }

    @Override
    public String prefix() {
        return Double.toString(value);
    }

    @Override
    public String suffix() {
        return Double.toString(value);
    }
}

// Classes representing arithmetic operations
class Addition implements ArithExpr {
    private ArithExpr left;
    private ArithExpr right;

    public Addition(ArithExpr left, ArithExpr right) {
        this.left = left;
        this.right = right;
    }

    @Override
    public double eval() {
        return left.eval() + right.eval();
    }

    @Override
    public String toString() {
        return "(" + left.toString() + " + " + right.toString() + ")";
    }

    @Override
    public String prefix() {
        return "(+ " + left.prefix() + " " + right.prefix() + ")";
    }

    @Override
    public String suffix() {
        return "(" + left.suffix() + " " + right.suffix() + " +)";
    }
}

public class ArithExprTest {
    public static void main(String[] args) {
        // Create constant values
        ArithExpr constant1 = new Constant(3.0);
        ArithExpr constant2 = new Constant(4.7);
        ArithExpr constant3 = new Constant(2.3);
        ArithExpr constant4 = new Constant(5.0);
    }
}
```



Si vous avez trouvé les exercices corrigés en Java de Mr Joëlyk intéressants et utiles, pourquoi ne pas les partager avec d'autres personnes qui pourraient également en bénéficier ? Partagez ce lien sur les réseaux sociaux ou envoyez-le à vos amis et collègues. Vous pourriez aider quelqu'un à améliorer ses compétences en programmation ou à trouver des solutions à des problèmes complexes. N'oubliez pas que la connaissance doit être partagée pour grandir. Merci pour votre soutien et votre partage !

★★★★★ 1 vote. Moyenne 5 sur 5.

Nom *

Anti-spam *

Ajouter

^

TUTORIELS

NODE.JS	5	▼
JAVA	1	▼
Algorithme Avances	4	▼
KOTLIN	11	▼
PHP	15	▼
RUST	7	▼
ANDROIDE DEVELOPPEMENT	8	▼
JAVASCRIPT	7	▼
UML	2	▼
Langage Pascal	2	▼
SECURITE INFORMATIQUE	1	▼
IONIC	1	▼
RESEAU	4	▼
LINUX	2	▼
CLOUD COMPUTING	8	▼
LANGAGE C	10	▼
Systeme d'exploitation	3	▼
CSS	6	▼
C++	1	▼
SQL	2	▼
COMPLEXITE DES ALGORITHMES	0	▼
CRYPTOGRAPHIE	1	▼
RECHERCHE OPERATIONNELLE	1	▼
HTML (FACILE)		
CSS (FACILE)		

Quiz









 QUIZ PROGRAMMATION C	0
--	---





	QUIZ PHP	
	QUIZ HTML5	
	QUIZ JAVASCRIPT	
	QUIZ CSS	
	QUIZ ARCHITECTURE DES ORDINATEURS	

	Quiz PYTHON	
	QUIZ C PLUS PLUS	
	QUIZ REACTJS	
	QUIZ OCL	

Forum

A propos du site web	
Présentation des membres	
 Programmation Mobile	
Discussions générales	
Algorithmique	
C++	
Flutter	





A LA UNE !

	C'est quoi le C++ ? Par joel_vk Le 10/02/2023
	C'est quoi le phishing ? Par joel_vk Le 08/02/2023
	C'est quoi le ransomware ? Par joel_vk Le 29/01/2023

TRADUIRE LE SITE

 Français
 English
 Español



 Italiano
 Deutsch
 Nederlands
 Portuguesa

TAGS

python

javascript

c++

programmation


programme

PHP

language

Java


Météo




-20 °C Couvert

Min: -20 °C | Max: -20 °C | Vent: 4 kmh 132°

Derniers quiz



QUIZ OCL #01



QUIZ REACTJS #01



QUIZ C PLUS PLUS 02



☐ Rester connecté

[Créer un compte](#) | [Mot de passe perdu ?](#)

[Twitter du Panda](#)



Tweets de @Pandacodeur

Suivre



Rien à voir ici. Pour le moment.

Quand il publiera des Tweets, ils apparaîtront ici.

Examen Algorithme Avance & Complexite

Algorithmiques Avancés et Comp
--

RECHERCHE OPERATIONNELLE

Cours des Méthodes de Résoluti
--

RECHERCHE OPERATIONNELLE

Coloriage des Cartes - CSP
Exercices Methodes du Simplexe

CLOUD COMPUTING

Google Mashup
HEROKU
OpenStack
CloudStack
Google App Engine (GAE).
MICROSOFT AZURE
SERVEUR NAS
La Sécurité dans le Cloud

CLOUD COMPUTING

Exercice Miscrosoft Azure : Az
Exercice Miscrosoft Azure : Co
Exercice Miscrosoft Azure : Az
Travaux Dirigés : Sécurité des
EXERCICE Connaissances général
Travaux Dirigés : CloudStack
Travaux diriges Serveurs Nas
Travaux Dirigés : VIRTUALISATI
Exercice Google App Engine

[EXAMEN CLOUD COMPUTING](#)

[EXAMEN CLOUD COMPUTING SUJET 1](#)[EXAMEN CLOUD COMPUTING SUJET 2](#)[EXAMEN CLOUD COMPUTING SUJET 3](#)

[Ingénierie des connaissances Examen](#)

[Examen Ingénierie des connaissances](#)

[Examen JAVA](#)

[Examen Java 01](#)

[TP KOTLIN](#)

[Examen Fouille de Donnée](#)

[Examen Fouille de donnée Sujet](#)[Examen Fouille de donnée Sujet](#)[Examen Fouille de donnée Sujet](#)

[Examen Recherche opérationnelle](#)

[Examen recherche opérationnelle](#)

Copyright © 2019 - 2024 [PandaCodeur](#) Tous les droits sont réservés. Par Joël Yankam .



[Politique de confidentialité](#) | [Gestion des cookies](#)

