

# Quel médecin 2

## TP du module 04 - Les associations

### Proposition de solution

1 - Ajout d'adresse pour les médecins généralistes et les patients :

```
package fr.eni.ecole.queleMedecin.bo;

import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.time.format.FormatStyle;

/**
 * Classe modélisant un patient
 *
 * @date 17 oct. 2018
 * @version P00 - V2.0
 * @author hboisgontier
 */
public class Patient {
    private String nom;
    private String prenom;
    private String numeroDeTelephone;
    private char sexe;
    private long numSecu;
    private LocalDate dateNaissance;
    private String commentaires;
    private Adresse adresse;

    /**
     * Constructeur
     *
     * @param nom
     *         nom du patient
     * @param prenom
     *         prénom du patient
     * @param numeroDeTelephone
     *         numéro de téléphone du patient
     * @param sexe
     *         sexe du patient : 'F' pour Féminin ou 'M' pour Masculin
     * @param numSecu
     *         numéro de Sécurité sociale du patient
     * @param dateNaissance
     *         date de naissance du patient
     * @param commentaires
     *         commentaires associés à ce patient
     *         (allergies, antécédents médicaux...)
     * @param adresse
     *         adresse postale du patient
     */
    public Patient(String nom, String prenom, String numeroDeTelephone, char sexe,
        long numSecu, LocalDate dateNaissance, String commentaires, Adresse adresse) {
```



```

        this.nom = nom.toUpperCase();
        this.prenom = prenom;
        this.numeroDeTelephone = numeroDeTelephone;
        this.sexe = sexe;
        this.numSecu = numSecu;
        this.dateNaissance = dateNaissance;
        this.commentaires = commentaires;
        this.adresse = adresse;
    }

    /**
     * Affiche sur la console sous la forme :
     * NOM Prénom
     * Téléphone : XXXXXXXXXX
     * Sexe : Féminin ou Masculin
     * Numéro de Sécurité sociale : XXXXXXXXXXXXXXXX
     * Date de naissance : XX mois XXXX
     * Commentaires : XXXXXXXXXXXXXXXX ou [aucun commentaire]
     * Adresse :
     */
    public void afficher() {
        System.out.printf("%s %s\nTéléphone : %s\nSexe : %s\n" +
            "Numéro de Sécurité sociale : %d\nDate de " +
            "naissance : %s\nCommentaires : %s\nAdresse : %n",
            this.nom, this.prenom, this.numeroDeTelephone,
            this.sexe == 'F' ? "Féminin" : "Masculin",
            this.numSecu,
            this.dateNaissance.format(DateTimeFormatter.ofLocalizedDate(FormatStyle.LONG)),
            this.commentaires != null ? this.commentaires : "[aucun commentaire]");
        this.adresse.afficher();
    }
}

```

---

```

public class MedecinGeneraliste {

    private String nom;
    private String prenom;
    private String numeroDeTelephone;
    private static int tarif = 25;
    private Adresse adresse;

    /**
     * Constructeur : crée une instance de Medecin
     *
     * @param nom
     *         nom du médecin
     * @param prenom
     *         prénom du médecin
     * @param numeroDeTelephone
     *         numéro de téléphone
     * @param adresse
     *         adresse postale du patient
     */
    public MedecinGeneraliste(String nom, String prenom, String numeroDeTelephone,
                               Adresse adresse) {

        this.nom = nom.toUpperCase();
        this.prenom = prenom;
    }
}

```

```

        this.numeroDeTelephone = numeroDeTelephone;
        this.adresse = adresse;
    }

    ...

    /**
     * Affiche sur la console sous la forme :
     * NOM Prénom
     * Téléphone : XXXXXXXXXX
     * Tarif : XX€
     * Adresse :
     * Complément
     * XXbis rue XXXXXXXXXX
     * 00000 XXXXXXXXXXXXXX
     */
    public void afficher() {
        System.out.printf("%s %s\nTéléphone : %s\nTarif : %d€\nAdresse :%n",
                           this.nom, this.prenom,
                           this.numeroDeTelephone, MedecinGeneraliste.tarif);
        this.adresse.afficher();
    }
}

```

Pour créer l'association unidirectionnelle entre Patient et Adresse et entre MedecinGeneraliste et Adresse, il suffit d'ajouter un attribut d'instance dans les classes Patient et MedecinGeneraliste.

