

1 Une calculette avec le schéma “Etat” (State) - Version 1

Réaliser avec l’environnement Smalltalk la classe `Calculette` à trois états donnée en Java dans le cours (<http://www.lirmm.fr/~dony/notesCours/cpatterns.pdf>). Ajoutez-y un attribut supplémentaire (`acc2` pour la suite du TD).

- Avec cette calculette on peut écrire la méthode `exemple1` (figure 1) et l’exécuter dans un *workspace* :

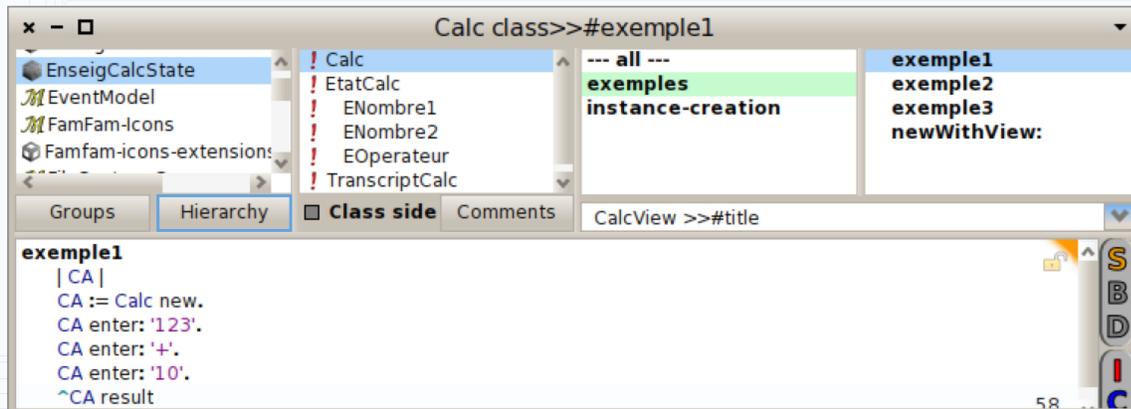


FIGURE 1 – Fonctionnement de base.

- n instances de la classe peuvent co-exister sans soucis.
- Définir la classe `Calculette`, avec ses 4 attributs (variables d’instances). Finissez par “accept”. Dans le menu contextuel de la classe `Calculette`, choisissez “analyse>generate Initialize method”. Une nouvelle méthode a été générée sous le protocole “initializing”. Complétez là pour initialiser les attributs aux bonnes valeurs. Finissez par *Accept*. Dans le même menu, vous pouvez utiliser “create inst var accessors” (pour générer des accesseurs en lecture) pour les attributs qui le nécessitent (attention tous les attributs sont cochés par défaut). Pour un attribut x les accesseurs sont usuellement nommés x (lecture) et $x:$ (écriture). Si vous souhaitez changer le nom d’un attribut, utiliser le menu “refactoring” de la classe. Plus largement, il y a des menus “refactoring” à tous les niveaux.
- Définissez les autres classes de façon équivalente.
- Pour appliquer l’opérateur, inutile de faire un “case”. Vous pouvez utiliser `perform:with:` qui permet un envoi de message calculé : `acc1 perform: op with: operand2`. C’est la même idée que l’appel de fonction calculée de Scheme (par exemple) : `(apply (car '(* +)) '(2 3))` vaut 6.

2 Jeux de Test

Faites une sous-classe de `TestCase` pour tester la calculette, comme indiqué en <http://pharo.gforge.inria.fr/PBE1/PBE1ch8.html>.

3 Une calculette étendue

Faites en sorte que la calculette permette d’utiliser n’importe quelle opération à 2 paramètres sur les nombres :

```
1 CA enter: '36'.
2 CA enter: 'gcd:'. "PGCD"
3 CA enter: '48'
4 CA result --> 12
```

4 Une calculette avec affichage textuel - (pattern MVC)

La vue est un écouteur de la calculette (son modèle). A chaque fois que la calculette change, elle doit le dire, sa vue sera alors prévenue. Ceci permet par exemple d'afficher les entrées réalisées et le résultat du calcul (figure 2).

