



USTHB/Faculté d'informatique

Gestion d'un Cabinet Médical

POO PROJECT 2024



by Kerrouche Myriam Sara
Bentaifour Yamina
Djaib Khadidja
Benabdellah Hajder



Notre application comprendra les fonctionnalités suivantes



Gestion et Suivi du Dossier Médical

Consiste à mettre à jour le dossier médical du patient au cours de son traitement et de ses visites



Gestion des Rendez-vous

L'organisation des rendez-vous selon les horaires libres et disponibles.

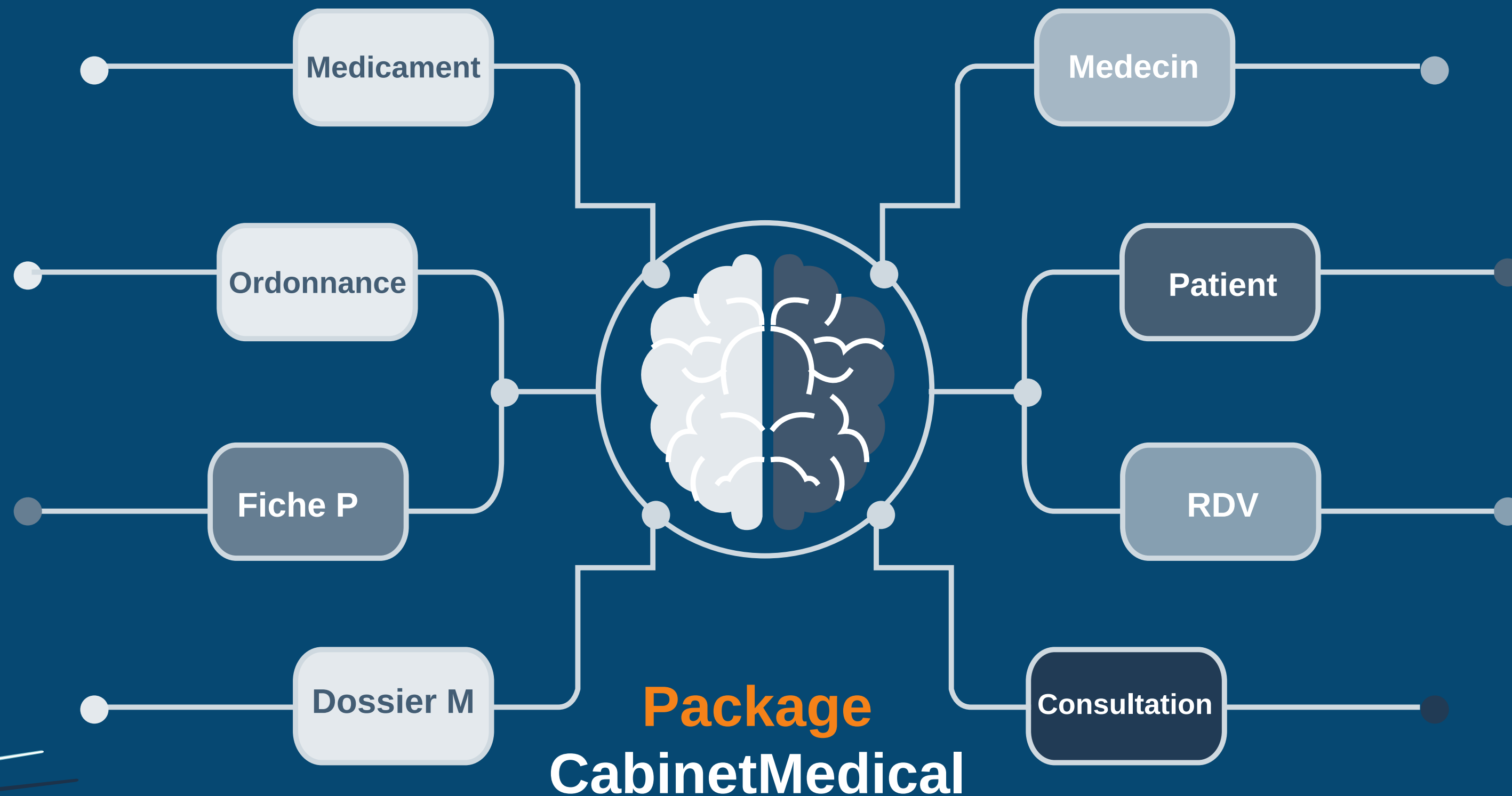


Gestion des Fiches Patients

Gère la prise de notes sur l'état du patient et collecte toutes les informations sur son cas.