2 - Ajout de créneaux horaires pour les médecins :

```

package fr.eni.ecole.que1Medecin.bo;

import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.time.format.FormatStyle;

/**
 * Classe modélisant un créneau horaire pour un médecin
 *
 * @date 18 oct. 2018
 * @version P00 - V2.0
 * @author hboisgontier
 */
public class Creneau {
    private LocalDateTime heureDebut;
    private int duree;
    private MedecinGeneraliste medecin;

    /**
     * Constructeur
     *
     * @param heureDebut
     *         heure de début du créneau
     * @param duree
     *         durée du créneau en minutes
     * @param medecin
     *         médecin possédant ce créneau
     */
}

```

```

    */
    public Creneau(LocalTime heureDebut, int duree, MedecinGeneraliste medecin) {
        this.heureDebut = heureDebut;
        this.duree = duree;
        this.medecin = medecin;
        medecin.ajouterCreneau(this);
    }

    /**
     * Affiche un créneau horaire sous la forme :
     * 00:00 - 00:00 (XX minutes)
     */
    public void afficher() {
        System.out.printf("%s - %s (%d minutes)%n",
            this.heureDebut.format(DateTimeFormatter.ofLocalizedTime(FormatStyle.SHORT)),
            this.heureDebut.plusMinutes(this.duree).format(
                DateTimeFormatter.ofLocalizedTime(FormatStyle.SHORT)), this.duree);
    }

    /**
     * Getter pour medecin.
     * @return the medecin
     */
    public MedecinGeneraliste getMedecin() {
        return medecin;
    }
}

```

---

```

public class MedecinGeneraliste {
    public static final int MAX_CRENEAUX = 15;

    private String nom;
    private String prenom;
    private String numeroDeTelephone;
    private static int tarif = 25;
    private Adresse adresse;
    private Creneau[] creneaux = new Creneau[MAX_CRENEAUX];

    ...

    /**
     * Affiche sur la console sous la forme :
     * NOM Prénom
     * Téléphone : XXXXXXXXXX
     * Tarif : XX€
     * Adresse :
     * Complément
     * XXbis rue XXXXXXXXXX
     * 00000 XXXXXXXXXXXXXX
     */
    public void afficher() {
        System.out.printf("%s %s\nTéléphone : %s\nTarif : %d€\nAdresse : %n",
            this.nom, this.prenom,
            this.numeroDeTelephone, MedecinGeneraliste.tarif);
        this.adresse.afficher();
        System.out.println("Créneaux :");
    }
}

```



```

        for(int i=0; i<this.creneaux.length; i++)
            if(this.creneaux[i]!=null)
                this.creneaux[i].afficher();
    }

    /**
     * @param creneauAAjouter
     */
    void ajouterCreneau(Creneau creneauAAjouter) {
        if (this != creneauAAjouter.getMedecin()) {
            System.err.println("Ce créneau ne peut être associé à ce médecin " +
                               "car il est déjà associé à un autre");
        } else {
            int pos = 0;
            while (pos < this.creneaux.length && this.creneaux[pos] != null)
                pos++;
            if (pos == this.creneaux.length)
                System.err.println(
                    "Trop de créneaux sont affectés à ce médecin");
            else
                this.creneaux[pos] = creneauAAjouter;
        }
    }
}

```

Une association bidirectionnelle est plus compliquée à réaliser puisqu'il faut un attribut supplémentaire dans les deux classes participant à l'association alors que pour une association unidirectionnelle, un seul suffit. De plus, il faut s'assurer d'avoir des informations cohérentes avec ces deux attributs.

Seul le constructeur de la classe Creneau modifie la valeur de l'attribut medecin. Une fois cette modification effectuée, il fait appel à la méthode ajouterCreneau(). Cette méthode est la seule à modifier l'attribut creneaux et a une visibilité package ce qui limite les appels possibles à cette méthode. Elle vérifie que le médecin associé au créneau correspond bien au créneau.

### 3 - La prise de rendez-vous :

```

package fr.eni.ecole.quelMedecin.bo;

import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.time.format.FormatStyle;

/**
 * Classe modélisant un rendez-vous pris entre un patient et un médecin
 * sur l'un de ses créneaux pour une date donnée
 * @date 19 oct. 2018
 * @version POO - V2.0
 * @author hboisgontier
 */
public class RendezVous {
    private Creneau creneau;
    private Patient patient;
    private LocalDate date;
}

```

```

/**
 * Constructeur
 * @param creneau créneau horaire du médecin pour le rendez-vous
 * @param patient patient ayant pris rendez-vous
 * @param date date du rendez-vous
 */
public RendezVous(Creneau creneau, Patient patient, LocalDate date) {
    this.creneau = creneau;
    this.patient = patient;
    this.date = date;
}

/**
 * Affiche sous la forme :
 * Rendez-vous du 00 XXXXXX 0000 00:00 - 00:00 (00 minutes)
 * avec le Dr Nom
 * pour NOM Prénom
 * Téléphone : XXXXXXXXXX
 * Sexe : Féminin ou Masculin
 * Numéro de Sécurité sociale : XXXXXXXXXXXXXXXX
 * Date de naissance : XX mois XXXX
 * Adresse :
 * Commentaires : XXXXXXXXXXXXXXXX ou [aucun commentaire]
 */
public void afficher() {
    System.out.printf("Rendez-vous du %s ",
        this.date.format(DateTimeFormatter.ofLocalizedDate(FormatStyle.LONG)));
    this.creneau.afficher();
    System.out.printf("avec le Dr %s\npour ",
        this.creneau.getMedecin().getNom());
    this.patient.afficher();
}
}

```

Cette nouvelle classe `RendezVous` a une association vers la classe `Creneau` et une autre vers la classe `Patient`.