

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
- NiveauEtude
- Employe
- Specialite
- SpecialiteEmploye

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(String 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(String 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(String idspecialite): retourner l'employé dont id=idspecialite
 - public List findByEmp(String idemp): prendre la liste des spécialités dont idemploye=idemp dans la vue v_specialite_employe
 - public List findNotInEmp(String 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(String idemp) -> empSpecialites
 - post: redirection vers Controlleur DetailsEmployeC

- Specialite.create()
- SpecialiteEmployeC
 - post: redirection vers controleur DetailsEmployeC
 - Employe.addSpecialite(specialite)