

Génie logiciel

Travail pratique #1

Programmation orientée-objet avec Java

Professeur : HO Tuong Vinh
Date énoncée: 18/1/2018
Date de remise : 8/2/2018

Une application simple de gestion bancaire

Ce travail a pour but de vous rappeler les concepts de la modélisation avec UML et programmation **orientée-objet** avec Java. De plus, il vous fait familiariser à un environnement de développement intégré (IDE) « libre » (Open Source) ECLIPSE. Il vous est demandé aussi d'utiliser GitHub (github.com) pour la gestion de source code de votre projet.

Vous avez à réaliser un programme en **Java** et à rédiger un rapport qui décrit votre travail.

1. PROGRAMMATION

1.1 Spécification de l'application

Voici la description du système à développer :

- La banque gère un ensemble de compte.
- Un client peut posséder plusieurs comptes avec des taux d'intérêt différents.
- Chaque compte contient les informations suivantes :
 - le numéro de compte, un entier positif, qui sera indiqué par la banque,
 - le numéro d'identification du titulaire du compte, un entier positif,
 - le nom du titulaire,
 - le solde du compte, un nombre flottant,
 - le taux d'intérêt, un nombre flottant entre 0 et 100.
- La banque peut exécuter les opérations suivantes :
 - créer un nouveau compte,
 - déposer un certain montant d'argent dans un compte,
 - retirer un certain montant d'argent d'un compte,
 - consulter le solde d'un compte,
 - calculer l'intérêt pour tous les comptes et mettre à jours leur solde,
 - produire un rapport qui inclut une liste de comptes avec les informations suivantes : le numéro de compte, le solde, et les transactions effectuées (déposer et retirer l'argent) depuis que le compte est créé.
 - faire une recherche qui permet, à partir du numéro d'identification d'une personne, de savoir si elle possède un ou des comptes dans la banque.
- En ce qui concerne les transactions d'un compte spécifique, les informations suivantes sont intéressantes :
 - la date de transaction,
 - le type de transaction : déposer ou retirer d'argent,
 - le montant d'argent.

Exemple des données de test pour le programme :

---- créer un nouveau compte pour Alice avec :

le numéro de compte = 001 ; nom = Alice, le numéro d'identification = 901 ; le montant = 100 Euros et le taux d'intérêt = 5%.

---- créer un nouveau compte pour Bob avec :

le numéro de compte = 002 ; nom = Bob, le numéro d'identification = 902 ; le montant = 50 Euros et le taux d'intérêt = 5%.

---- créer un nouveau compte pour Alice avec :
le numéro de compte = 003 ; nom = Alice, le numéro d'identification = 901 ; le montant = 200 Euros et le taux d'intérêt = 10%.

---- créer un nouveau compte pour Eve avec :
le numéro de compte = 004; nom = Eve, le numéro d'identification = 903 ; le montant = 200 Euros et le taux d'intérêt = 10%.

---- déposer un montant de 100 Euros pour le compte 001 au 15 juillet 05
---- déposer un montant de 100 Euros pour le compte 001 au 31 juillet 05
---- déposer un montant de 150 Euros pour le compte 002 au 1 juillet 05
---- déposer un montant de 150 Euros pour le compte 002 au 15 juillet 05
---- déposer un montant de 200 Euros pour le compte 003 au 5 juillet 05
---- déposer un montant de 250 Euros pour le compte 004 au 31 juillet 05

---- retirer un montant de 10 Euros depuis le compte 001 au 10 juillet 05
---- retirer un montant de 10 Euros depuis le compte 002 au 15 juillet 05
---- retirer un montant de 10 Euros depuis le compte 003 au 31 juillet 05
---- retirer un montant de 10 Euros depuis le compte 004 au 31 juillet 05

1.2 Contraintes de programmation

- Il faut programmer de façon orientée objet.
- Il faut assurer que la communication entre l'utilisateur et le système est conviviale.
- Les consignes de codage doivent être respectées
(<https://google-styleguide.googlecode.com/svn/trunk/javaguide.html>).

2. RAPPORT

Le rapport doit aborder les points suivants :

- Une introduction de votre travail
- La spécification de l'application
- La conception
- L'implémentation et test
- Conclusion (auto-évaluation de votre travail)
- Code source (en annexe)

3. EVALUATION

Le programme sera évalué sur 65% et le rapport sur 35%. Un programme qui plante à l'exécution, même dans une situation extrême respectant la spécification haute, se verra attribuer zéro sur 65%.