

ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ
ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ
CoreBank Project

ΕΥΑΓΓΕΛΟΠΟΥΛΟΣ ΓΕΩΡΓΙΟΣ ΑΕΜ:23079

ΠΑΠΑΣΤΕΡΓΙΟΥ ΙΩΑΝΝΗΣ ΑΕΜ:23136

ΥΠΕΥΘΥΝΟΣ ΚΑΘΗΓΗΤΗΣ: ΚΕΚΑΤΟΣ ΝΙΚΟΛΑΟΣ



- Εισαγωγή & Motivation

- Τι αφορά το project;

Το παρόν project αφορά την ανάπτυξη ενός τραπεζικού συστήματος σε C++, το οποίο λειτουργεί αποκλειστικά μέσα από το τερματικό. Μέσω του συστήματος, ο χρήστης μπορεί να δημιουργήσει τραπεζικούς λογαριασμούς, να διαχειριστεί το υπόλοιπό του (καταθέσεις, αναλήψεις, μεταφορές), να επεξεργαστεί τα προσωπικά του στοιχεία και να προστατεύσει την πρόσβαση με PIN.

- Ποιο πρόβλημα λύνει;

Το πρόβλημα που έρχεται να λύσει είναι η απλή, γρήγορη και ασφαλής διαχείριση τραπεζικών συναλλαγών χωρίς τη χρήση βάσεων δεδομένων ή σύνδεσης στο

διαδίκτυο. Ουσιαστικά, προσομοιώνει τις βασικές λειτουργίες ενός ATM ή e-banking, σε τοπικό περιβάλλον.

- Γιατί επιλέξατε αυτό το αντικείμενο;

Είναι πρακτικό και βασισμένο σε πραγματικό σενάριο που ο κάθε χρήστης μπορεί να κατανοήσει.

Μας επέτρεψε να εξασκηθούμε ουσιαστικά σε έννοιες της C++ όπως αρχεία, συναρτήσεις, ροή προγράμματος και διαχείριση καταστάσεων.

-Objective & Scope

- Ποιοι είναι οι βασικοί στόχοι;

- Ανάπτυξη δομημένου κώδικα σε C++ με καλή αρχιτεκτονική και αναγνώσιμη μορφή.
- Υλοποίηση πλήρους ροής χρήστη: από τη δημιουργία λογαριασμού μέχρι τη διαγραφή του.
- Προσθήκη προστασίας μέσω PIN, ώστε κάθε λογαριασμός να έχει στοιχειώδη ασφάλεια.
- Καταγραφή όλων των συναλλαγών με ημερομηνία και ώρα (timestamp) για ιστορικό και παρακολούθηση.

- Ποιες δυνατότητες περιλαμβάνονται;

- Δημιουργία λογαριασμού με όνομα, ID, PIN και αρχικό υπόλοιπο
- Κατάθεση / Ανάληψη ποσών, με ενημέρωση του υπολοίπου και καταγραφή της συναλλαγής.
- Μεταφορά χρημάτων από έναν λογαριασμό σε άλλον, με αντίστοιχες εγγραφές και στα δύο αρχεία.
- Προβολή λογαριασμού (σε δομημένη μορφή από το .txt αρχείο).
- Επεξεργασία στοιχείων (όνομα και PIN) με έλεγχο ταυτότητας.
- Διαγραφή λογαριασμού.
- Αποθήκευση όλων των δεδομένων σε αρχεία .txt με εύκολη προσβασιμότητα.

Το σύστημα είναι επεκτάσιμο, ασφαλές σε βασικό επίπεδο, και έχει σχεδιαστεί με γνώμονα την κατανόηση και την ευχρηστία — και από τον προγραμματιστή και από τον χρήστη.

3. System Architecture

- Πώς λειτουργεί το σύστημα;

Το σύστημα λειτουργεί πλήρως μέσω του τερματικού (console), όπου ο χρήστης καλείται να αλληλεπιδράσει με ένα απλό αριθμητικό μενού επιλογών. Κάθε ενέργεια ζητά τα κατάλληλα στοιχεία (όπως ID και PIN) και κατόπιν εκτελεί την αντίστοιχη λειτουργία.

