RAPPORT PROJET TP POO
ZAIT FOUAD

Description des classes et methodes :

CLASSE AdrMail :

Attribut : String pseudo ,String site ,String mot_de_passe

constructeur : AdrMail

Getters et setters de chaque attribut

Public boolean pseudovalide() : verifie la validité du pseudo selon les conditions données si il est valide renvoie true sinon false .

Public boolean mdpvalide () : verifie la validité du mot de passe selon les conditions données si il est valide renvoie true sinon false .

Public String toString () : redéfinition de la methode to String elle retourne l'adresse mail et le mot de passe

Public String affiche_adr() : retourne l'adresse mail (pseudo@site) uniquement

Public boolean equals (): redéfinition de la méthode equals elle verifie l'égalité entre l'adresse mail courante et un objet donné .

Public void saisir() : permet de saisir une adresse mail (pseudo + site) et mot de passe .

Public void saisir_pseudo_site () : permet de saisir le pseudo et site . On en aura besoin pour la saisie d'une adresse d'un destinataire.

Public void modif_mot_de_passe (): permet de modifier le mot de passe en saisissant l'ancien .

Public void modif_pseudo () : permet de modifier le pseudo .

Classe AdrMailProf : extends AdrMail

Attributs en plus : String nom_entreprise , String domaine_activite

Constructeur :AdrMailProf

Getters et setters

Public String toString : redéfinition de la methode toString elle retourne l'adresse mail , le mot de passe, le nom de l'entreprise et le secteur d'activité.

Public void saisir : appelle la methode saisir()de la super classe AdrMail et permet la saisie du nom de l'entreprise et le secteur d'activité.

Public boolean equals : redéfinition de la méthode equals elle verifie l'égalité entre l'adresse mail Professionnelle courante et un objet donné .

Classe Profil :

Attribut : AdrMail adr,String nom,String prenom ,String genre, int age ,long numero_tel,String pays_de_residence.

Constructeur :Profil

Getters et setters

Public String toString : retourne un string contenant le nom , prenom , l'adresse mail , genre , age , num de téléphone et pays de résidence.

Public void saisir : permet de saisir les attribus de la classe profil .

Public boolean equals :verifie l'égalité entre le profil courant et un objet donné en paramètre.

Classe Message :

Attributs : String titre, String contenu , Date date_de_creation, état etat_msg , long taille_msg .

état est une énumération { crée, envoyé,reçu ….etc} .

on a defini avec l'état une methode reculu :on en aura besoin pour la classe Boite Mail pour obliger le message à etre non lu jusqu'à la lecture du message .

Constructeur :Message

Getters et setters

Public String affichedate : retourne l'année le mois et le jour de la creation du message . On en aura besoin pour la methode to String .

Public String toString : retourne un String des attributs concaténés et séparés par un saut de ligne

Public boolean equals : verifie l'égalité entre le Message courant et un objet donné en paramètre.

Public String creedatesys () :creation du format de la date de creation

Public void saisir () : permet de saisir les attributs de la classe Message (la date de creation recevra la date et lheure de la saisie (creation du message)(date system).

Public void modifcontenu :permet de modifier le contenu du message .

Public void afficheTitre : permet d'afficher le titre de message . On en aura besoin dans le menu .

Classe MessageAttach : extends Message

Attributs : pièce jointe p .

Constructeur : MessageAttach

Getters et setters

Public void saisir () : permet la saisie dun message super.saisir() suivi d'une saisie d'une pièce jointe (nom+taille)

Public String toString : retourne un String dun message super.saisir() suivi d'une pièce jointe (nom+taille)

Public boolean equals : verifie l'égalité entre le MessageAttach courant et un objet donné en paramètre.

Public long taille_msg : retourne la taille du message (contenu) super.getTaillemsg() + la taille de la pièce jointe p.getTaille () on en aura besoin dans la classe BoiteMail pour le stockage des messages .

