

Αυτόματη Ταμειολογική Μηχανή (Automated Teller Machine, ATM)

ΣΥΝΤΕΛΕΣΤΕΣ: Σπανουδάκης Σταύρος (Α.Ε.Μ.: ...),
Φιλιππίδης Χρήστος (Α.Ε.Μ.: ...)

Μάθημα: Δομημένος Προγραμματισμός

Ακαδημαϊκό έτος: 2022 - 2023

Τμήμα: Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Σχολή: Πολυτεχνική

Πανεπιστήμιο: Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Περιεχόμενα

<u>Σύντομη περιγραφή</u>	σελ. 3
<u>Οδηγίες μεταγλώττισης, εγκατάστασης και εκτέλεσης</u>	σελ. 4
<u>Οδηγίες χρήσης</u>	σελ. 5
<u>Διαγραμματική περιγραφή προγράμματος</u>	σελ. 9
<u>Συναρτήσεις</u>	σελ. 11
<u>Αναφορές</u>	σελ. 17
<u>Άδεια δικαιωμάτων για ανοικτό ελεύθερο κώδικα</u>	σελ. 18

Σύντομη Περιγραφή

Το παρόν έγγραφο αναφέρεται στο πρόγραμμα “BOP_ATM.cbp” και παρουσιάζει τα χαρακτηριστικά του. Πρόκειται για ένα πρόγραμμα προσομοίωσης βασικών λειτουργιών μιας αυτόματης ταμειολογιστικής μηχανής, γραμμένο στην γλώσσα C. Ο χρήστης δύναται να χρησιμοποιήσει τον λογαριασμό του ή να δημιουργήσει έναν νέο, για την πραγματοποίηση χρηματικών συναλλαγών - συγκεκριμένα: κατάθεση, ανάληψη και μεταφορά. Προσφέρεται, επίσης, η δυνατότητα στον χρήστη να ελέγξει το υπόλοιπο του λογαριασμού του, τα στοιχεία του και το ιστορικό συναλλαγών του. Ο χρήστης εισάγει τα εκάστοτε απαιτούμενα δεδομένα και πραγματοποιεί επιλογές πληκτρολογώντας χαρακτήρες, ενώ η αντίστοιχη έξοδος των δεδομένων τυπώνεται σε ένα παράθυρο κονσόλας. Τα στοιχεία του κάθε λογαριασμού, καθώς και το σύνολο των συναλλαγών που έχουν λάβει χώρα, αποθηκεύονται σε σχετικά αρχεία.

Οδηγίες μεταγλώττισης, εγκατάστασης και εκτέλεσης

Το πρόγραμμα “BOP_ATM.cbp” (και το αντίστοιχο εκτελέσιμο αρχείο) λειτουργεί μόνο σε λογισμικό windows. Για την εκτέλεσή του, θα πρέπει να έχετε κατεβασμένο στον υπολογιστή σας τον φάκελο “BOP_ATM” και κάποιον μεταγλωττιστή για τη γλώσσα προγραμματισμού C. Στη συνέχεια δίνονται οδηγίες σχετικά με τον μεταγλωττιστή GNU Compiler Collection (GCC).

I. Command prompt

Αρχικά, ανοίξτε το Command Prompt πληκτρολογώντας: `windows key > Command Prompt > ENTER`. Θα πρέπει τώρα να βλέπετε μπροστά σας το παράθυρο του Command Prompt.

Έπειτα, πληκτρολογήστε `cd`, αφήστε ένα `κενό` και γράψτε το `μονοπάτι` (path) στο οποίο έχετε αποθηκεύσει τον φάκελο “BOP_ATM”. Πατήστε ENTER.

Για τη μεταγλώττιση, τώρα, θα πρέπει να πληκτρολογήσετε την παρακάτω εντολή και να πατήσετε ENTER:

```
gcc -c ATM.c user_interface.c file_access.c
```

Ενώ για τη διασύνδεση και την παραγωγή του εκτελέσιμου αρχείου BOP_ATM.exe, την ακόλουθη εντολή, ακολουθούμενη και πάλι από ENTER:

```
gcc ATM.o user_interface.o file_access.o -o BOP_ATM.exe
```

Τέλος, για την εκτέλεση του προγράμματος πληκτρολογήστε `BOP_ATM.exe` και αφού πατήσετε ENTER, η αρχική οθόνη του προγράμματος θα πρέπει να εμφανισθεί στην οθόνη σας.

II. CodeBlocks

Μία εναλλακτική προσέγγιση είναι η χρήση του ολοκληρωμένου περιβάλλοντος ανάπτυξης CodeBlocks. Ανοίξτε το αρχείο “BOP_ATM.cbp” μέσω της εφαρμογής CodeBlocks και, αφού επιλέξετε το κύριο αρχείο `ATM.c`, χρησιμοποιήστε την λειτουργία “Build and run”, ώστε να μεταγλωττίσετε και να τρέξετε το πρόγραμμα.

Οδηγίες Χρήσης

Αλληλεπίδραση

Μετά την είσοδο δεδομένων από το πληκτρολόγιο, είναι απαραίτητο να πατήσετε το πλήκτρο ENTER, προκειμένου να καταχωρηθούν στο πρόγραμμα. Τα δεδομένα που πληκτρολογείτε τυπώνονται στην οθόνη.