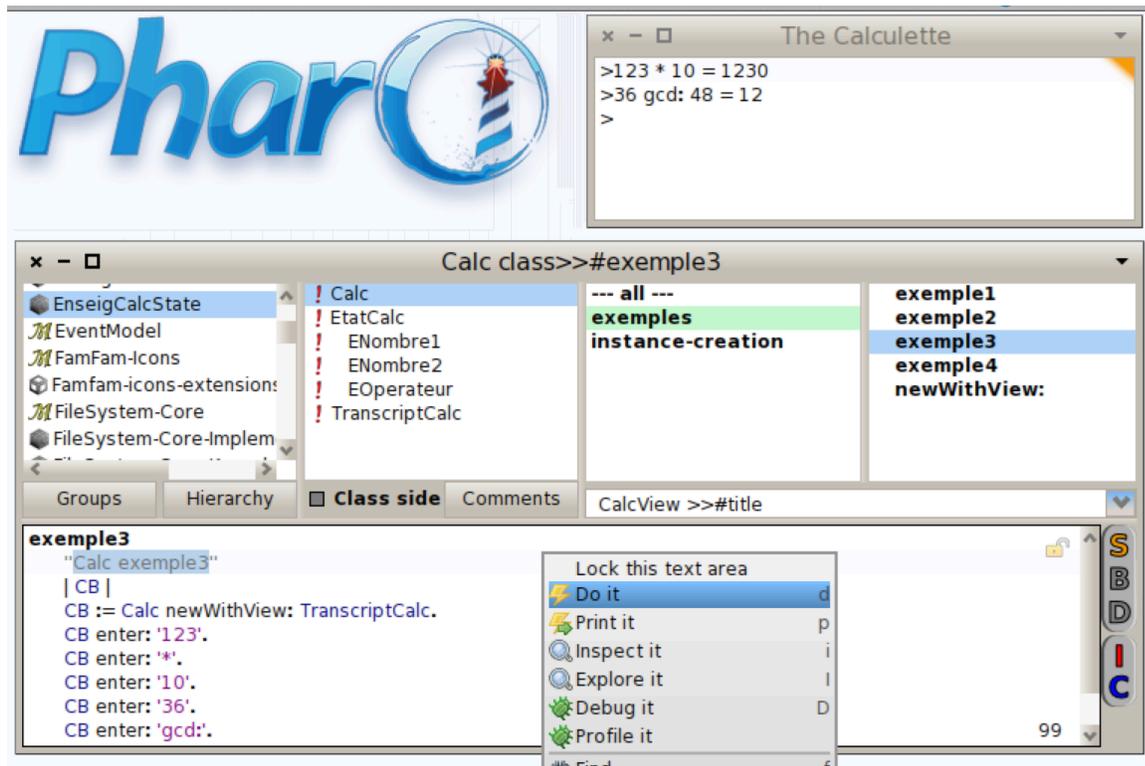


FIGURE 2 – Calculette avec Afficheur texte ("The Calculette").

5 Une calculette avec affichage graphique - (MVC également)

L'étape suivante est de réaliser une jolie calculette graphique. Si vous avez envie de vous lancer, vous trouverez un exemple pas très éloigné avec le fichier ci-dessous. Copiez le dans votre répertoire Smalltalk puis charger le via l'outil Tools>FileBrowser.

<http://www.lirmm.fr/~dony/enseig/IL/TicTacToe.st>.

6 Environnement, quelques indications

- Menu Général : clic sur fond d'écran. Les items essentiels dans un premier temps sont *System Browser*, *Playground*, *Tools-Transcript* si vous voulez afficher des messages (`Transcript show: 'Hello World'; cr.`), *Tools-FileBrowser* pour récupérer vos futurs travaux.
- Menu Contextuel d'un élément donné : s'obtient avec le bouton droit de la souris pointée l'élément, ou avec la combinaison de touches (Ctrl-clic) ou équivalent.
- Les raccourcis claviers pour éviter d'ouvrir systématiquement les menus sont obtenus par combinaison d'une touche de contrôle (Ctrlm ou Cmd ou ...) et d'une lettre.

7 La calculette avec State - version 2 - changement de classe avec *adoptInstance*

Voir les notes de cours <http://www.lirmm.fr/~dony/notesCours/cpatterns.pdf> et la présentation de cette solution alternative d'implantation du schéma *State*.