

Dossier Technique

Description de chaque classe et explication détaillée de chaque méthode

1. AffichagePers.java

Description: Cette classe contient des méthodes pour afficher les informations du personnel.

- **AffichageListe(ArrayList<Personnel> Person):**
 - **Description:** Affiche la liste de tout le personnel actif en format tabulaire.
 - **Paramètres:** ArrayList<Personnel> Person - Liste des objets Personnel.
 - **Retour:** Aucun.
- **AffichageUnePersonne(Personnel pers):**
 - **Description:** Affiche les informations détaillées d'une seule personne.
 - **Paramètres:** Personnel pers - Objet Personnel.
 - **Retour:** Aucun.

2. ClassMain.java

Description: Classe principale de l'application qui initialise et lance l'interface graphique.

- **main(String[] args):**
 - **Description:** Méthode principale de l'application qui applique le look and feel Nimbus à l'interface utilisateur et rend visible la fenêtre principale de l'application GestionJFrame.
 - **Paramètres:** String[] args - Arguments de la ligne de commande.
 - **Retour:** Aucun.

3. ControleSaisie.java

Description: Cette classe contient des méthodes pour valider les saisies utilisateur.

- **valideNom(String text, int taille):**

- **Description:** Valide un nom en fonction des contraintes de taille et du pattern défini.
- **Paramètres:** String text, int taille.
- **Retour:** boolean - Vrai si le nom est valide, faux sinon.
- **valideNomFichier(String file):**
 - **Description:** Valide le nom d'un fichier en tentant de le créer.
 - **Paramètres:** String file.
 - **Retour:** boolean - Vrai si le nom du fichier est valide, faux sinon.
- **valideEmail(String email):**
 - **Description:** Valide un email en utilisant le pattern défini.
 - **Paramètres:** String email.
 - **Retour:** boolean - Vrai si l'email est valide, faux sinon.
- **oui_non(String result):**
 - **Description:** Vérifie si l'entrée est "oui" ou "non".
 - **Paramètres:** String result.
 - **Retour:** boolean - Vrai si l'entrée est valide, faux sinon.
- **un_deux(String result):**
 - **Description:** Vérifie si l'entrée est "1" ou "2".
 - **Paramètres:** String result.
 - **Retour:** boolean - Vrai si l'entrée est valide, faux sinon.
- **valideDate(String date):**
 - **Description:** Valide une date au format "dd/MM/yyyy".
 - **Paramètres:** String date.
 - **Retour:** boolean - Vrai si la date est valide, faux sinon.
- **valideDept(String dept):**
 - **Description:** Valide un département (1 à 4).
 - **Paramètres:** String dept.
 - **Retour:** boolean - Vrai si le département est valide, faux sinon.

4. *Departement.java*

Description: Énumération représentant les différents départements d'une entreprise.

- **Compta("Comptabilité"), HR("Ressources Humaines"), Prod("Production"), SEC("Sécurité"):**
 - **Description:** Valeurs de l'énumération représentant les différents départements.
- **getDeptNom():**

- **Description:** Obtient le nom du département.
- **Paramètres:** Aucun.
- **Retour:** String - Le nom du département.

5. Emprunt.java

Description: Classe représentant un emprunt de matériel par un personnel.

- **Emprunt(Personnel emprunteur, Produit materiel, int nombre):**
 - **Description:** Constructeur pour initialiser l'objet Emprunt.
 - **Paramètres:** Personnel emprunteur, Produit materiel, int nombre.
 - **Retour:** Aucun.
- **getEmprunteur():**
 - **Description:** Getter pour emprunteur.
 - **Paramètres:** Aucun.
 - **Retour:** Personnel.
- **setEmprunteur(Personnel emprunteur):**
 - **Description:** Setter pour emprunteur.
 - **Paramètres:** Personnel emprunteur.
 - **Retour:** Aucun.
- **toString():**
 - **Description:** Surcharge de la méthode toString pour fournir une représentation en chaîne de caractères de l'objet Emprunt.
 - **Paramètres:** Aucun.
 - **Retour:** String.
- **getMateriel():**
 - **Description:** Getter pour materiel.
 - **Paramètres:** Aucun.
 - **Retour:** Produit.
- **getArticle():**
 - **Description:** Méthode pour obtenir le nom et la description du produit.
 - **Paramètres:** Aucun.
 - **Retour:** String.
- **getNombre():**
 - **Description:** Getter pour nombre.
 - **Paramètres:** Aucun.
 - **Retour:** int.