Ακύρωση της συναλλαγής/λειτουργίας

Αν σε οποιοδήποτε στάδιο της εικονικής λειτουργίας της ΑΤΜ επιθυμείτε τον τερματισμό της τρέχουσας λειτουργίας και την έξοδο από τον λογαριασμό αρκεί να πληκτρολογήσετε τρεις φορές τον χαρακτήρα '%', δηλαδή «%%%», χωρίς τα εισαγωγικά. Το πρόγραμμα επαναφέρεται στο αρχικό μενού (WELCOME MENU).

Έκφραση πνευματικής ιδιοκτησίας

Κατά την εκκίνηση του προγράμματος, τυπώνεται στην οθόνη σημείωμα περί των πνευματικών δικαιωμάτων των δημιουργών του προγράμματος. Για επακριβή αναφορά στην έλλειψη εγγύησης που φέρει η χρήση του προγράμματος, πληκτρολογήστε τον χαρακτήρα 'w'. Για την εκτέλεση του κυρίου μέρους του προγράμματος πληκτρολογήστε τον χαρακτήρα 'i'. Οι όροι, όπως προβλέπονται από το πρότυπο της άδειας GNU GPL v.3, υπό τους οποίους επιτρέπεται διαμοιρασμός του προγράμματος, αναφέρονται στην ενότητα «Άδεια δικαιωμάτων για ελεύθερο κώδικα».

WELCOME MENU

Η λειτουργία της εικονικής ΑΤΜ, ξεκινά με το παράθυρο του αρχικού μενού. Για να δημιουργήσετε νέο λογαριασμό πληκτρολογήστε, τον χαρακτήρα 'r'. Για να συνδεθείτε σε υπάρχων λογαριασμό, πληκτρολογήστε τον χαρακτήρα 'i'.

Register (Δημιουργία λογαριασμού)

Προκειμένου να δημιουργήσετε έναν νέο λογαριασμό εισάγετε τα απαραίτητα στοιχεία, με σειρά αίτησης: όνομα (name), επώνυμο (surname), όνομα χρήστη (username), προσωπικό αριθμό ταυτοποίησης (PIN) και αριθμό φορολογικού μητρώου (ΑΦΜ) (Tax Identification Number - TIN). Το όνομα θα πρέπει να έχει μέγεθος μεταξύ 2 και 16 χαρακτήρων. Το επώνυμο μεταξύ 2 και 24 χαρακτήρων και το όνομα χρήστη μεταξύ 4

και 16. Το PIN πρέπει να έχει μέγεθος ακριβώς 4 ψηφίων και το ΑΦΜ ακριβώς 9 ψηφίων. Πρόσθετα, δεν είναι επιτρεπτό το όνομα χρήστη να περιλαμβάνει αποκλειστικά αριθμούς ή να ταυτίζεται με αυτό διαφορετικού λογαριασμού, ενώ το όνομα και το επώνυμο δεν επιτρέπεται να περιλαμβάνουν αριθμούς. Αντίστοιχα, τόσο το PIN, όσο και το ΑΦΜ απαγορεύεται να περιέχουν γράμματα. Σε περίπτωση μη επιτρεπτών δεδομένων, αναγράφεται το αντίστοιχο μήνυμα σφάλματος και ακούγεται ο ήχος σφάλματος.

Login (Σύνδεση σε λογαριασμό)

Προκειμένου να συνδεθείτε σε υπάρχων λογαριασμό εισάγετε το όνομα χρήστη σας και έπειτα το PIN σας. Διαφορετικά, εισάγετε τον αριθμό λογαριασμού σας (ID) και έπειτα το PIN σας.

MAIN MENU

Μετά την ολοκλήρωση της επιτυχούς σύνδεσης σε λογαριασμό ή της δημιουργίας νέου, τυπώνεται το βασικό μενού (MAIN MENU) του προγράμματος, το οποίο προβάλλει τις διαθέσιμες λειτουργίες της ΑΤΜ. Επιλέξτε την επιθυμητή, πληκτρολογώντας το αντίστοιχο γράμμα. Αναφορικά οι λειτουργίες και ο χαρακτήρας επιλογής τους, είναι:

- Deposit (κατάθεση), με χαρακτήρα επιλογής: 'd'
- Withdrawal (ανάληψη), με χαρακτήρα επιλογής: 'w'
- Transfer (μεταφορά), με χαρακτήρα επιλογής: 't'
- Check Balance (έλεγχος υπολοίπου), με χαρακτήρα επιλογής: 'b'
- Transaction History (ιστορικό συναλλαγών), με χαρακτήρα επιλογής: 'h'
- Check Credentials (έλεγχος στοιχείων λογαριασμού), με χαρακτήρα επιλογής: 'c'

Deposit (Κατάθεση)

Πληκτρολογήστε το επιθυμητό ποσό προς κατάθεση. Προσθέστε στην συνέχεια μία σύντομη περιγραφή της συναλλαγής (προαιρετικά – ENTER για παράλειψη). Το ενημερωμένο υπόλοιπο του λογαριασμού τυπώνεται στην οθόνη.

Withdrawal (Ανάληψη)

Πληκτρολογήστε το επιθυμητό ποσό προς ανάληψη, το οποίο δεν θα πρέπει να ξεπερνά το τρέχων υπόλοιπο του λογαριασμού. Προσθέστε στη συνέχεια μία σύντομη περιγραφή της συναλλαγής (προαιρετικά – ENTER για παράλειψη). Το ενημερωμένο υπόλοιπο του λογαριασμού τυπώνεται στην οθόνη.

