

Tutorat d'ouverture

Ce document a pour but de vous préparer au tutorat d'ouverture. Pour chaque paragraphe de l'énoncé de la problématique, il faut faire l'exercice :

- d'identifier des mots-clés
- de répondre aux questions posées

Il faut ensuite en déduire les étapes essentielles de résolution de la problématique. Pour profiter pleinement de la rencontre de tutorat, il est indispensable que vous ayez auparavant fait cet exercice avant la rencontre.

Comment peut-on formuler brièvement la problématique ?

.

.

.

.

PARAGRAPHES 1 et 2 :

Dans l'entreprise où vous travaillez, un langage de programmation de haut niveau est utilisé pour le développement d'applications devant être téléchargées dans des systèmes embarqués. Un compilateur a été développé dans le passé pour l'environnement cible qui a été jusqu'alors utilisé. Pour des raisons techniques et économiques, la direction de l'entreprise a décidé de changer d'environnement cible. Le compilateur jusqu'alors utilisé doit donc être abandonné et un nouveau compilateur doit être conçu.

Afin d'éviter d'avoir à l'avenir à concevoir un compilateur en entier chaque fois que l'entreprise change d'environnement cible, votre patron vous confie le mandat de concevoir et réaliser la partie dite frontale d'un compilateur. Votre patron vous propose de suivre les conseils d'un collègue expérimenté en compilation mais qui n'a pas le temps de réaliser lui-même la tâche qui vous est demandée. Après une première réunion avec votre collègue et suite à ses conseils, vous vous êtes fixé pour premier objectif de produire un prototype qui effectue l'analyse syntaxique d'expressions arithmétiques. C'est l'objectif qui sera considéré durant cette APP.

Q Que faut-il faire pour résoudre le problème ?

R ...

Produire un prototype qui effectue l'analyse syntaxique d'expressions arithmétique

PARAGRAPHE 3 :

Les expressions arithmétiques sont constituées d'opérandes, d'opérateurs et de parenthèses ouvrantes et fermantes. Les opérandes sont de type entier. Les opérateurs sont l'addition (+), la soustraction (-), la multiplication (*) et la division (/). Les priorités des opérateurs sont comme suit : + et - ont même priorité; * et / ont même priorité; + et - sont moins prioritaires que * et /. Les quatre opérateurs sont associatifs à droite. Noter que les priorités peuvent être forcées en utilisant des parenthèses.

Q De quoi est constituée une expression arithmétique ?

R Opérande, opérateur, parenthèses ouvrantes et fermantes

Q Qu'est-ce que la syntaxe ?

R

Q Que signifie la priorité d'opérateur ?

R ...

Q Que signifie l'associativité d'opérateur (à droite et à gauche) ?

R ...

Q Comment fait-on pour forcer une priorité d'opérateur ?

R

PARAGRAPHE 4 :

Un opérande peut avoir une des deux formes suivantes :

- Séquence de longueur non nulle constituée des chiffres 0 à 9.
- Lettre majuscule suivie d'une séquence possiblement vide constituée de caractères tels que :
 - chaque caractère est une lettre majuscule ou minuscule ou un tiret bas « _ » (underscore),
 - un tiret bas ne peut, ni être suivi par un autre tiret bas, ni être le dernier élément de la séquence.

Exemples corrects : AbCdE ZyXw_v_Ut

Exemples incorrects : aBcDe ZyXw__v_Ut ZyXw_v_Ut_

Voici un exemple d'expression arithmétique : $(X_a + Y_b) * Z_c / 59$. Les opérations y sont effectuées dans l'ordre suivant : l'addition, la division, et enfin la multiplication.

Q Quelle est la syntaxe d'une expression arithmétique ?

R ...

Q Donner un exemple d'expression arithmétique

R ...

PARAGRAPHE 5 :

Après une seconde réunion avec votre collègue et d'un commun accord avec lui, vous avez décidé de commencer à concevoir un module particulier appelé analyseur lexical et qui sera utilisé par l'analyseur syntaxique. L'approche choisie par votre collègue et votre patron, consiste à spécifier les unités lexicales à l'aide d'expressions régulières (ER) et d'automates à états finis (plus brièvement : automates); un analyseur lexical peut ensuite être dérivé d'une manière systématique à partir des automates qui auront été produits. Il est décidé de ne pas utiliser Lex ou tout autre outil logiciel similaire.