Après avoir compris le déroulement de chaque fonctionnalité, nous avons pu distinguer les classes primaires



Visibilité

Les classes sont déclarées **Public** et les attributs
sont **Private**.

Un seul package comportera l'ensemble des classes

Medecin du cabinet

```
public class Medecin {  
    private String nom;  
    private String prenom;  
    private String specialité;  
    private String motdepasse;  
  
    public Medecin() {  
        this.nom = "BENIA";  
        this.prenom = "LAMIR";  
        this.specialité = "Ophtalmologue";  
        this.motdepasse = "droitacceslimite";  
    }  
  
    public String getNom() {  
        return nom;  
    }  
  
    public String getPrenom() {  
        return prenom;  
    }  
  
    public String getSpecialité() {  
        return specialité;  
    }  
  
    public String getMotdePasse() {  
        return motdepasse;  
    }  
}
```



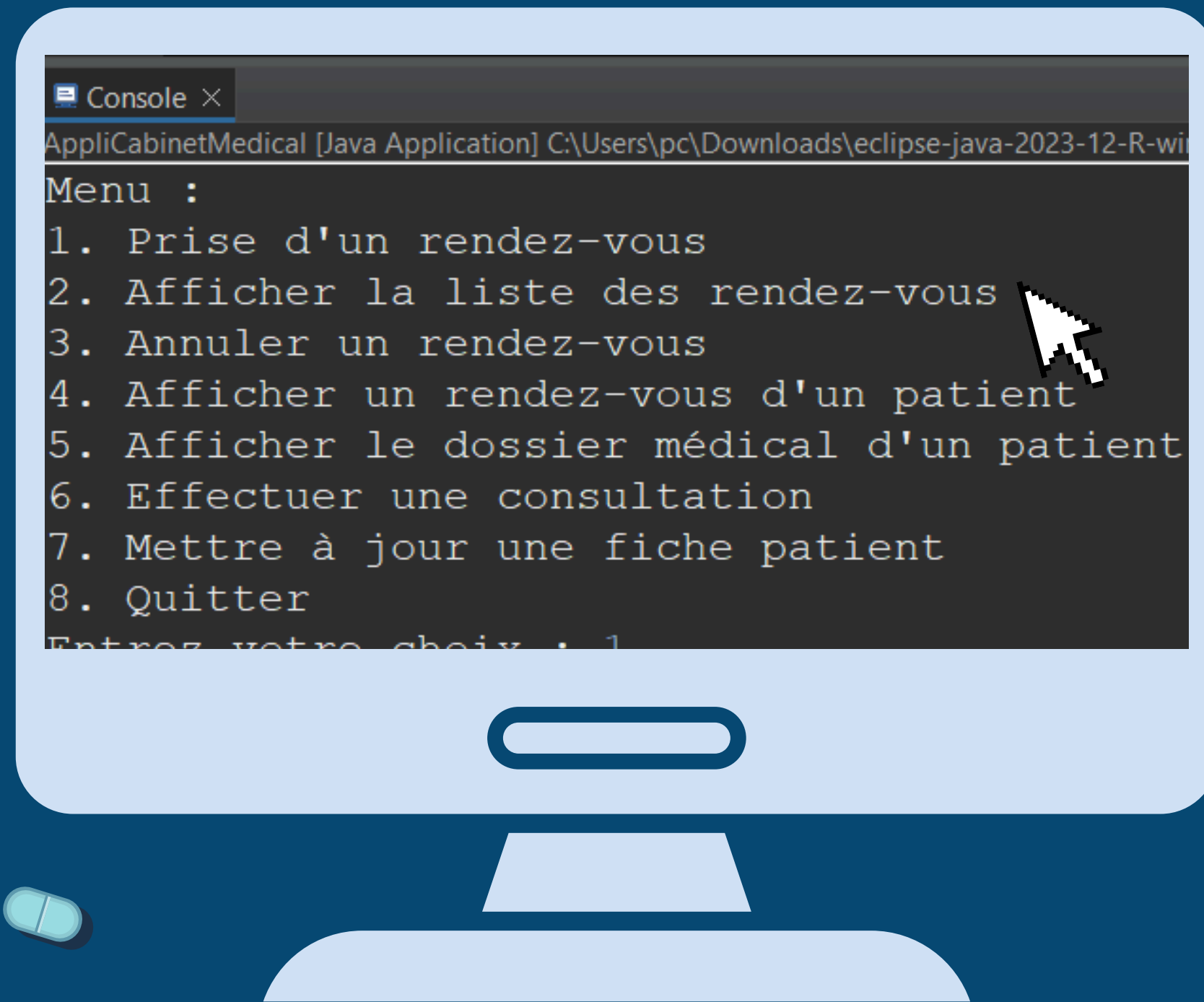
Les informations du médecin seront initialisées une fois qu'on créera un seul objet médecin.

