Contrôle continu de Bases de Données Objet

Maîtrise d'Informatique - 9 Novembre 2000

Documents autorisés. (durée 1h)

On considère le schéma suivant :

```
class Personne public type tuple (
                nom : string,
                prenom : string,
                datenais: Date)
end;
class Docteur inherit Personne public type tuple(
                qualification: integer,
                service: Service,
                dateint: Date,
                superieur: Docteur,
                traitement: integer,
                vacation: integer,
                consultations: list(Consult),
                conges: list(Conge))
end;
class Patient inherit Personne public type tuple(
                insee: string,
                profession: string,
                consultations: list(Consult))
end;

class Service inherit Object public type tuple(
                nom: string,
                localisation: string,
                batiment: string,
                tlphn: string)
end;
class Consult inherit Object private type tuple(
                patient: Patient,
                docteur: Docteur,
                dateclt: Date)
end;
class Conge inherit Object public type tuple(
                docteur:Docteur,
                datedeb:Date,
                datefin:Date)
end;
```

On dispose des objets nommés :

name Lesdocteurs : unique set(Docteur);
name Lespatients : unique set(Patient);

1 Requêtes en OQL

1. Affichez tous les patients (nom et profession).
2. Donnez le nom et la localisation de tous les services.
3. Quel est le nombre total de consultations effectuées dans l'hôpital?
4. Quel est le nom du docteur qui a effectué le moins de consultations?
5. Classez les docteurs selon leur nombre de départs en congés.

6. Écrire une requête qui réponde vrai ou faux à la question : existe-t'il un patient qui a effectué toutes ses consultations dans le même service?

7. Regroupez les patients selon leur profession en utilisant un group by (résultat sous la forme d'un ensemble de couples (nom de profession, ensemble de nom de patients)).

2 Le langage O_2C

La classe **Date** est supposée implémentée avec

- une méthode d'initialisation admettant 3 arguments entiers : le jour, le mois et l'année
- et une méthode **difference(d:Date) : integer** qui calcule le nombre de jours séparant la date sur laquelle est appliquée la méthode de la date fournie en argument.

Le patient Alfred Hitchcock prend rendez-vous le 29 avril 1980 avec le docteur Cary Grant (déjà présent dans la base de données, (né le 18 janvier 1904)). Alfred Hitchcock n'est jamais venu dans cet hôpital auparavant.
On supposera que le mode de transaction courant est **Lecture seule**. Si vous utilisez des méthodes vous devez les **implémenter préalablement** (à l'exception des méthodes données dans la classe Date). Notez bien que la classe Consult est **privée**.

1. Insérez Alfred dans la liste des Patients. Alfred Hitchcock est né le 13 août 1899, sa profession est cinéaste.

2. Insérez ensuite sa consultation.

3. Créez une méthode permettant de calculer l'âge moyen des patients d'un docteur qui prend la date du jour en paramètre. Écrivez une requête utilisant cette méthode permettant d'obtenir le nom et l'âge moyen des patients pour tous les docteurs de la base. (Si votre méthode est inutilisable en mode requête modifiez-la pour qu'elle le soit).