6. FixedLenght.java

Description: Cette classe contient une méthode pour fixer la longueur d'une chaîne de caractères.

- **setFixedLength(String source, int longueur):**
 - **Description:** Méthode pour fixer la longueur d'une chaîne de caractères en ajoutant des espaces.
 - **Paramètres:** String source, int longueur.
 - **Retour:** String.

7. GestionEmprunt.java

Description: Classe pour gérer les emprunts et retours de matériel.

- **Magasin(Magasin mag):**
 - **Description:** Méthode pour ajouter des produits au magasin et afficher la liste des produits.
 - **Paramètres:** Magasin mag.
 - **Retour:** Aucun.
- **Emprunt(Magasin mag, ArrayList<Personnel> Person, ArrayList<Emprunt> pret):**
 - **Description:** Méthode pour gérer les emprunts de matériel.
 - **Paramètres:** Magasin mag, ArrayList<Personnel> Person, ArrayList<Emprunt> pret.
 - **Retour:** Aucun.
- **RetourEmprunt(ArrayList<Emprunt> pret):**
 - **Description:** Méthode pour gérer le retour des emprunts de matériel.
 - **Paramètres:** ArrayList<Emprunt> pret.
 - **Retour:** Aucun.

8. GestionJFrame.java

Description: Classe principale de l'interface graphique de l'application.

- **GestionJFrame():**
 - **Description:** Constructeur pour initialiser et configurer la fenêtre principale avec ses composants et actions associées.
 - **Paramètres:** Aucun.
 - **Retour:** Aucun.

9. GestionPersonnel.java

Description: Classe pour gérer les opérations liées au personnel.

- **LoadPersonnel(ArrayList<Personnel> Person):**
 - **Description:** Charge les données du personnel depuis un fichier XML.
 - **Paramètres:** ArrayList<Personnel> Person.
 - **Retour:** Aucun.
- **ModifPersonnel(ArrayList<Personnel> Person):**
 - **Description:** Modifie les données d'un personnel.
 - **Paramètres:** ArrayList<Personnel> Person.
 - **Retour:** Aucun.
- **SuppressionPersonnel(ArrayList<Personnel> Person):**
 - **Description:** Supprime un personnel.
 - **Paramètres:** ArrayList<Personnel> Person.
 - **Retour:** Aucun.
- **AjoutPersonnel(ArrayList<Personnel> person):**
 - **Description:** Ajoute un nouveau personnel.
 - **Paramètres:** ArrayList<Personnel> person.
 - **Retour:** Aucun.

10. Individu.java

Description: Classe de base représentant un individu avec des attributs communs.

- **Individu(String nom, String prenom, Sexe sexe, MyDate dateNaissance):**
 - **Description:** Constructeur pour initialiser un objet Individu.
 - **Paramètres:** String nom, String prenom, Sexe sexe, MyDate dateNaissance.
 - **Retour:** Aucun.
- **getNom():**
 - **Description:** Getter pour l'attribut nom.
 - **Paramètres:** Aucun.
 - **Retour:** String.
- **setNom(String nom):**
 - **Description:** Setter pour l'attribut nom.
 - **Paramètres:** String nom.
 - **Retour:** Aucun.

