Maîtrise Informatique 1999-2000 — Bordeaux I

TFP

TD 10 - Common Lisp Object System

Premiers pas en programmation objet en Lisp

Considérer la classe virtuelle (i.e. non-instanciable) et sans créneau:

```
(defclass polygon () ())

(defgeneric number-of-sides ((f polygon))
  (:documentation "returns the number of sides of a polygon"))

(defgeneric perimetre ((f polygon)))

(defgeneric get-sides ((f polygon))
  (:documentation "returns the list of sides of a polygon"))

(defmethod perimetre ((f polygon))
  (reduce #'+ (get-sides f)))
```

1. Parmi les polygones que l'on souhaite traiter, considérer le cas des triangles. Tenir compte du fait que le calcul de l'aire et du perimètre d'un triangle peut être simplifié pour certaines formes régulières comme les triangles isocèles, rectangles, et équilatéraux.

   Préciser les créneaux, méthodes, constructeurs et relations d'héritage de votre modèle.

2. Ajouter des assertions contrôlant la cohérence des triangles construits. Considérer également la possibilité de modifier une valeur d'un créneau mutable tout en maintenant cette cohérence.

3. Ajouter dans les créneaux l'attribut "`surface`" tout en maintenant l'interface ci-dessus.

4. A quel moment la surface se (re)calcule-t-elle ?

5. Voici la classe correspondant aux ronds:

   ```
   (defstruct color (r 0) (g 0) (b 0))

   (defclass rond ()
     ((rayon :initarg :rayon
             :reader rayon)
      (couleur :initarg :couleur
               :reader couleur)))
   ```

   (a) Ecrire le constructeur de rond (rouge par défaut).

   (b) Définir les méthodes nécessaires pour pouvoir faire évaluer des expressions du genre:

   ```
   (let ((l (list (make-scalene 3 5 2)
                  (make-isocele 3 4)
                  (make-triangle-rectangle 3 4)
                  (make-rond 10 :couleur (make-color)))))
     (mapcar #'perimetre l))
   ```