il sera le seul object de son type. (unique instance)

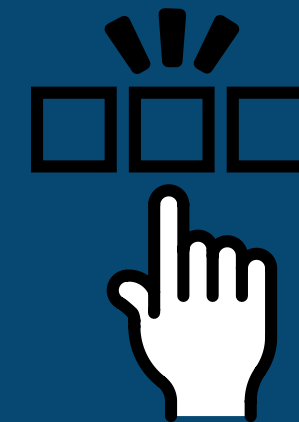
Simulation d'une interface

le prog **Main** (Application)

principe d'encapsulation



Le **Main** est sous forme d'un **menu à choix multiples** qui jouera le rôle d'une interface logicielle pour l'utilisateur



analyse du contenu d'un dossier médical et d'une fiche patient

Informations du Patient

(attribut de type patient)



<List> Ordonnance
<List> Fiche Patient

antecedents Medicaux
antecedents Chirurgicaux
donnees Terrain



Simulation d'une base de données

grace aux **Arrays**

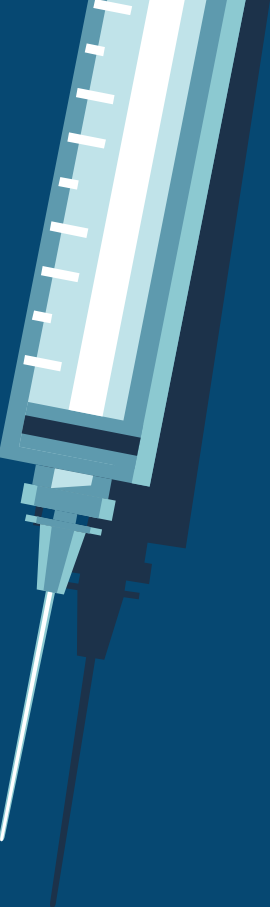
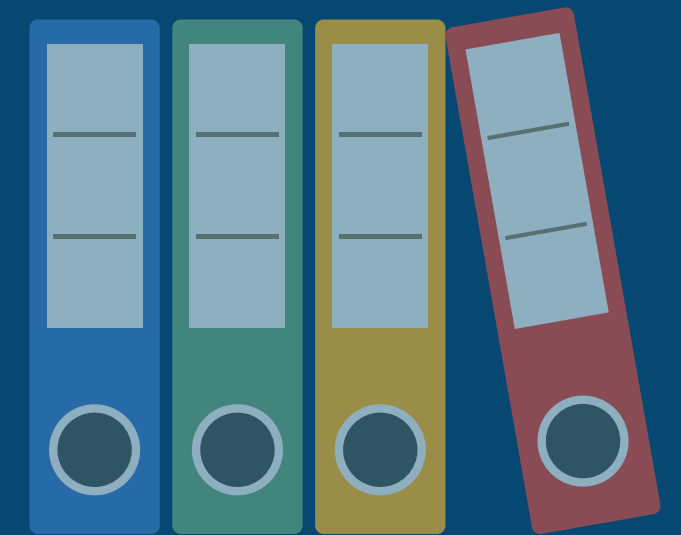
Dans notre cas :

Les **patients** du cabinet seront stocker dans une List de type Patient

L'**ordonnance** aura une List de Medicaments comme attribut

La classe **Dossier Medical** aura une List d'ordonnance + List fiche patient

Les **Dates de rendez vous** et les observations de la **consultation** seront archivé dans une List