Classe Piècejointe :

Attributs : String nom, long taille

Constructeur : PièceJointe

Getters et setters

Classe creation_msg :

Public static Message creation : permet de créer le message souhaiter par l'utilisateur avec ou sans attachement et le renvoie en respectant les exceptions données.

Classe BoiteMail :

Attributs : AdrMail ad , Arraylist de messages reçu , Arraylist de messages envoyés , Arraylist de messages brouillons, Arraylist de messages archives, Arraylist de messages corbeille , Arraylist de messages spam , final long Stockage Max on l'a supposé 2000MO ,et deux arraylist supplémentaire de messages lu et non lu on en aura besoin dans la méthode recevoir .

Constructeur :BoiteMail

Getters et setters

Public void envoi_msg : permet d'envoyer un message à un ou plusieurs destinataires en respectant les exception données et le stocké dans la arraylist de messages envoyés.

Public void recevoir : permet de recevoir un message il est marqué comme non lu (stocké dans la arraylist de messages non lu) jusqu'à son ouverture. et le stocké dans la arraylist de messages reçus.

Public void affiche_msg :permet d'afficher un message donné.

Public void supprim_msg : permet de supprimer un message si il se trouve dans les messages reçus ou envoyés.

Public void supprim_corbeille : permet de supprimer un message de la corbeille

Public void deplacer_spam_recus : permet de déplacer un message des messages spam vers les messages reçus.

Public void archiver_msg : permet d'ajouter un message dans les messages archivés.

Public void cree_msg : permet de créer un message en utilisant la classe creation_msg , le stocké dans brouillons avant de l'envoyer dès qu'il sera envoyé il sera supprimé des messages brouillons.

Public int nb_recu_lu :retourne le nombre de messages reçus lus

Public int recu_non_lu :retourne le nombre de messages reçus non lus

Public int nb_msg_envoyé : retourne le nombre de messages envoyés.

Public int nb_archives : retourne le nombre de messages dans le dossier archives.

Public int nb_corbeille : retourne le nombre de messages dans le dossier corbeille.

Public int nb_spam : retourne le nombre de messages dans le dossier spam .

Public long Stockage_array : retourne le stockage d'une arraylist si elle contient un message sans attachement on ajoute la taille du message sinon on ajoute la taille du message plus la taille de la pièce jointe . on en aura besoin pour la methode espace_stockage pour calculer l'espace de tous les dossiers de messages (reçus envoyés….. etc)

Public long espace_stockage :retourne le stockage de la boite mail (tous les dossiers) si le nombre atteint 80% du stockage max un message sera afficher .

Public long stockage_restant : retourne le stockage restant de la boite mail

Public void affiche : affiche la boite mail (ladresse +tous les dossiers de messages) on en aura besoin dans le menu .

___AppMessagerie____

Public static int menu() : affiche le menu on ajoutera une boucle while au main pour qu'il s'affiche de manière itérative .

Public static CréerAdresses : on donne en paramètre deux arraylist qui seront une arraylist d'adresses et profils à chaque fois qu'on crée une adresse mail on lajoutera à l'arraylist d'adresses de même pour les profils (à conditions qu'elle ne soit pas une adresse mail Professionnelle car un profil est associé seulement à une adresse mail simple .

Public static void affiche_adresse_categorie2 :on donne en paramètre une arraylist qui sera une arraylist d'adresse mail au main , on demandera à l'utilisateur si il veut afficher les adresses mail Professionnelle ou non avec un switch et les afficher selon la demande en parcourant la arraylis ou seront stockées les adresses.

Public static void affiche_adresse_categorie1 :on donne en paramètre une arraylist qui sera une arraylist d'adresse mail au main , on demandera à l'utilisateur de donner le site dont il veut afficher les adresses mail et les afficher en parcourant la arraylist ou seront stockées les adresses.

Public static AdrMail ajouter_adr_mail : on donne en paramètre une arraylist qui sera une arraylist d'adresse mail au main et on ajoutera une adresse mail saisie selon ce que veut l'utilisateur (Professionnelle ou non) et l'ajouter dans la array list .

