Übung Java Account Rate static

Ausgangslage

Diese Übung ist die Fortsetzung der Übung Account (Class).In einem ersten Schritt sind die Attribute als private zu deklarieren und die set/get-Methoden zu programmieren.In einem zweiten Schritt sollen alle notwendigen Konstruktoren (inkl. default constructor) programmiert werden.In einem dritten Schritt soll der Zinssatz (rate) als Klassenvariable (static) definiert werden. Weiter sollen die Klassenmethoden setRate(double rate) und double getRate () erstellt werden. Das folgende UML Klassendiagramm zeigt die komplette Klassen auf:

Vorgehen

Lösen Sie bitte die Aufgabe wie folgt:Öffnen Sie die Klasse Account. In Eclipse können Sie mit dem Key Ctrl-Shift-T nach Java Klassen suchen.Markieren Sie alle Attribute als private oder verwenden Sie das Refactoring "Encapsulate Field".Programmieren Sie die set/get-Methoden aus, falls nicht schon automatisch via Refactoring erstellt.Programmieren Sie die Konstruktoren. Überlegen Sie sich welche Konstruktoren in der Praxis angwendet würden.Passen Sie die Klasse TestAccount entsprechend an, so dass solche wieder kompiliert und ausgeführt werden kann.Definieren Sie das Klassenattribute double rate in der Klasse Account. Klassenattribute werden in der Regel vor den Member Attributen definiert.Programmieren Sie die Getter-, Setter-Klassenmethoden void setRate(double rate) und double getRate () aus.Erweitern Sie die Klasse TestAccount indem Sie 2 Account Instanzen erstellen und geben Sie den Zinssatz an die Konsole aus.Ändern Sie den Zinssatz und geben Sie alsdann für beide Instanzen den Zinssatz erneut aus.Untersuchen Sie bitte das Verhalten. Was stellen Sie fest ?

Lösung

Eine mögliche Lösung finden Sie hier

Kontakt

Simtech AG
Finkenweg 23
3110 Münsingen
Schweiz

Impressum

Das Copyright für sämtliche Inhalte dieser Website liegt bei Simtech AG, Schweiz.
Beachten Sie auch unsere Hinweise zum Urheberrecht, Datenschutz und Haftungsauschluss.
Jeder Hinweis auf Fehler nehmen wir gerne entgegen.

Copyright

2024 Simtech AG, All rights reserved, Powered by stack.ch written in Golang by Daniel Schmutz

https://www.simtech-ag.ch/überlegen