

Introduction à la programmation 420-309-AH

TP 3

- - Les TPs peuvent être effectués seul ou en équipe de deux.
- - La date limite de remise de ce TP sera le 30 mars à 23h59.
- - La remise se fait sur LÉA (Omnivox).
- - Il faudra remettre 1 seul fichier .py.
- - Veuillez indiquer sur le fichier les noms des membres de votre équipe. Ex. PrenomNom1_PrenomNom2.py
- - Toutes documentations permises
- - Il faut compléter le code troué donné.
- - Il est essentiel que votre code s'exécute correctement.

Énoncé :

Vous devez écrire un programme Python qui va simuler un distributeur automatique de billets. Ce programme devra permettre à un utilisateur de retirer de l'argent à partir de son compte en banque en suivant les étapes suivantes :

1. Le programme va commencer par afficher un message de bienvenue et demander à l'utilisateur d'entrer son numéro de compte.
2. Le programme va vérifier si le numéro de compte entré par l'utilisateur est valide. Si le numéro de compte n'est pas valide, le programme doit afficher un message d'erreur .
3. Si le numéro de compte est valide, le programme va demander à l'utilisateur de saisir son code PIN. Si le code PIN est incorrect, le programme doit afficher un message d'erreur.
4. Si le code PIN est correct, le programme va afficher un menu d'options qui permet à l'utilisateur de choisir le montant qu'il souhaite retirer. Les options devraient être les suivantes :
 - 1. Retirer 20 \$
 - 2. Retirer 50 \$
 - 3. Retirer 100 \$
 - 4. Retirer un montant personnalisé
5. Si l'utilisateur choisit l'option 1, 2 ou 3, le programme doit effectuer la transaction et afficher un message de confirmation indiquant que le montant a été retiré avec succès. Si l'utilisateur choisit l'option 4, le programme doit lui demander le montant qu'il souhaite retirer.
6. Vous devez aussi valider que l'utilisateur a assez d'argent dans son compte pour effectuer la transaction.
7. Vous devez ensuite afficher à l'utilisateur son solde.

Le programme principal est déjà fait pour vous. Vous devez vous assurer que le code s'exécute correctement.

Vous devez écrire les fonctions décrites dans les TODO du fichier tp3.py.

Les comptes utilisés sont dans le fichier comptes.py. Il contient un dictionnaire avec les données relatives à chaque compte.

Voici les fonctions que vous devez écrire :

1. Écrire une fonction nommée **is_account_valid** qui vérifie si un numéro de compte est valide. Cette fonction prendra en paramètre un numéro de compte (integer) et retournera un booléen (True ou False).

2. Écrire une fonction nommée `is_pin_valid` qui vérifie si le code PIN entré par l'utilisateur est correct. Cette fonction prendra en paramètre un numéro de compte (integer) et un code pin (string) et retournera un dictionnaire qui correspond aux informations du compte si le code pin est valide. Si le code pin est invalide, cette fonction retournera None.
3. Écrire une fonction nommée `call_withdrawl` qui gère les options de retrait. Cette fonction prendra en entrée un chiffre qui correspond à l'option choisie et le dictionnaire qui correspond au compte de l'utilisateur. Si l'utilisateur entre une option valide, on appelle la fonction `withdrawl` avec le retrait voulu. Sinon, on affiche une erreur et on retourne.
4. Écrire une fonction nommée `is_withdrawl_possible` qui valide si le retrait est possible. Cette fonction prendra en entrée le compte de l'utilisateur (un dictionnaire), et le montant du retrait. Cette fonction retournera True si l'argent disponible est plus grand égal au montant du retrait. Elle retournera False dans le cas contraire.
5. Écrire une fonction nommée `withdrawl` qui effectue un retrait. Cette fonction prendra en entrée le compte de l'utilisateur (un dictionnaire) et le montant du retrait. Cette fonction vérifiera que le solde est assez grand pour effectuer le retrait. Si ce n'est pas le cas, elle affichera à l'écran un message d'erreur et retournera le dictionnaire sans avoir effectué d'opération. Cette fonction retournera le dictionnaire qui correspond au compte après l'opération.
6. Écrire une fonction nommée `get_account_number` qui demande à l'utilisateur son numéro de compte et le retourne sous forme d'Integer (int).
7. Écrire une fonction nommée `get_pin` qui demande à l'utilisateur son code pin et le retourne sous forme de string.
8. Écrire une fonction nommée `get_choice` qui affiche à l'utilisateur les options de retrait et lui demande son choix. Cette fonction retourne le choix de l'utilisateur sous forme d'integer (int).
9. Écrire une fonction nommée `get_account_balance` qui prend en entrée le dictionnaire qui correspond au compte de l'utilisateur. Cette fonction retourne le solde d'un utilisateur.
10. Écrire une fonction nommée `get_withdrawl_amount` qui demande à l'utilisateur quel montant il veut retirer. Cette fonction retourne le montant entré sous forme de float.

Il est important de bien organiser le code en utilisant des fonctions pour chaque tâche spécifique afin de rendre le programme facile à lire et à comprendre.