Prise des Rendez-vous

cas new
Patient

Vérification de
l'existence du patient
dans la BDD

Gestion
Patient

creation obj

Patient

Ajoute du patient dans la liste

creation obj

Dossier
Medical

creation dossier vierge

Grace aux methodes :
Rendezvousdispo
PriseRendezvous

creation obj

Rendez
Vous

si horaire non dispo alors echec



Add dans la List des rdv maj





Prise des Rendez-vous

Package importé pour cette fonctionnalité:

`java.time.LocalDateTime;`

`java.util.ArrayList;`

`java.time.LocalDate;`

Méthodes implémentées selon la nécessité:

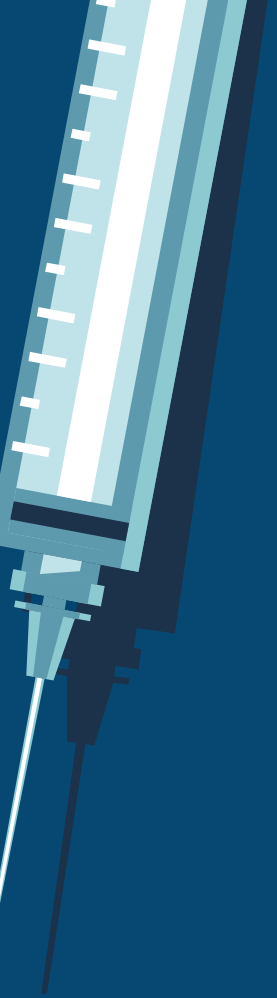
`boolean rendezVousDisponible();`

`priseRendezVous();`

`annulerRendezvous();`

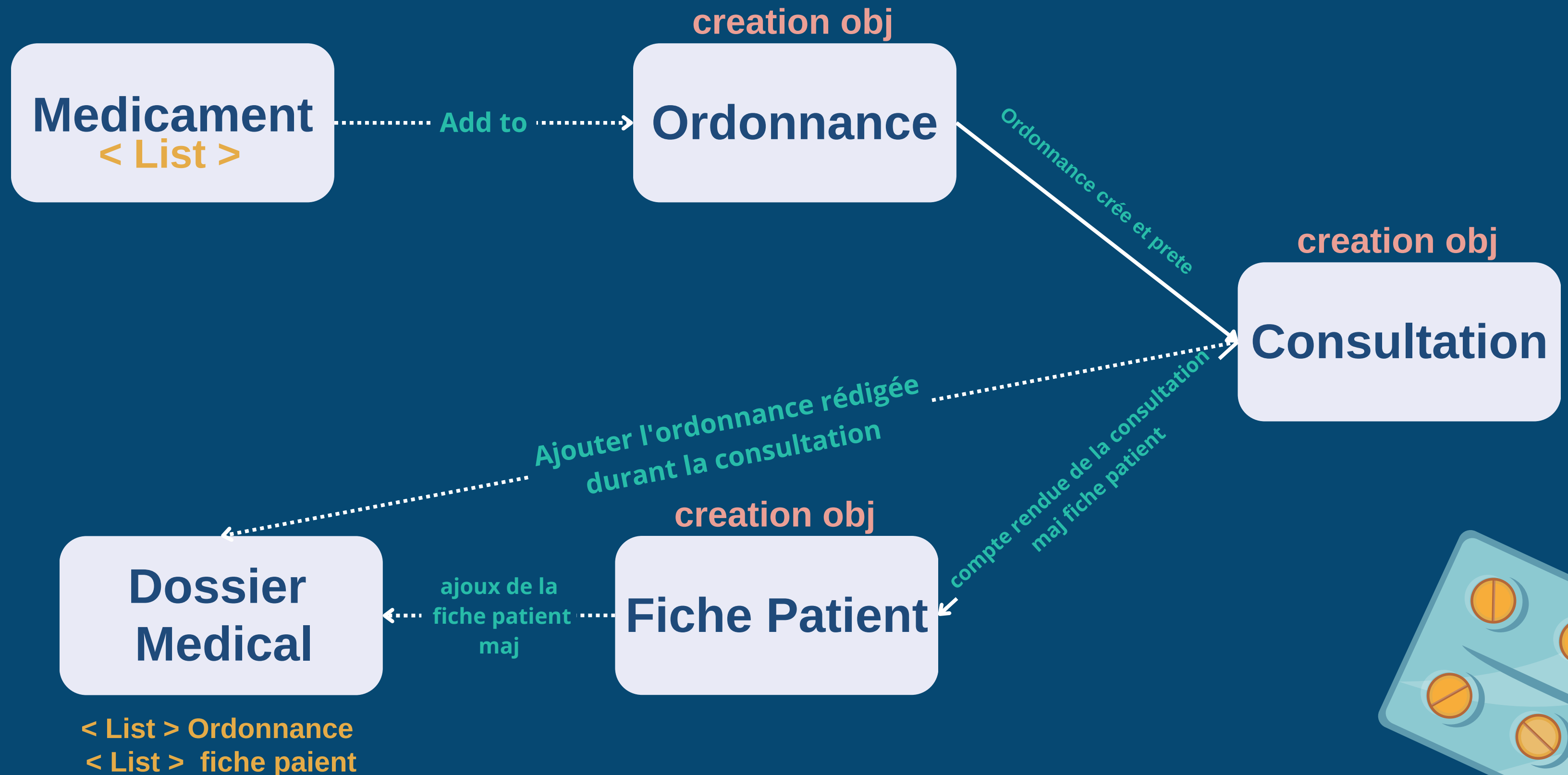
`GestionPatient.ajouterPatient(patient)`