Public static void supprimer_adr_mail : on donne en paramètre une arraylist qui sera une arraylist d'adresse mail au main et on supprimera une adresse mail saisie par l'utilisateur et la supprimer de la array list tant que l'adresse saisie existe .

Public static void modifier_adr_mail : on donne en paramètre une arraylist qui sera une arraylist d'adresse mail au main et on modifiera une adresse mail saisie et la modifier en parcouant la arraylist si l'adresse saisie existe. si c est une adresse mail Professionnelle on peut modifier le pseudo et le mot de passe sinon on peut modifier pseudo mot de passe et profil .(avec un switch demandant à l'utilisateur ce qu'il veut).

Public static void creation_boite_mail : on donne en paramètre deux arraylist qui seront au main une d'adresses mail et une de boite mail . On parcours la arraylist d'adresses mail et on crée les boites mail correspondant et les ajouter à la arraylist de boites mails .

Public static void ajout_boite : on donne en paramètre deux arraylist qui serot au main une d'adresses mail et une de boite mail pour ajouter un boute mail il faut d'abord ajouter une adresse mail à la arraylist d'adresses mail et créer sa boite mail et lajouter à la arraylist de boitemail .

Public Static void affichage_boite_mail : on donne en paramètre une araylist qui sera une arraylist de boites mail au main . On parcours cette arraylist et on affiche toutes les boites mail à partir de la méthode affiche d'une boite mail.

Public static int sous_menu() : affiche le sous menu et permet la saisie du numero du choix si l'on choisit au main le numero 7 qui est la gestion de boite mail .

Public static void ajout_auto_msg : on donne en paramètre une arraylist qui sera une arraylist de boite mail au main . Cette methode permet de parcourir toutes les boites mail et de créer des messages (reçus,envoyés,brouillons) selon la demande de l'utilisateur.

Public static void affiche_boite : on donne en paramètre une boite mail qu' on souhaite afficher . Cette methode affichera la boite mail avec tous les dossiers et l'espace utilisé et l'espace restant.

Public static void envoi_message : on donne en paramètre unee arraylist qui sera une array list de boite mail au main et une array list qui sera une arraylist d'adresses pour rechercher l'adresse du destinataire (on rajoutera 3arraylist LDEST ou on seront stockées les adresses des destinataires LEXP ou on seront stockées les adresses des expediteurs et M ou seront stockées les messages envoyés tous ces 3 seront nécessaires pour efectuer la requête 6).On crée le message qu'on souhaite envoyer, on saisie ladresse du destinataire et si elle ne se trouve pas dans la arraylist une exception sera générée sinon l'envoie sera effectuée à l'adresse saisie.

Public static affiche_msg : on donne en paramètre une araylist qui sera une arraylist de boites mail au main . Cette methode permet d'afficher un message donné si il se trouve dans la boite mail donnée .

Public static supp_msg : on donne en paramètre une araylist qui sera une arraylist de boites mail au main . Cette methode permet de supprimer un message donné si il se trouve dans la boite mail donnée.

Public static void archiver_msg : on donne en paramètre une arraylist de boite mail au main .cette methode permet de stocker un message dans le dossier archives de la boite mail donnée.

Public static void archiver_msg_date : on donne en paramètre une arraylist qui sera une arraylist boite mail . Cette methode permettra d'archiver les messages reçus avant une date d donnée d'une boite mail donnée.

Public static void restaurer_msg : on donne en paramètre une arraylist . Cette methode permettra de restaurer un message donné restituer de la corbeille d'une boite mail donnée.