Transfer (Μεταφορά)

Πληκτρολογήστε το IBAN του λογαριασμού στον οποίο επιθυμείτε να μεταφέρετε χρήματα. Έπειτα, πληκτρολογήστε το επιθυμητό ποσό προς μεταφορά, το οποίο δεν θα πρέπει να ξεπερνά το τρέχων υπόλοιπο του λογαριασμού. Προσθέστε, τέλος, μία σύντομη περιγραφή της συναλλαγής (προαιρετικά – ENTER για παράλειψη). Το ενημερωμένο υπόλοιπο του λογαριασμού τυπώνεται στην οθόνη.

Check Balance (Έλεγχος υπολοίπου)

Το τρέχων υπόλοιπο του λογαριασμού τυπώνεται στην οθόνη.

Transaction History (Ιστορικό συναλλαγών)

Το πρόγραμμα τυπώνει το μενού ιστορικού συναλλαγών (Transaction History Menu), βλ. αντίστοιχη αναφορά παρακάτω.

Check Credentials (Έλεγχος στοιχείων λογαριασμού)

Τυπώνονται όλα τα στοιχεία του λογαριασμού του συνδεδεμένου χρήστη, δηλαδή το όνομα (name), το επώνυμο (surname), το όνομα χρήστη (username), ο προσωπικός αριθμός ταυτοποίησης (PIN), ο ΑΦΜ (TIN), ο διεθνής λογαριασμός τραπέζης (IBAN) και ο αριθμός λογαριασμού (ID).

TRANSACTION HISTORY MENU

Το πρόγραμμα μεταβαίνει στο παράθυρο Transaction History Menu, όταν η αντίστοιχη λειτουργία επιλεγθεί από το βασικό μενού (MAIN MENU). Για την τύπωση του ιστορικού

καταθέσεων, πληκτρολογήστε τον χαρακτήρα 'd'. Για την τύπωση του ιστορικού αναλήψεων, πληκτρολογήστε τον χαρακτήρα 'w'. Για την τύπωση του ιστορικού μεταφορών πληκτρολογήστε τον χαρακτήρα 't'. Σε κάθε περίπτωση, ανακτώνται από τα αρχεία οι απαραίτητες πληροφορίες και τυπώνονται στην οθόνη. Οι πληροφορίες αυτές περιλαμβάνουν το ποσό, την ημερομηνία και την περιγραφή της συναλλαγής. Στην περίπτωση των μεταφορών αναγράφεται και το IBAN του αποστολέα ή δέκτη της εκάστοτε μεταφοράς.

Περαιτέρω χρήση

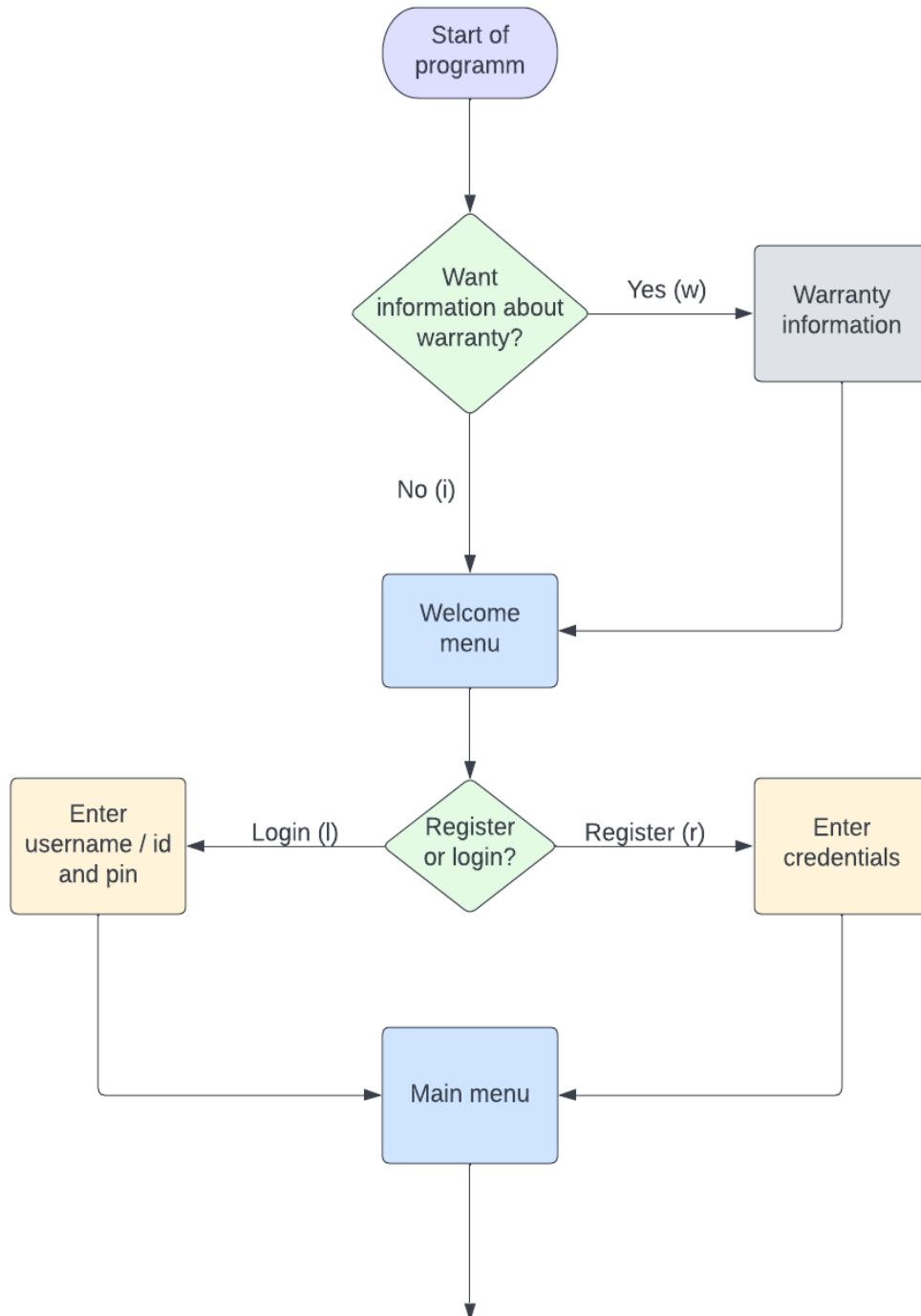
Έπειτα από την ολοκλήρωση οποιασδήποτε από τις διαθέσιμες λειτουργίες είναι δυνατή η έναρξη νέας λειτουργίας. Τυπώνεται στην οθόνη σχετικό μήνυμα. Πληκτρολογήστε τον χαρακτήρα 'y' για να επιστρέψετε στο βασικό μενού (MAIN MENU) και να επιλέξετε νέα λειτουργία. Διαφορετικά, πληκτρολογήστε τον χαρακτήρα 'n' ώστε να αποσυνδεθεί ο λογαριασμός. Σε αυτήν την περίπτωση το πρόγραμμα επιστρέφει στο αρχικό μενού (WELCOME MENU), εν αναμονή νέου χρήστη.

