

# Remarque

- Les types de retour List ne sont pas obligatoires: on peut utiliser Vector ou ArrayList...
- Toujours utiliser des setters dans les constructeurs
- Vérifier la fermeture des connexions en les utilisant et limiter le nombre d'utilisation des connexions

## Classes

---

### Modele

- Genre
  - id
  - nom
- NiveauEtude
  - id
  - nom
- Employe
  - id
  - matricule
  - nom
  - prenom
  - dateNaissance
  - idgenre
  - idniveauEtude
  - dateEmbauche
  - Genre genre
  - NiveauEtude niveauEtude
- Specialite
  - id
  - nom
  - description
- SpecialiteEmploye
  - idemploye
  - idspecialite
  - Employe employe
  - Specialite specialite

### Servlet

- NewEmployeC
- SearchEmployeC
- DetailsEmployeC

•SpecialiteEmployeC

# Fonctions:

---

•Genre

- public List findAll(): prendre toute les données de la table Genre

•NiveauEtude

- public List findAll(): prendre toute les données de la table NiveauEtude

•SpecialiteEmploye

- public void create(): insérer dans la table SpecialiteEmploye

- public List findByEmp(int idemp): prendre toute les données de la table SpecialiteEmploye dont idemploye=idemp

•Employe

- public int getAge(): calculer l'age de l'employé

- private int getAnneActuelle(): retourner l'année de la date actelle

- private int getAnneDate(Date date): retourner l'année de la date correspondante

- public void create(): insérer l'employé dans la base de donnée

- private void setDateNaissance(Date dN): setters pour l'attribut dateNaissance

- public boolean checkAge(): vérifier l'age de la date de naissance

- si getAge()<18 -> throws new exception('employé mineur')

- sinon this.dateNaissance=dN

- public List findAll(): prendre toutes les données depuis la table Employe

- public Employe findById(int idemploye): retourner l'employé dont id=idemploye

- public Employe addSpecialite(Specialite s):

- ajouter la spécialité s dans specialites (List: attribut de la table Employe)

- utiliser SpecialiteEmploye.create()

- public List filter(String pattern): filtrer la listes des employés cirtères->[matricule like '%(pattern)%' || nom like '%(pattern)%' || prenom like '%(pattern)%']

- si pattern==null -> return Employe.findAll()

•Specialite

- public void create(): insérer la spécialité dans la base de donnée

- public List findAll(): prendre toutes les données depuis la table Specialite

- public Specialite findById(int idspecialite): retourner l'employé dont id=idspecialite

- public List findByEmp(int idemp): prendre la liste des spécialités dont idemploye=idemp dans la vue v\_specialite\_employe

- public List findNotInEmp(int idemp): prendre toutes les données de la table Specialite qui n'est pas dans la table SpecialiteEmploye dont l'idemploye=idemp
- NewEmployeC
  - get: redirection vers la page(new\_emp)
    - Genre.findAll() -> genres
    - NiveauEtude.findAll() -> niveau\_etudes
  - post: redirection vers controlleur SearchEmployeC
    - Employe.create()
- SearchEmployeC
  - get: redirection vers la page list\_emp?pattern=
    - Employe.filter(pattern) -> employes
- DetailsEmployeC
  - get: redirection vers la page details\_emp?idemp=
    - Specialite.findNotInEmp(idemp) -> specialites
    - Employe.findById(idemp) -> employe
    - Specialite.findByEmp(int idemp) -> empSpecialites
  - post: redirection vers Controlleur DetailsEmployeC
    - Specialite.create()
- SpecialiteEmployeC
  - post: redirection vers controlleur DetailsEmployeC
    - Employe.addSpecialite(specialite)

# Jan/12/2023

---

## Classes

### Modele

- CategorieService
  - id
  - nom
- Service
  - id
  - idcategorieService
  - CategorieService categorieService
- AffectationService
  - id
  - idemploye
  - idspecialite
  - idservice
  - duree
  - Employe employe
  - Specialite specialite

- Service service
- Salaire
  - id
  - idemploye
  - montant
  - Employe employe

## Servlet

- AffectationServiceC
- SalaireC

## Fonctions

- List CategorieService.findAll()
- List Service.findByCategorie(int idcategorie)
- List Employee.findWithSpecialite(): utiliser view v\_specialite\_employe
- Service.addEmployeeSpecialite(int idemploye,int idspecialite)
  - AffectationService.create(int idemploye,int idspecialite, int duree)
- Salaire.create()