Public static void repondre_msg : on donne en paramètre une arraylist qui sera une arraylist de boites mail au main et une array list qui sera une arraylist d'adresses pour rechercher l'adresse de celui à qui on veux répondre (on rajoutera 3arraylist LDEST ou on seront stockées les adresses des destinataires LEXP ou on seront stockées les adresses des expediteurs et M ou seront stockées les messages envoyés tous ces 3 seront nécessaires pour efectuer la requête 6).On crée le message qu'on souhaite envoyer, on saisie ladresse du destinataire et si elle ne se trouve pas dans la arraylist une exception sera

générée sinon l'envoie sera effectuée à l'adresse saisie.Public void vider_spam : on donne en paramètre une boite mail . Cette methode permet de vider le dossier spam de la boite .

Public void vider_envoyés : on donne en paramètre une arraylist qui sera une arraylist de boite mail au main . Cette methode permet de vider le dossier envoyé d'une boite donnée.

Public static void trier_msg_par_date : on donne en paramètre une arraylist qui sera une arraylist de boite mail cette méthode permet de trier les messages par date d'une boite donnée on utiliste le tri à bulle .

Public static void trier_msg_par_objet : on donne en paramètre une arraylist qui sera une arraylist de boites mail au main cette méthode permet de trier les messages d'une boites données par objets ordre alphabétique.on utiliste la méthode de la superclass array Collections.sort .

Public int menu_autres_requetes : cette méthode permet d'afficher le menu des autres requêtes et retournera le numero de la requête qu'on souhaite retourner

Public static void requete1 : on donne en paramètre une arraylist qui sera une arraylist de boite mail . Cette methode permettra d'afficher toute les boites ayant reçu un message donné .

Public static void requete2 : on donne en paramètre une arraylist qui sera une arraylist de boite mail . Cette methode permettra d'afficher les boites qui sont remplies à plus de 50% de leur capacité.

Public static void requete3 : on donne en paramètre une arraylist qui sera une arraylist de boite mail . Cette methode permettra d'eclater la array de boite mail enbdeux arraylist une contiendra les boites publiques et l'autre contiendra les boites professionnelles.

Public static void requete4 : on donne en paramètre une arraylist qui sera une arraylist de boite mail . Cette methode permettra de retourner le pourentage d'utilisation par la catégorie entre 18 et 35ans et afficher le résultat à la fin .

Public static void recherche_dossier_affiche : on donne en paramètre une arraylist et on la parcourt. Si l'objet à l'indice i est un MessagAttach (message avec pièce jointe) on laffiche . On en aura besoin pour la requete 5 .

Public static void requete5 : cette methode permet d’afficher tous les messages ayant des pièces jointes pour une boite donnée on utilise la methode recherche_dossier_affiche pour afficher tous les dossiers de la boites ayant des pièces jointes .

On fera un switch après avoir choisi la requête 6 . On demandera à l'utilisateur la manière dont il veut afficher les messages .

Public static void requete6_recherche_destinataire : on donne en paramètre une arraylist LDEST ou seront stockées les adresses des destinataires et une arraylist ou seront stockées les messages correspondant . On demande à l'utilisateur de donner ladresse du destinataire on la cherche et on affiche le message correspondant stocké dans M.

Public static void requete6_recherche_expediteur : on donne en paramètre une arraylist LDEST ou seront stockées les adresses des destinataires et une arraylist ou seront stockées les messages correspondant . On demande à l'utilisateur de donner ladresse du destinataire on la cherche et on affiche le message correspondant stocké dans M.

Public static void requete6_affiche_mot_clé : pour une boite donnée vérifier dans tous les dossiers de la boite si le mot clé donnée est contenu dans le titre ou dans le contenu des messages . Si c est le cas afficher ce message .

Public static void requete7 : on donne en paramètre deux arraylist qui seront au main une de boites mail et une de profil . On commence par parcourir la arraylist de boites mail et compter le nombre de répétition de la boite mail n° i

Si le nombre est Supérieur ou égale à 2 on affiche le profil de cette boite .

Public static void requete8 : on donne en paramètre une arraylist qui sera une boite mail au main . Cette méthode permettra de vider toutes les boites d'un site donner . On parcours la arraylist et si le site de l'adresse de la boîte est egale au site donné on la supprime .