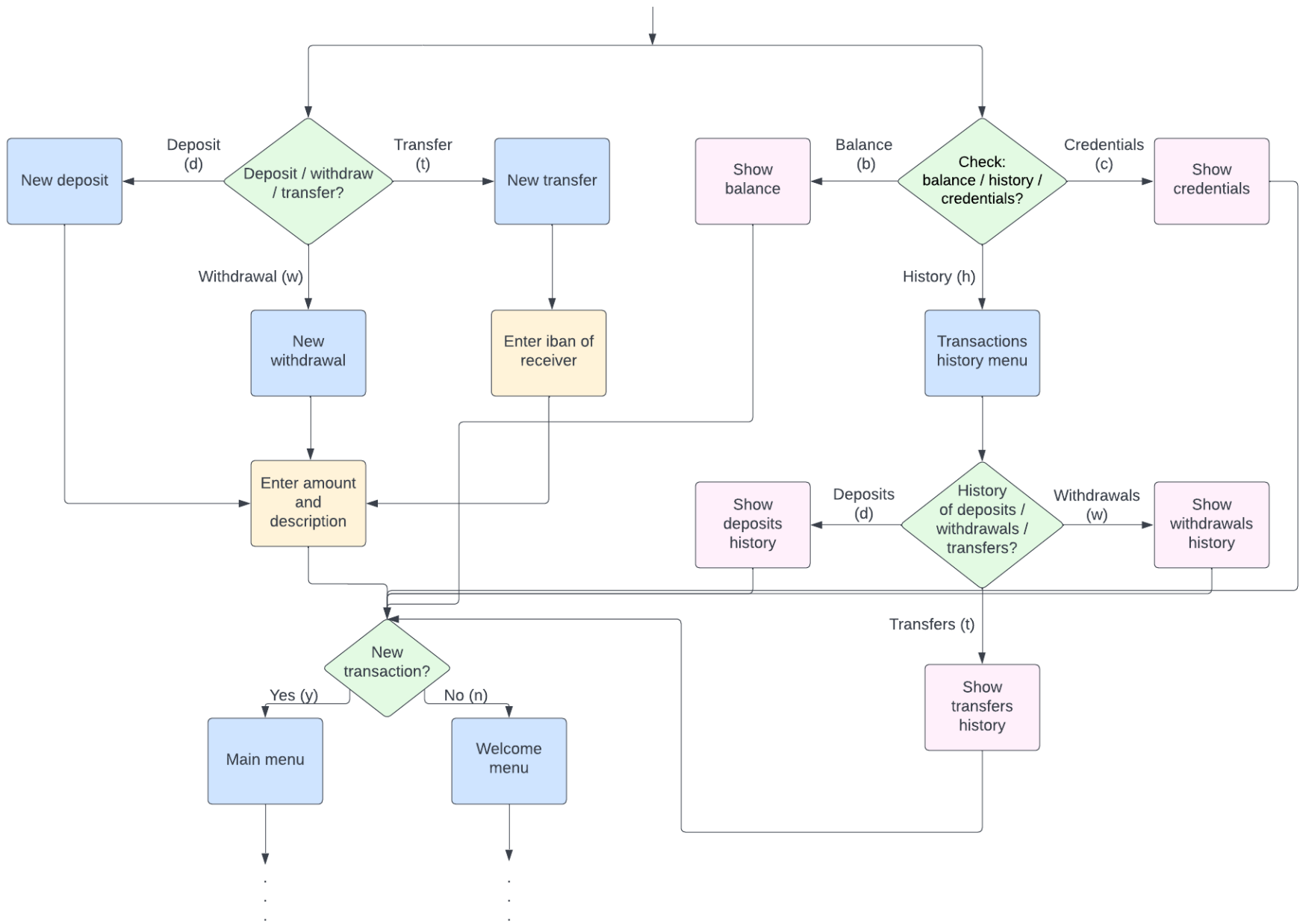
Πίνακας προεγκατεστημένων λογαριασμών

Προκειμένου να διευκολυνθεί η κατανόηση των λειτουργιών του προγράμματος, έχουν προεγκατασταθεί τα στοιχεία τεσσάρων λογαριασμών, τα οποία παρατίθενται παρακάτω. Η χρήση τους είναι προαιρετική.