- **getPrenom():**
 - **Description:** Getter pour l'attribut prenom.
 - **Paramètres:** Aucun.
 - **Retour:** String.
- **setPrenom(String prenom):**
 - **Description:** Setter pour l'attribut prenom.
 - **Paramètres:** String prenom.
 - **Retour:** Aucun.
- **getSexe():**
 - **Description:** Getter pour l'attribut sexe.
 - **Paramètres:** Aucun.
 - **Retour:** String.
- **setSexe(Sexe sexe):**
 - **Description:** Setter pour l'attribut sexe.
 - **Paramètres:** Sexe sexe.
 - **Retour:** Aucun.
- **getDateNaissance():**
 - **Description:** Getter pour l'attribut dateNaissance.
 - **Paramètres:** Aucun.
 - **Retour:** MyDate.
- **setDateNaissance(MyDate dateNaissance):**
 - **Description:** Setter pour l'attribut dateNaissance.
 - **Paramètres:** MyDate dateNaissance.
 - **Retour:** Aucun.
- **getDateddMMyyyy():**
 - **Description:** Méthode pour obtenir la date de naissance au format jj-mm-aaaa.
 - **Paramètres:** Aucun.
 - **Retour:** String.
- **toString():**
 - **Description:** Surcharge de la méthode toString pour fournir une représentation en chaîne de caractères de l'objet Individu.
 - **Paramètres:** Aucun.
 - **Retour:** String.

11. InputData.java

Description: Classe pour gérer les saisies utilisateur avec validation.

- **inputNomPrenom(String libelle, int tailleNom):**

- **Description:** Saisit un nom ou un prénom avec validation.
- **Paramètres:** String libelle, int tailleNom.
- **Retour:** String.
- **inputEmail():**
 - **Description:** Saisit un email avec validation.
 - **Paramètres:** Aucun.
 - **Retour:** String.
- **inputNomFichier():**
 - **Description:** Saisit un nom de fichier avec validation.
 - **Paramètres:** Aucun.
 - **Retour:** String.
- **suppression():**
 - **Description:** Confirme une suppression.
 - **Paramètres:** Aucun.
 - **Retour:** boolean.
- **inputSexe():**
 - **Description:** Saisit le sexe avec validation.
 - **Paramètres:** Aucun.
 - **Retour:** String.
- **InputDepartement():**
 - **Description:** Saisit le département avec validation.
 - **Paramètres:** Aucun.
 - **Retour:** String.
- **inputDateNaissance():**
 - **Description:** Saisit une date de naissance avec validation.
 - **Paramètres:** Aucun.
 - **Retour:** String.

12. Lire.java

Description: Classe pour gérer la saisie utilisateur via la console.

- **Lire():**
 - **Description:** Constructeur pour initialiser le Scanner.
 - **Paramètres:** Aucun.
 - **Retour:** Aucun.
- **SaiseTexte(String libelle):**
 - **Description:** Saisit du texte avec un libellé donné.
 - **Paramètres:** String libelle.
 - **Retour:** String.

- **saisieInt():**
 - **Description:** Saisit un entier avec validation.
 - **Paramètres:** Aucun.
 - **Retour:** int.

13. Magasin.java

Description: Classe représentant un magasin de produits.

- **Magasin():**
 - **Description:** Constructeur pour initialiser la map des produits.
 - **Paramètres:** Aucun.
 - **Retour:** Aucun.
- **AjouterProduit():**
 - **Description:** Ajoute des produits à la map.
 - **Paramètres:** Aucun.
 - **Retour:** Aucun.
- **listeMap():**
 - **Description:** Affiche la liste des produits.
 - **Paramètres:** Aucun.
 - **Retour:** Aucun.
- **getProduit(String nom):**
 - **Description:** Obtient un produit par son nom (clé).
 - **Paramètres:** String nom.
 - **Retour:** Produit.
- **isEmpty():**
 - **Description:** Vérifie si la map de produits est vide.
 - **Paramètres:** Aucun.
 - **Retour:** boolean.

14. MyDate.java

Description: Classe représentant une date avec jour, mois et année.

- **MyDate(int jour, int mois, int annee):**
 - **Description:** Constructeur pour initialiser une date.
 - **Paramètres:** int jour, int mois, int annee.
 - **Retour:** Aucun.
- **getJour():**

- **Description:** Getter pour le jour.
- **Paramètres:** Aucun.
- **Retour:** int.
- **getMois():**
 - **Description:** Getter pour le mois.
 - **Paramètres:** Aucun.
 - **Retour:** int.
- **getAnnee():**
 - **Description:** Getter pour l'année.
 - **Paramètres:** Aucun.
 - **Retour:** int.

