

## Projet de TP POO - Programmation en java

**NB : le travail doit être fait par binôme (ou monôme)**

**Le travail doit être envoyé (rapport + captures d'exécution+ code source) à l'adresse**  
**[tracking.POOL2@gmail.com](mailto:tracking.POOL2@gmail.com)**

**Au plus tard le mardi 30/06/2020 en précisant « ProjetTP-POO – noms ».**

**Les programmes écrits doivent être lisibles et bien commentés.**

**Les cas de plagia seront sanctionnés**

Dans ce TP, on voudrait simuler la gestion de boîtes de messagerie électronique (boîtes e-mail).

Une boîte e-mail est identifiée par une adresse électronique (exemple [Section2aAcad@gmail.com](mailto:Section2aAcad@gmail.com))

Une boîte e-mail comporte un ensemble de messages organisés dans différents dossiers (messages reçus, messages envoyés, brouillons, corbeille, archives, ...etc)

### **Description des classes**

Une **adresse mail** (AdrMail) est définie par un nom d'utilisateur (pseudo) suivi du nom du site (par exemple gmail.com, usthb.dz, yahoo.fr), un mot de passe et est associée à un profil.

Le *nom d'utilisateur* doit commencer par une lettre, il peut comporter des chiffres, le caractère tiret (- ou \_) ou point (.).

Le *mot de passe* doit être composé de huit (08) caractères au moins, il doit comporter des lettres au moins un chiffre et au moins un caractère spécial. On peut modifier le mot de passe, en saisissant l'ancien mot de passe d'abord. Toute modification doit être confirmée d'abord (saisir une deuxième fois le mot de passe).

Un **profil** (Profil) est décrit par un nom, un prénom, un genre (Feminin, Masculin) un âge, un numéro de téléphone et le pays de résidence.

Parmi les adresses, on peut avoir des adresses professionnelles (liées à une entreprise, université, ...)

Une **adresse professionnelle** (AdrMailProf) comporte en plus, le nom de l'entreprise, son domaine d'activité (recherche, production, éducation, santé, enseignement supérieur, ...).

Un **message** (Message) est décrit par un objet (String, le titre du message), un contenu (le texte du message), une date et heure de création et un état (« créé » initialement, « envoyé », « reçu », « en cours », « stocké » dans le dossier « brouillon », « supprimé », « archivé », « retourné » en cas d'un échec d'envoi) et une taille (en nombre d'octets).

Parmi les messages, on peut avoir des **messages avec attachement** (pièces jointes) (MessageAttach). On ajoute alors l'ensemble des pièces jointes (sans répétition bien sur).

Une **pièce jointe** (PièceJointe) est un fichier qui est identifié par un nom, et possède une taille (en octets).

On suppose qu'on ne peut pas attacher plus de 10 Mo (méga-octets) de pièces à un même message.

Une exception doit être générée si :

- L'attachement dépasse 10Mo ou
- on essaie d'attacher deux fois la même pièce.

On peut :

- créer un message (avec ou sans attachement),
- afficher un message,
- modifier son contenu (ou objet) avant de l'envoyer, ...etc.

Une **boîte de messagerie** (boîte e-mail) (BoiteMail) est rattachée à une adresse mail, elle comporte plusieurs dossiers de messages (reçus, envoyés, brouillons, archives, corbeille et spam) et une capacité de stockage maximale (en Mo).

Le nombre de messages dans chaque dossier peut augmenter ou diminuer de manière dynamique (pas de limite pour le nombre de messages).

A partir d'une boîte de messagerie, on peut :

- envoyer un message à un destinataire (ou plusieurs), il faudra spécifier l'adresse (ou les adresses) du (des) destinataire(s)). Il faudra l'ajouter dans le dossier « envoyés ».

Une exception sera générée si une adresse de destinataire est incorrecte ou le message est vide.

- recevoir un message, il faudra l'ajouter dans le dossier « reçus », il est marqué comme « non lu » jusqu'à son ouverture (afficher son contenu).
- ouvrir un message (l'afficher)
- supprimer un message (soit du dossier « envoyés » ou « reçus »), il faudra le mettre dans le dossier « corbeille ».
- supprimer un message de la corbeille, il sera définitivement supprimé
- déplacer un message du dossier « spam » vers le dossier « reçus »
- archiver un message, il faudra le déplacer vers le dossier « archives »
- créer un message et le stocker dans le dossier « brouillons » avant de l'envoyer, dès qu'il sera envoyé, il passe automatiquement dans le dossier « envoyés ».
- retourner le nombre de messages reçus (lus et non lus), le nombre de messages envoyés, le nombre de messages dans spam, ... etc.
- retourner l'espace de stockage utilisé, l'espace de stockage restant dans la boîte

Une alerte sera générée si la taille des messages dans une boîte e-mail (tous dossiers confondus et pièces jointes comprises) atteint 80% de la capacité de stockage de la boîte. L'alerte doit prévenir qu'il faut supprimer certains messages.