Name	Surname	Username	PIN	ID	IBAN
Dennis	Ritchie	first	1972	9916057485646156	4348156251189365
Alex	Delion	KingNY	1010	8466090217208159	3898199083741369
Chuck	Norris	administrator	1111	8299822613380392	3621921489823502
Dory	Nemo	notfound	6824	5673506565752441	3005605331295651

Διαγραμματική περιγραφή προγράμματος





Συναρτήσεις

I. User Interface

void welcome(void);

Υποδέχεται τον χρήστη και τον παροτρύνει να εισάγει τον συνδυασμό '%%%' οποιαδήποτε στιγμή θέλει να τερματίσει το πρόγραμμα. Η συνάρτηση δεν δέχεται παραμέτρους και δεν επιστρέφει τιμή.

void register_or_login(char *choice_r_l);

Παροτρύνει τον χρήστη να επιλεξει αν θελει να δημιουργήσει καινούργιο λογαριασμο ή να συνδεθει σε ήδη υπάρχον και μετά τον απαραίτητο έλεγχο, αποθηκεύει την επιλογή του. Η συνάρτηση δέχεται σαν παράμετρο έναν δείκτη στην επιλογή του χρήστη και δεν επιστρέφει τιμή.

void yes_or_no(char *choice_y_n);

Παροτρύνει τον χρήστη να επιλεξει αν θελει να κάνει νέα συναλλαγή ή να αποχωρήσει και μετά τον απαραίτητο έλεγχο, αποθηκεύει την επιλογή του. Η συνάρτηση δέχεται σαν παράμετρο έναν δείκτη στην επιλογή του χρήστη και δεν επιστρέφει τιμή.

void select_history(char *choice_d_w_t);

Παροτρύνει τον χρήστη να επιλεξει αν θέλει να δει το ιστορικό των καταθέσεων, των αναλήψεων ή των μεταφορών και μετά τον απαραίτητο έλεγχο, αποθηκεύει την επιλογή του. Η συνάρτηση δέχεται σαν παράμετρο έναν δείκτη στην επιλογή του χρήστη και δεν επιστρέφει τιμή.

void select_info(char *choice_info);

Δέχεται εισαγωγή χαρακτήρα από τον χρήστη σχετικά με την άδεια χρήσης και μετά τον απαραίτητο έλεγχο, αποθηκεύει την επιλογή του. Η συνάρτηση δέχεται σαν παράμετρο έναν δείκτη στην επιλογή του χρήστη και δεν επιστρέφει τιμή.

void get_creds_login(char username[], unsigned long long int *id, short int *pin);

Παροτρύνει τον χρήστη να εισάγει τα στοιχεία του, ώστε να συνδεθεί στον λογαριασμό του και μετά τον απαραίτητο έλεγχο, τα αποθηκεύει. Η συνάρτηση δέχεται σαν παραμέτρους δεικτη στο όνομα χρήστη, δείκτη στον αριθμό λογαριασμού και δείκτη στον προσωπικό αριθμό ταυτοποίησης του χρήστη, ενώ δεν επιστρέφει τιμή.

void get_creds_register(char name[], char surname[], char username[], short int *pin, unsigned int *tin);

Παροτρύνει τον χρήστη να εισάγει τα στοιχεία του, ώστε να δημιουργήσει εναν καινούργιο λογαριασμό και μετά τον απαραίτητο έλεγχο, τα αποθηκεύει. Η συνάρτηση δεχεται σαν παραμέτρους δεικτη στο όνομα, δεικτη στο επίθετο, δεικτη στο όνομα χρήστη, δείκτη στον προσωπικό αριθμό ταυτοποίησης και δεικτη στο ΑΦΜ του χρήστη, ενώ δεν επιστρέφει τιμή.

void select_mode(char *choice_d_w_t_b_c_h);

Παροτρύνει τον χρήστη να επιλέξει αν θέλει να κάνει κατάθεση, ανάληψη ή μεταφορά χρημάτων ή αν θέλει να ελέγξει το χρηματικό υπόλοιπο του λογαριασμού του και μετά τον απαραίτητο έλεγχο, αποθηκεύει την επιλογή του. Η συνάρτηση δέχεται σαν παράμετρο δείκτη στην επιλογή του χρήστη και δεν επιστρέφει τιμή.

void get_amount_description(float *amount, char description[]);

Παροτρύνει τον χρήστη να εισάγει την ποσότητα των χρημάτων που αφορούν τη συναλλαγή και μετά τον απαραίτητο έλεγχο, την αποθηκεύει. Επίσης, παροτρύνει τον χρήστη να εισάγει μία περιγραφή για την συναλλαγή και μετά τον απαραίτητο έλεγχο, την αποθηκεύει. Η συνάρτηση δέχεται σαν παραμέτρους δείκτη στην ποσότητα των χρημάτων και δείκτη στην περιγραφή της συναλλαγής, ενώ δεν επιστρέφει τιμή.

void get_user_to_transfer(unsigned long long int *iban);

Παροτρύνει τον χρήστη να εισάγει το IBAN του χρήστη στον οποίο θέλει να μεταφέρει χρήματα και μετά τον απαραίτητο έλεγχο, το αποθηκεύει. Η συνάρτηση δέχεται σαν παράμετρο δείκτη στο IBAN του χρήστη στον οποίο θα μεταφερθούν τα χρήματα και δεν επιστρέφει τιμή.

void print_balance(float balance);

Τυπώνει στην οθόνη του χρήστη το χρηματικό υπόλοιπο του λογαριασμού του. Η συνάρτηση δέχεται σαν παράμετρο το χρηματικό υπόλοιπο του χρήστη και δεν επιστρέφει τιμή.

