

Project Bank Management

Γενική Περιγραφή

Στην υλοποίησή μας η τράπεζα την πρώτη μέρα λειτουργίας της δημιουργεί 7 λογαριασμούς (Accounts) με τυχαίους χρήστες, στη μέθοδο getRandomOwners επιστρέφει 1-3 τυχαίους πελάτες (Clients) αφού γίνει ποιο πριν έλεγχος ότι δεν έχει επιλεγεί κάποιος από αυτούς δεύτερη φορά (duplicate), επίσης όταν καλείτε να φτιαχτεί ένας νέος λογαριασμός υπάρχει πιθανότητα: 40% ταμιευτηρίου (SimpleCreditAccount), 40% υπέρ-ταμιευτηρίου (SuperCreditAccount) και 20% προθεσμίας (ReserveAccount). Στη προσομοίωση κάθε μέρα εκτελούνται από 1-3 συναλλαγές (Transactions), η κάθε συναλλαγή από αυτές έχει πιθανότητα: 15% άνοιγμα νέου λογαριασμού (openAccount), 40% κατάθεση (deposit), 40% ανάληψη (withdraw), 5% κλείσιμο λογαριασμού (closeAccount) και στο τέλος κάθε ημέρας γίνεται ο τοκισμός (payInterest) όπου χρειάζεται και οι κλειστοί λογαριασμοί μεταφέρονται στο φάκελο που ακολουθεί τη διαδρομή: ..\BANK_FILES\Bank_Database\Accounts\Closed.

Όλα τα παραπάνω ποσοστά υποθέτουμε ότι είναι χωρίς βλάβη της γενικότητας.

Στο restore mode γίνεται ανάκτηση από τα αρχεία των προηγούμενων εκτελέσεων και δίνεται ιδιαίτερη έμφαση στη διατήρηση του σωστού IBAN (αλλαγή static πεδίου με βάση το τελευταίο IBAN της τελευταίας εκτέλεσης).

Τέλος, αφού γίνει η οποιαδήποτε λειτουργία ενημερώνονται το υπόλοιπο και όλα τα αρχεία, ποιο συγκεκριμένα ξαναγράφονται τα Details.txt, Account.bin και γράφουμε στο τέλος των xxxxxx.txt και xxxxxx.bin (όπου xxxxxx ο αριθμός IBAN του κάθε λογαριασμού) που είναι οι συναλλαγές. Αυτό επιτυγχάνετε καλώντας μέσα στις μεθόδους των λειτουργιών την update (με όρισμα το αντίγραφο -clone του τρέχοντος λογαριασμού).

Υλοποίηση χωρίς ArrayList <Transaction> transactions

Μη θέλοντας να αλλάξουμε σε πολύ μεγάλο βαθμό το πρόγραμμα μετά από αυτήν την αλλαγή τροποποιήσαμε όπου υπήρχε η μέθοδος getTransaction σε getTransactionFromFile, μέσα στην οποία καλούσαμε την File_Management.TransactionLoader.allTransactions (τρέχων λογαριασμός από αντιγραφή -clone) η οποία διαβάζει το αρχείο με τις συναλλαγές και δημιουργεί μια λίστα με αυτές δυναμικά (ανά κλήση μόνο σαν buffer from file). Τέλος τροποποιήσαμε την File_Management.Update.ActiveAccountFolder να δέχεται σαν όρισμα την συναλλαγή από κει που καλείτε.

Χρονική διάρκεια εκτέλεσης των προγραμμάτων

Οι παρακάτω μετρήσεις έγιναν σε υπολογιστή με μνήμη 8gb και Intel Core i-5 στα 3.30 GHz και είναι ενδεικτικές. Οι παρακάτω χρονικές διάρκειες εκτέλεσης του προγράμματος είναι μεγάλες γιατί αφενός έχουμε πολλούς λογαριασμούς και γίνονται συχνά update, αφετέρου το restore mode επιβαρύνει τη διάρκεια εξαιτίας του φορτώματος των αρχείων της προηγούμενης κατάστασης.

Έκδοση	1 έως 360 μέρες	x έως x+90 (restore mode) μέρες
Standard	35sec	55sec
Bonus	30sec	38sec

ΣΥΜΠΕΡΑΣΜΑ: οι ημέρες που δίνει ο χρήστης καλό θα ήταν να είναι λιγότερες από αυτές (για την πρώτη εκτέλεση: 150-180 μέρες και τις επόμενες σε restore mode 30-50 μέρες) με αποτέλεσμα να εκτελεστεί ποιο γρήγορα το πρόγραμμα.