15. *Personnel.java*

Description: Classe représentant un membre du personnel.

- **Personnel(String nom, String prenom, Sexe sexe, MyDate dateNaissance, String email, Departement dept):**
 - **Description:** Constructeur pour initialiser un objet Personnel.
 - **Paramètres:** String nom, String prenom, Sexe sexe, MyDate dateNaissance, String email, Departement dept.
 - **Retour:** Aucun.
- **toString():**
 - **Description:** Surcharge de la méthode toString pour fournir une représentation en chaîne de caractères de l'objet Personnel.
 - **Paramètres:** Aucun.
 - **Retour:** String.
- **getIdPersonnel():**
 - **Description:** Getter pour l'identifiant du personnel.
 - **Paramètres:** Aucun.
 - **Retour:** int.
- **getDepartement():**
 - **Description:** Getter pour le nom du département.
 - **Paramètres:** Aucun.
 - **Retour:** String.
- **getEmail():**
 - **Description:** Getter pour l'email.
 - **Paramètres:** Aucun.
 - **Retour:** String.
- **setEmail(String email):**

- **Description:** Setter pour l'email.
- **Paramètres:** String email.
- **Retour:** Aucun.
- **isActif():**
 - **Description:** Getter pour le statut actif.
 - **Paramètres:** Aucun.
 - **Retour:** boolean.
- **setActif(boolean actif):**
 - **Description:** Setter pour le statut actif.
 - **Paramètres:** boolean actif.
 - **Retour:** Aucun.

16. *PersonnelXml.java*

Description: Classe pour gérer la lecture des données du personnel à partir d'un fichier XML.

- **Lecture():**
 - **Description:** Lit les données du personnel depuis un fichier XML et retourne une liste de nœuds.
 - **Paramètres:** Aucun.
 - **Retour:** NodeList.

17. *Produit.java*

Description: Classe représentant un produit.

- **Produit(String nom, String description):**
 - **Description:** Constructeur pour initialiser un produit.
 - **Paramètres:** String nom, String description.
 - **Retour:** Aucun.
- **getNom():**
 - **Description:** Getter pour le nom du produit.
 - **Paramètres:** Aucun.
 - **Retour:** String.
- **setNom(String nom):**
 - **Description:** Setter pour le nom du produit.
 - **Paramètres:** String nom.
 - **Retour:** Aucun.
- **getDescription():**

- **Description:** Getter pour la description du produit.
- **Paramètres:** Aucun.
- **Retour:** String.
- **setDescription(String description):**
 - **Description:** Setter pour la description du produit.
 - **Paramètres:** String description.
 - **Retour:** Aucun.
- **toString():**
 - **Description:** Surcharge de la méthode toString pour fournir une représentation en chaîne de caractères de l'objet Produit.
 - **Paramètres:** Aucun.
 - **Retour:** String.

18. Sauvegarde.java

Description: Classe pour gérer la sauvegarde des données du personnel et des emprunts dans un fichier texte.

- **Sauvegarde(ArrayList<Personnel> Person, ArrayList<Emprunt> pret):**
 - **Description:** Sauvegarde les données du personnel et des emprunts dans un fichier texte.
 - **Paramètres:** ArrayList<Personnel> Person, ArrayList<Emprunt> pret.
 - **Retour:** Aucun.

19. Sexe.java

Description: Énumération représentant le sexe d'un individu.

- **HOMME("Garçon"), FEMME("Fille"):**
 - **Description:** Valeurs de l'énumération représentant les sexes possibles.
- **getLabel():**
 - **Description:** Obtient le label du sexe.
 - **Paramètres:** Aucun.
 - **Retour:** String.

20. WordUtils.java

Description: Classe utilitaire pour manipuler les chaînes de caractères (capitalisation et minuscule).

- **mettreEnMajscule(String inputString):**
 - **Description:** Met en majuscule chaque partie d'une chaîne de caractères.
 - **Paramètres:** String inputString.
 - **Retour:** String.
- **mettreEnMinusculeMail(String inputString):**
 - **Description:** Met en minuscule chaque partie d'une adresse e-mail.
 - **Paramètres:** String inputString.
 - **Retour:** String.