**void print_creds(char name[], char surname[], char username[], short int pin,
unsigned int tin, unsigned long long int iban,
unsigned long long int id);**

Τυπώνει στην οθόνη του χρήστη στοιχισμένα όλα του τα στοιχεία. Η συνάρτηση δέχεται σαν παράμετρο δείκτη στο ονομα, δείκτη στο επίθετο, δείκτη στο όνομα χρήστη, τον προσωπικό αριθμό ταυτοποίησης, το ΑΦΜ, το IBAN και τον αριθμό λογαριασμού του χρήστη, ενώ δεν επιστρέφει τιμή.

void print_deposits_history(struct Deposit deposits_history[]);

Τυπώνει στην οθόνη του χρήστη στοιχισμένα το ιστορικό των καταθέσεων του. Η συνάρτηση δέχεται σαν παράμετρο έναν πίνακα τύπου "Deposit", ο οποίος περιέχει τα στοιχεία κάθε κατάθεσης του χρήστη και δεν επιστρέφει τιμή.

void print_withdrawals_history(struct Withdrawal withdrawals_history[]);

Τυπώνει στην οθόνη του χρήστη στοιχισμένα το ιστορικό των αναλήψεων του. Η συνάρτηση δέχεται σαν παράμετρο έναν πίνακα τύπου "Withdrawal", ο οποίος περιέχει τα στοιχεία κάθε ανάληψης του χρήστη και δεν επιστρέφει τιμή.

void print_transfers_rec_history(struct Transfer transfers_rec_history[]);

Τυπώνει στην οθόνη του χρήστη στοιχισμένα το ιστορικό των μεταφορών που έχει λάβει. Η συνάρτηση δέχεται σαν παράμετρο έναν πίνακα τύπου "Transfer", ο οποίος περιέχει τα στοιχεία κάθε μεταφοράς που έχει λάβει και δεν επιστρέφει τιμή.

void print_transfers_send_history(struct Transfer transfers_send_history[]);

Τυπώνει στην οθόνη του χρήστη στοιχισμένα το ιστορικό των μεταφορών που έχει αποστείλει. Η συνάρτηση δέχεται σαν παράμετρο έναν πίνακα τύπου "Transfer", ο οποίος περιέχει τα στοιχεία κάθε μεταφοράς που έχει αποστείλει και δεν επιστρέφει τιμή.

void string_is_number(char input[], short int *only_digits);

Ελέγχει αν στην εισαγωγή περιέχεται τουλάχιστον ένα γράμμα, οπότε δεν είναι αριθμός. Η συνάρτηση δέχεται σαν παραμέτρους δείκτη στην εισαγωγή και δείκτη στη μεταβλητή only_digits, ενώ δεν επιστρέφει τιμή.

void string_is_float_number(char input[], short int *only_float);

Ελέγχει αν στην εισαγωγή περιέχεται παραπάνω από μία υποδιαστολή ή αν περιέχονται γράμματα, οπότε δεν είναι αριθμός κινητής υποδιαστολής. Η συνάρτηση δέχεται σαν παραμέτρους δείκτη στην εισαγωγή και δείκτη στη μεταβλητή only_float, ενώ δεν επιστρέφει τιμή.

void string_contains_digits(char input[], short int *contains_digits);

Ελέγχει αν στην εισαγωγή περιέχεται τουλάχιστον ένα ψηφίο, οπότε δεν είναι λέξη. Η συνάρτηση δέχεται σαν παραμέτρους δείκτη στην εισαγωγή και δείκτη στη μεταβλητή contains_digits, ενώ δεν επιστρέφει τιμή.

void string_is_one_word(char input[], short int *only_word);

Ελέγχει αν στην εισαγωγή περιέχεται τουλάχιστον ένας χαρακτήρας κενού, οπότε αποτελείται από παραπάνω από μία λέξεις. Η συνάρτηση δέχεται σαν παραμέτρους δείκτη στην εισαγωγή και δείκτη στη μεταβλητή only_word, ενώ δεν επιστρέφει τιμή.

short int terminate_transaction(char input[]);

Ελέγχει αν η εισαγωγή είναι ο συνδυασμός εξόδου '%%%' και τυπώνει το αντίστοιχο μήνυμα. Η συνάρτηση δέχεται σαν παράμετρο δείκτη στην εισαγωγή και επιστρέφει 1, αν η εισαγωγή ήταν ο συνδυασμός εξόδου, ενώ σε διαφορετική περίπτωση επιστρέφει 0.

void SetColor(int ForgC);

Αλλάζει το χρώμα του κειμένου στην ροή stdout. Η συνάρτηση δέχεται σαν παράμετρο τον κωδικό του χρώματος και δεν επιστρέφει τιμή.

void print_message_slowly(char message[]);

Τυπώνει γράμμα προς γράμμα μια συμβολοσειρά, με μία μικρή καθυστέρηση ανάμεσα στηντύπωση των γραμμάτων. Η συνάρτηση δέχεται σαν παράμετρο δείκτη στη συμβολοσειρά και δεν επιστρέφει τιμή.

short int cooldown(short int *invalid_input);

Ελέγχει αν η παράμετρός της είναι ίση ή μεγαλύτερη του τρια, οπότε ο χρήστης έχει κάνει τρεις ή παραπάνω αποτυχημένες προσπάθειες σύνδεσης και του απαγορεύει να κάνει νέα προσπάθεια για 10

δευτερόλεπτα. Η συνάρτηση δέχεται σαν παράμετρο δείκτη στη μεταβλητή `invalid_input` και επιστρέφει 1, αν ο χρήστης έκανε τρεις ή περισσότερες αποτυχημένες προσπάθειες, ενώ σε διαφορετική περίπτωση επιστρέφει 0.