### **Implémentation des classes**

Pour chacune des classes su-décrites, donner une implémentation en java. Chaque classe doit comporter : des constructeurs, une méthode toString (redéfinie), une méthode equals (redéfinie), une méthode saisir, les accesseurs et les autres méthodes nécessaires à la manipulation des objets de la classe. Les méthodes de chaque classe doivent être déduites à partir de la description donnée plus haut.

### PS :

- utiliser l'héritage dans l'implémentation des classes (classes abstraites éventuellement aussi)
- utiliser les collections pour les attributs de type « liste, ensemble, groupe, ... »
- *Il faut respecter le principe d'encapsulation.*

### **Utilisation des classes implémentées**

Une fois les classes implémentées avec tous les traitements possibles (la partie méthodes à définir pour chaque classe), on demande de créer dans un package externe, une classe Application qui doit utiliser ces classes.

La classe **Application** (AppMessagerie) doit comporter une méthode main qui fait appel à une méthode menu (définie dans la classe Application). La méthode « menu » affiche un menu de manière itérative (boucle), pour répondre aux requêtes de l'utilisateur.

Le menu doit comporter les options suivantes :

1. Création des adresses e-mail (avec les profils)
2. Ajouter/suppr/modifier une adresse
3. Affichage des adresses créées (par catégorie)
4. Création des boîtes e-mail correspondantes
5. Ajouter une boîte
6. Affichage des boîtes e-mails (avec leur contenu)
7. Gestion des boîtes e-mails
8. Quitter

### **Indications**

- A la création d'une boîte mail, tous les dossiers sont vides sauf un message de « Bienvenue » dans le dossier « reçus ».
- les catégories d'adresses, on peut avoir un affichage par site (gmail, usthb, yahoo, ...), par type (professionnel ou non)
- pour l'option « modifier », on peut modifier le mot de passe, le pseudo ou le profil
- la création se fait par programme (des méthodes CréerAdresses , CréerBoites, ..., dans la classe Application. On permettra ensuite, l'ajout...

L'option « Gestion des boîtes e-mails » donne lieu à un sous-menu

1. Ajout automatique de messages (reçus, envoyés, brouillons)
2. Afficher le contenu d'une boîte (en séparant les dossiers, en donnant des infos espace utilisé et espace restant)
3. Envoyer message (une exception doit être générée si une adresse de destinataire n'existe pas dans le répertoire des adresses)
4. Afficher le contenu d'un message (avec l'en-tête)
5. Supprimer un message
6. Archiver un message / ou les messages reçus avant une date donnée
7. Restaurer un message donné (restituer de la corbeille)
8. répondre à un message
9. vider un dossier (« spam » ou « envoyés »)

10. trier les messages par date (le plus récent d'abord), par objet (ordre alphabétique)
11. Quitter

Autres requêtes :

1. Afficher toutes les boîtes (adresses mail) ayant reçu un message donné (on donnera l'objet du message)
2. Afficher les boîtes qui sont remplies à plus de 50% de leur capacité
3. Eclater la collection des boîtes en deux collections :  
la première contenant les boîtes publiques et l'autre contenant les boîtes professionnelles
4. Pour un site donné, calculer le pourcentage d'utilisation par la catégorie d'âge comprise entre 18 et 35 ans.
5. Afficher les messages ayant des pièces jointes (pour une boîte donnée)
6. Rechercher les messages par destinataire, par expéditeur, par mot clé.
7. Afficher les noms, prénoms des profils ayant au moins deux boîtes de messagerie.
8. Vider toutes les boîtes d'un site donné (piratage du site)

### **Remarques importantes**

Un effort de création des *données à mettre dans les structures* doit être fourni, en donnant des adresses correctes ([l2acadA@gamil.com](mailto:l2acadA@gamil.com), [sboukhedouma@usthb.dz](mailto:sboukhedouma@usthb.dz), ...), des messages avec des objets et des textes corrects (par exemple : objet « Cours POO », le contenu : « un cours est programmé samedi à 10h00 »).

Créer au moins une dizaine de boîtes avec 5 à 10 messages dans chaque catégorie, les messages peuvent se répéter.

Les noms de classes, d'attributs et de méthodes doivent être significatifs.

Les affichages doivent être bien travaillés et lisibles

### **A remettre**

#### **Dans un fichier word (ou pdf)**

- La description des classes : attributs (avec leurs types) et signatures de méthodes (plus une brève explication pour chaque méthode sans donner le corps de la méthode), les liens d'héritage, ...
- Scénarios d'exécution : captures d'exécution pour les principales requêtes (avec commentaires).

#### **Répertoire à part**

- le code source des classes (bien structuré et bien commenté)

**Bon courage**