`patient.setDossierMedical(dossiermedical);`



Gestion du Dossier et des fiches patients

Jour de la consultation :





Gestion du Dossier et des fiches patients

methodes jouant un role dans la gestion :

```
consultation.getPatient().getDossierMedical().ajouterOrdonnance(ordonnance);  
consultation.getPatient().getDossierMedical().ajouterFichePatient(fichepatient);
```

```
ajouterFichePatient();  
ajouterOrdonnance();  
ajouterMedicament();  
List getFichesPatient()  
List getOrdonnances()
```



Quelques méthodes utiles au cabinet ont été ajoutées



Rendez Vous	Patient	Gestion Patient	Fiche Patient	Consultation	Ordonnance
getRendezvousliste()	getDossierMedical()	existePatient();	getAntecedentsMedic aux	getObservations()	ajouterMedicament()
getDate()	setDossierMedical()	supprimerPatient();	getAntecedentsChirur gicaux()	effectuerConsultation ()	remplirOrdonnance()
getHeure()	afficherDossierMedic al()	rechercherPatient();	getDonneesTerrain()	getMedecin();	
afficherRendezVousP arPatient()				getOrdonnance()	
				verifierMotDePasse()	

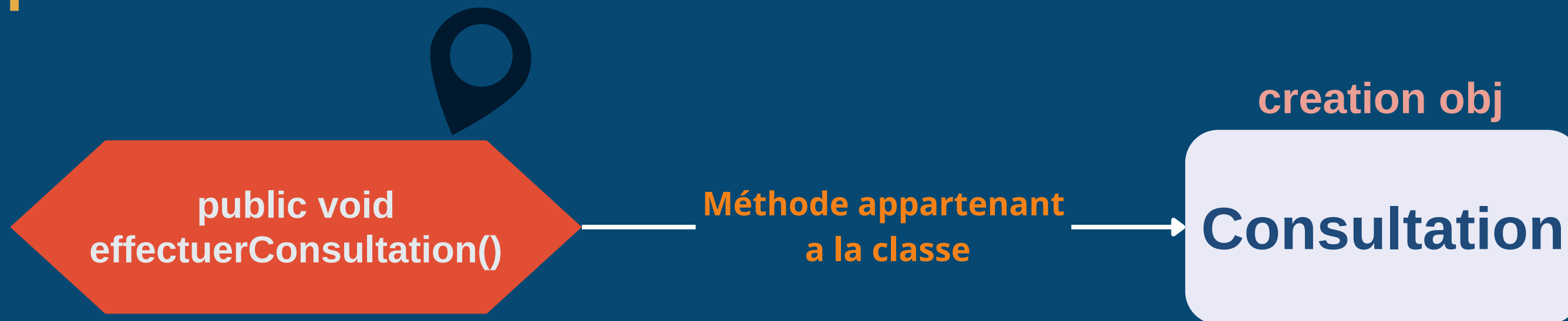


Méthodes **static**



Vu la nécessité d'accéder à certaines classes avant la **création de l'objet**, des méthodes spécifiques ont été déclarées **statiques**

exemple :



1 remplit toutes les informations prises durant la consultation + autres traitements

2 une fois effectuée, remplit l'**ordonnance** + **maj** fiche patient + **maj** dossier Medical





Thanks!



Q&A / feedback