`void error_sound(void);`

Παράγει ήχο αποτυχίας. Η συνάρτηση δεν δέχεται παραμέτρους και δεν επιστρέφει τιμή.

`void success_sound(void);`

Παράγει ήχο επιτυχίας. Η συνάρτηση δεν δέχεται παραμέτρους και δεν επιστρέφει τιμή.

II. File Access

`void check_files(void);`

Ελέγχει αν υπάρχουν καθένα από τα αρχεία `data`, `deposits`, `withdrawals` και `transfers`. Αν δεν υπάρχουν, τα δημιουργεί. Αν αποτύχει, τυπώνει το αντίστοιχο μήνυμα σφάλματος και τερματίζει το πρόγραμμα. Η συνάρτηση δεν δέχεται παραμέτρους και δεν επιστρέφει τιμή.

`struct User get_user_from_username(char *username);`

Διαβάζει το αρχείο `"data.bin"` έως ότου εντοπίσει τον χρήστη ο οποίος έχει όνομα χρήστη την παράμετρο. Η συνάρτηση δέχεται σαν παράμετρο δείκτη στο όνομα χρήστη και επιστρέφει την δομή τύπου `User`, `users`, η οποία περιέχει τα στοιχεία του χρήστη.

`struct User get_user_from_id(unsigned long long id);`

Διαβάζει το αρχείο `"data.bin"` έως ότου εντοπίσει τον χρήστη ο οποίος έχει αριθμό λογαριασμού την παράμετρο. Η συνάρτηση δέχεται σαν παράμετρο τον αριθμό λογαριασμού και επιστρέφει την δομή τύπου `User`, `users`, η οποία περιέχει τα στοιχεία του χρήστη.

`struct User get_user_from_iban(unsigned long long int iban);`

Διαβάζει το αρχείο `"data.bin"` έως ότου εντοπίσει τον χρήστη ο οποίος έχει IBAN την παράμετρο. Η συνάρτηση δέχεται σαν παράμετρο το IBAN και επιστρέφει την δομή τύπου `User`, `users`, η οποία περιέχει τα στοιχεία του χρήστη.

`void load_deposits_history(unsigned long long int id);`

Διαβάζει το αρχείο `"deposits.bin"` έως ότου εντοπίσει κατάθεση από τον χρήστη με αριθμό λογαριασμού αυτόν της παραμέτρου. Τότε, αποθηκεύει τα στοιχεία της συγκεκριμένης κατάθεσης στον πίνακα `"deposits_history"` και επαναλαμβάνει τη διαδικασία μέχρι όλες οι καταθέσεις του εκάστοτε χρήστη να έχουν αποθηκευτεί στον πίνακα `"deposits_history"`. Η συνάρτηση δέχεται σαν παράμετρο τον προσωπικό αριθμό λογαριασμού του χρήστη και δεν επιστρέφει τιμή.

void load_withdrawals_history(unsigned long long int id);

Διαβάζει το αρχείο "withdrawals.bin" έως ότου εντοπίσει κατάθεση από τον χρήστη με αριθμό λογαριασμού αυτόν της παραμέτρου. Τότε, αποθηκεύει τα στοιχεία της συγκεκριμένης ανάληψης στον πίνακα "withdrawals_history" και επαναλαμβάνει τη διαδικασία μέχρι όλες οι αναλήψεις του εκάστοτε χρήστη να έχουν αποθηκευτεί στον πίνακα "withdrawals_history". Η συνάρτηση δέχεται σαν παράμετρο τον προσωπικό αριθμό λογαριασμού του χρήστη και δεν επιστρέφει τιμή.

void load_transfers_rec_history(unsigned long long int);

Διαβάζει το αρχείο "transfers.bin" έως ότου εντοπίσει μεταφορά από τον χρήστη με αριθμό λογαριασμού αυτόν της παραμέτρου. Τότε, αποθηκεύει τα στοιχεία της συγκεκριμένης μεταφοράς στον πίνακα "transfers_rec_history" και επαναλαμβάνει τη διαδικασία μέχρι όλες οι μεταφορές από τον εκάστοτε χρήστη να έχουν αποθηκευτεί στον πίνακα "transfers_rec_history". Η συνάρτηση δέχεται σαν παράμετρο τον προσωπικό αριθμό λογαριασμού του χρήστη και δεν επιστρέφει τιμή.

void load_transfers_send_history(unsigned long long int id);

Διαβάζει το αρχείο "transfers.bin" έως ότου εντοπίσει μεταφορά προς τον χρήστη με αριθμό λογαριασμού αυτόν της παραμέτρου. Τότε, αποθηκεύει τα στοιχεία της συγκεκριμένης μεταφοράς στον πίνακα "transfers_rec_history" και επαναλαμβάνει τη διαδικασία μέχρι όλες οι μεταφορές προς τον εκάστοτε χρήστη να έχουν αποθηκευτεί στον πίνακα "transfers_rec_history". Η συνάρτηση δέχεται σαν παράμετρο τον προσωπικό αριθμό λογαριασμού του χρήστη και δεν επιστρέφει τιμή.

void random_id(unsigned long long int *id);

