1

TD1

Main Author(s): Brest
But du TD: Structuration d'expression Smalltalk: objet, message. Les littéraux du langage. Messages unaires, binaires et à mots clès. Les régles de priorités.

1.1 Rappels

1. Repondez aux questions suivantes :
   - Lister les différents littéraux
   - Lister trois objets qui ne sont **pas** des littéraux
   - Lister les catégories de messages. Pour chacun d'entre elles donner le nombre de paramètres, et les signes distinctifs.
   - Lister les règles de priorité entre messages.
2. Si on vous donne le code suivant, trouvez les variables, les littéraux, les messages, les receveurs et les paramètres des messages, donner l'ordre d'évaluation des messages.

```
| toto titi z |
toto := 34 glop glop. titi := 23 +- toto.
toto pas: #te glop:#ti pas:'tte''te' glop: $r r.
z := toto = titi
schmilblick.
((titi z z: z z: #(z)) z: 12) z: 'titi'
```

On rappèle que l'ensemble des messages binaires est connu par le système. Un message binaire est composé de un ou deux caractères parmi les suivants: `!%&*+,-/<=>?@\~`

1.2 L'objet: c'est du gateau!

Dans cet exercice on souhaite décrire une opération courante (une recette de cuisine) en utilisant le formalisme objet message receveur de SmallTalk. Soit la recette suivante (crumble simplifié sans garniture) : "*Dans un plat disposez un poids égal de farine, beurre et sucre (150 g), mélangez, puis faite cuire à thermostat 6 pendant 40 minutes. Le gâteau est prêt*".

On donne les indications suivantes :

- Classe PlatAGateau :
  - message pour créer une instance de PlatAGateau : new
  - message pour ajouter un ingrédient dans une instance de PlatAGateau : ajouter: avec en argument l'ingrédient à ajouter
  - message pour mélanger le contenu d'une instance de PlatAGateau : mixer

  - message pour cuire une instance de PlatAGateau : cuireThermostat: duree: avec en premier argument la valeur du thermostat et en deuxième argument la durée en minutes
  - message pour démouler le contenu d'une instance de PlatAGateau : démouler qui renvoi en retour une instance de la classe Gateau.

- Classe Beurre :
  - message pour créer une instance de Beurre : new: avec en argument le poids en grammes

- Classes Farine et Sucre :
  - Mme message de création que la pour la classe Beurre

1. En utilisant les Classes et les méthodes fournies, écrivez en Smalltalk la recette. Vous ferez en sorte de ne pas mettre plus d'un message par ligne. Vous pouvez utiliser des variables locales avec des affectations.
2. Ecrivez l'équivalent de ce code sur une seule ligne de manière à avoir directement un gateau. Prenez soin de correctement positionner des parenthèses si nécessaire.

1.3 Priorité de messages et variable temporaire

1.3.1 Fonctions trigonométriques

Un nombre (en radians) comprend les messages correspondant aux fonctions trigonométriques sin cos tan arcSin arcCos arcTan.

On convertit un nombre de Degré à Radian en lui envoyant le message degreesToRadians.

Calculer (à l'aide d'une fonction trigonométrique) le côté d'un carré dont la diagonale mesure 1.41421 mètres.

1.3.2 Maximum

La méthode max: unAutreNombre appliqué à un nombre renvoie le plus grand des deux nombres.
Exemple : 2 max: 6 renvoie 6.

1. Calculer le maximum de la somme des couples formés à partir de trois variables a b c contenant des valeurs quelconques, en stockant dans des variables les résultats intermédiaires.
2. Calculer le maximum de la somme des couples formés à partir de trois variables a b c contenant des valeurs quelconques **sans utiliser de variables intermédiaires**

1.3.3 Conversion

Le message binaire // correspond à l'opération de division entière. Le message binaire
correspond à l'opération de modulo (reste de la division entière).
On utilise ces messages pour convertir en binaire un nombre hexadécimal (de 0 à 15 en base 10, de 0000 à 1111 en base 2).

En effectuant une série de divisions entières, les chiffres binaires sont obtenus **de la droite vers la gauche** grâce au reste de la division entière (le modulo).

Appliquer cet algorithme pour convertir le nombre hexadecimal $(F)_{16}$ (valeur décimale 15) en base 2.

1.4 Tableau et point

Les messages à mots clés at: unIndex et at: unIndex put:uneValeur permettent l'accès en lecture ou écriture d'un élément se trouvant dans un tableau. Les messages x, y, x: uneValeur et y: uneValeur permettent l'accès en lecture ou écriture de l'abscisse ou de l'ordonnée d'un point.

1.4.1 Analyse de code

Repérer dans les lignes de code suivantes, les messages unaires, binaires et à mots clés. En déduire l'évaluation des expressions et des portions de code.

```
| aPoint |
aPoint:= Point x:2 y:1.
aPoint x: aPoint x * 2.
aPoint
```

```
| x tab |
tab := #(4 5.0 'toto' 1111).
x := tab at:1.
tab at: 1 put: (Fraction  numerator: x*2 denominator: 7 + x negated).
tab
```

1.4.2 Exercices sur les tableaux et les points

1. Multiplier par 2 le 2 ème élément d'un tableau,
2. Remplacer la valeur du 2 ème élément d'un tableau par son opposé.
3. Remplacer la valeur du 3 ème élément par la valeur du 2 ème élément.
4. Remplacer la valeur du 3 ème élément par la somme des 2 ème et 3ème (ancienne valeur) éléments.

1.4.3 Pour aller plus loin

1. Le 2 ème du tableau étant une fraction, remplacer cette fraction par la fraction inverse dans le tableau.
2. On désire définir le barycentre de points. La notion de barycentre s'applique à un ensemble de points effectés par une masse. Soit p_1,p_2,. . ., p_n un ensemble de points affectés des masses m_1,m_2,. . ., m_n, on appelle barycentre le point p défini de la manière suivante:

$$m.p = \sum_{i=1}^{n} m_i.p_i \; avec \; m = \sum_{i=1}^{n} m_i$$

   Définir des expressions smalltalk permettant de définir le barycentre p, on prendra $n = 4$ et les valeurs des masses et des points seront donnés dans des tableaux que l'on définira.