Q Quelle est la tâche d'un analyseur lexical ?

R ...

Q Quelle est l'approche choisie pour concevoir l'analyseur lexical ?

R ...

Q Quelle est l'utilité des expressions régulières (ER) et des automates ?

R ...

Q Comment les analyseurs lexical et syntaxique interagissent-ils ?

R ...

PARAGRAPHE 6 :

Une fois l'analyseur lexical conçu, vient le moment de concevoir l'analyseur syntaxique. Après une recherche, vous avez constaté qu'il existe deux catégories de méthodes d'analyses syntaxiques : les méthodes ascendantes et les méthodes descendantes. Toutes ces méthodes nécessitent de modéliser la syntaxe des expressions à analyser par une grammaire. Afin d'obtenir un compilateur simple dont la conception ne nécessite pas l'utilisation d'outils logiciels spécialisés, la méthode qui vous a été suggérée, et qui a été choisie, est appelée méthode de la descente récursive (MDR).

Q Qu'est-ce qu'une grammaire et à quoi sert-elle ?

R

Q Quels sont les deux types de méthodes d'analyse syntaxique ?

R ...

Q Quel est le principe de base de chacune d'elles ?

R ...

Q Quel est le principe de base de la MDR ?

R ...

PARAGRAPHE 7 :

Votre collègue vous avise que pour que la MDR soit applicable, il faut que la grammaire qui spécifie la syntaxe des expressions arithmétiques soit d'un certain type et respecte certaines contraintes. Pour que vous soyez en mesure de construire une grammaire adéquate, il vous conseille d'étudier les méthodes d'analyse syntaxique descendante, en particulier la méthode LL(1). La grammaire doit aussi avoir une forme qui garantit le respect des priorités des opérateurs.

Q Qu'est-ce qu'une méthode LL ?

R ...

Q Est-ce que la méthode descendante est applicable pour n'importe quelle grammaire ?

R ...

Q Quel est le lien entre une méthode descendante, la méthode LL(1) et la MDR ?

R ...

Q En plus d'être LL(1), quelle autre contrainte doit respecter la grammaire ?

R ...

PARAGRAPHE 8 :

En plus de déterminer si une expression arithmétique est syntaxiquement correcte, une tâche importante d'un analyseur syntaxique est la construction d'un arbre syntaxique abstrait (*Abstract Syntax Tree*, AST) qui modélise la structure syntaxique de l'expression arithmétique. Après construction de l'AST et afin de vérifier que cette construction est correcte, vous devez lire l'AST, c-à-d. traduire la structure de données de l'AST sous une forme lisible (sur écran ou sur fichier) et aussi déterminer son expression postfix. Pour souligner une utilité particulière de l'AST, vous devez aussi évaluer l'AST, c-à-d. évaluer le résultat du calcul de l'expression arithmétique (si tous les opérandes sont des valeurs).

Q Quelles sont les tâches essentielles de l'analyseur syntaxique ?

R ...

Q En quoi consiste la lecture d'un AST ?

R ...

Q Donner une utilité de la lecture d'AST

R ...

Q En quoi consiste l'évaluation d'un AST ?

R ...

Q Donner une utilité de l'évaluation d'AST

R ...

Q Comment fait-on la lecture et l'évaluation d'un AST?

R ...

Q Comment construire l'expression postfix correspondant à un AST ?

R ...

Q Qu'a-t-on besoin de préparer pour pouvoir construire, lire et évaluer un AST ?

R ...

Q Que faut-il faire pour valider le produit logiciel réalisé ?

R ...

PARAGRAPHE 9 :

Pour bénéficier des avantages de la programmation orientée-objet, vous avez pensé utiliser un des deux langages orientés-objet que vous connaissez : C++ et Java. Comme l'aspect temps-réel n'est pas primordial et pour avoir une bonne portabilité, la direction de l'entreprise a choisi Java. De plus, votre collègue a préparé des squelettes de classes java à utiliser (voir page Web de l'APP6).

Q Quel langage de programmation sera utilisé ?

R ...

Q Quelles sont les raisons qui ont mené au choix de Java plutôt qu'un autre langage ?

R ...

Étapes de résolution de la problématique

À partir des réponses apportées aux différentes questions, établissez les étapes de résolution de la problématique :

1. ...

2. ...

3. ...

4. ...