Δημιουργεί έναν τυχαίο αριθμό λογαριασμού για κάθε νέο χρήστη, αξιοποιώντας την ώρα ως seed στη συνάρτηση srand(). Η συνάρτηση δέχεται σαν παράμετρο δείκτη στον αριθμό λογαριασμού και δεν επιστρέφει τιμή.

void random_iban(unsigned long long int *iban);

Δημιουργεί ένα τυχαίο iban για κάθε νέο χρήστη, αξιοποιώντας την ώρα ως seed στη συνάρτηση srand(). Η συνάρτηση δέχεται σαν παράμετρο δείκτη στο iban και δεν επιστρέφει τιμή.

unsigned long long int create_new_user(char *name, char *surname, char *username, short int pin, unsigned int tin);

Διαβάζει το αρχείο "data.bin" έως ότου εντοπίσει ένα κενό στοιχείο στη συστοιχία users των δομών τύπου User. Στη συνέχεια, δημιουργεί σε αυτό το στοιχείο έναν καινούργιο χρήστη, σύμφωνα με τις αντίστοιχες παραμέτρους, και αποθηκεύει τα στοιχεία του στο αρχείο "data.bin". Η συνάρτηση δέχεται σαν παραμέτρους το όνομα, το επώνυμο, το όνομα χρήστη, τον προσωπικό αριθμό ταυτοποίησης και το ΑΦΜ του χρήστη, ενώ επιστρέφει τον αριθμό λογαριασμού του χρήστη.

void new_deposit(unsigned long long int id, float amount, char *description);

Διαβάζει το αρχείο "deposits.bin" έως ότου εντοπίσει τον χρήστη ο οποίος έχει αριθμό λογαριασμού την αντίστοιχη παράμετρο. Στη συνέχεια, καταθέτει ποσότητα χρημάτων στον λογαριασμό αυτό και

προσθέτει μία περιγραφή για τη συναλλαγή. Τα στοιχεία της κατάθεσης αποθηκεύονται στο αρχείο "deposits.bin". Η συνάρτηση δέχεται σαν παραμέτρους τον αριθμό λογαριασμού, την ποσότητα των χρημάτων και την περιγραφή, ενώ δεν επιστρέφει τιμή.

void new_withdrawal(unsigned long long int id, float amount, char *description);

Διαβάζει το αρχείο "withdrawals.bin" έως ότου εντοπίσει τον χρήστη ο οποίος έχει αριθμό λογαριασμού την αντίστοιχη παράμετρο. Στη συνέχεια, αναλήπτει ποσότητα χρημάτων από τον λογαριασμό αυτό και προσθέτει μία περιγραφή για τη συναλλαγή. Τα στοιχεία της ανάληψης αποθηκεύονται στο αρχείο "withdrawals.bin". Η συνάρτηση δέχεται σαν παραμέτρους τον αριθμό λογαριασμού, την ποσότητα των χρημάτων και την περιγραφή, ενώ δεν επιστρέφει τιμή.

void new_transfer(unsigned long long int id_sender, unsigned long long int id_receiver, float amount, char *description);

Διαβάζει το αρχείο "transfers.bin" έως ότου εντοπίσει τον χρήστη ο οποίος έχει αριθμό λογαριασμού την αντίστοιχη παράμετρο του αποστολέα. Στη συνέχεια, μεταφέρει ποσότητα χρημάτων από τον λογαριασμό του αποστολέα στον λογαριασμό του παραλήπτη και προσθέτει μία περιγραφή για τη συναλλαγή. Τα στοιχεία της ανάληψης αποθηκεύονται στο αρχείο "transfers.bin". Η συνάρτηση δέχεται σαν παραμέτρους τον αριθμό λογαριασμού του αποστολέα, τον αριθμό λογαριασμού του παραλήπτη, την ποσότητα των χρημάτων και την περιγραφή, ενώ δεν επιστρέφει τιμή.

Αναφορές

- [How to change text color and console color in code::blocks?](#)
- [What are the most common naming conventions in C?](#)
- [What does %\[^\n\] mean in C?](#)
- [atoi vs atol vs strtol vs strtoul vs sscanf](#)
- [How to prevent multiple definitions in C?](#)
- [GNU General Public License](#)
- [Data Types in C](#)
- [Floating Point Arithmetic](#)
- [GCC C Compiler](#)
- C για Επιστήμονες και Μηχανικούς - Harry H. Cheng: Κεφάλαιο 14, Πρόγραμμα 14.17
- Διαφάνειες διαλέξεων - Μάθημα: Δομημένος Προγραμματισμός (004), Διδάκτορας: Σεβαστιάδης Χρήστος, Πανεπιστήμιο: Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Άδεια δικαιωμάτων για ανοιχτό ελεύθερο κώδικα

Το εν λόγω λογισμικό (software), το εκτελέσιμο αρχείο και ο πηγαίος κώδικας αυτού, διατίθεται ως ανοιχτός, ελεύθερος κώδικας σύμφωνα με τους όρους που προβλέπονται από την άδεια GNU General Public License version 3. Ο διαμοιρασμός του λογισμικού αυτού θα πρέπει να πραγματοποιείται σύμφωνα με τους όρους αυτούς. Δεν παρέχεται καμία απολύτως εγγύηση σχετικά με τη λειτουργία ή μη του κώδικα. Παρακαλώ ανατρέξτε στους αναλυτικούς όρους και προϋποθέσεις της άδειας στο αρχείο κειμένου “LICENSE.txt” που συνοδεύει το σύνολο των αρχείων του πηγαίου κώδικα.