Όλες οι πληροφορίες των λογαριασμών αποθηκεύονται σε αρχεία .txt τοπικά στον φάκελο του project. Η πρόσβαση στους λογαριασμούς προστατεύεται με PIN, το οποίο αποθηκεύεται στο ίδιο αρχείο για απλότητα.

Η ροή είναι απόλυτα γραμμική και ελεγχόμενη, ιδανική για αρχάριους προγραμματιστές αλλά με αρκετή “δύναμη” ώστε να προσομοιώνει μια τράπεζα.

- Ποιες είναι οι βασικές του συνιστώσες;

Συνιστώσα	Ρόλος
main()	Εκκινεί το μενού και συνδέει τις ενέργειες
createAccount()	Δημιουργεί νέο αρχείο .txt με τα στοιχεία του λογαριασμού
verifyPIN()	Ελέγχει την εγκυρότητα του PIN που δίνει ο χρήστης
deposit() / withdraw() / transfer()	Πραγματοποιούν συναλλαγές και ενημερώνουν το υπόλοιπο
viewAccount()	Διαβάζει και εμφανίζει το περιεχόμενο του .txt
editAccount()	Επιτρέπει αλλαγή ονόματος και PIN
getBalance(), updateBalance()	Εργαλεία ανάγνωσης και τροποποίησης του υπολοίπου

- Διάγραμμα ροής



- Τεχνολογίες που Χρησιμοποιήθηκαν

- Ποιες και γιατί;

Το σύστημα αναπτύχθηκε εξ ολοκλήρου σε **C++**, αξιοποιώντας τις βασικές αλλά και πιο προχωρημένες δυνατότητες της γλώσσας, όπως η διαχείριση αρχείων, οι συναρτήσεις, και η αλληλεπίδραση με το περιβάλλον του χρήστη.

- C++ – Επιλέχθηκε λόγω της ταχύτητας, της ευελιξίας και της δυνατότητας να δουλεύουμε κοντά στο hardware, κάτι που ταιριάζει με το concept ενός "τοπικού τραπεζικού συστήματος".
- Dev-C++ ή Code::Blocks – για compilation και δοκιμή.
-

-Αναφορά βιβλιοθηκών που χρησιμοποιήθηκαν

Βιβλιοθήκη	Χρήση
<code>#include <iostream></code>	Είσοδος/Εξοδος στην κονσόλα (cin, cout)
<code>#include <fstream></code>	Διαχείριση αρχείων .txt για τους λογαριασμούς
<code>#include <string></code>	Χρήση συμβολοσειρών για ονόματα, PIN κ.λπ.
<code>#include <ctime></code>	Για την καταγραφή ημερομηνίας/ώρας (timestamp)
<code>#include <iomanip></code>	Για μορφοποίηση αριθμών (π.χ. 2 δεκαδικά ψηφία στο υπόλοιπο)

#include <windows.h>	Για αλλαγή χρώματος στο κείμενο της κονσόλας (SetConsoleTextAttribute)
----------------------	---

`#include <sstream>` Για μετατροπή αριθμών σε strings και αντίστροφα

Αυτές οι βιβλιοθήκες επέτρεψαν στο project να έχει:

- Καθαρό τερματικό με έγχρωμες εξόδους
- Σωστά φορμαρισμένα οικονομικά δεδομένα
- Διαχείριση λογαριασμών και ιστορικού σε τοπικά αρχεία

- Κώδικας & Υλοποίηση

- Πώς υλοποιήθηκε η βασική λειτουργικότητα;

Η υλοποίηση του CoreBank έγινε εξ ολοκλήρου σε C++, με δομημένο και modular τρόπο, ώστε κάθε λειτουργία να είναι εύκολα επεκτάσιμη ή τροποποιήσιμη. Κάθε λειτουργικό στοιχείο του συστήματος (π.χ. κατάθεση,

δημιουργία λογαριασμού, έλεγχος PIN) βρίσκεται σε δική του ξεχωριστή συνάρτηση.

- Το μενού εμφανίζεται μέσω της `main()`, όπου ο χρήστης εισάγει αριθμό επιλογής.
- Οι λειτουργίες όπως `createAccount()`, `editAccount()`, `deposit()`, `withdraw()` κ.ά., καλούνται μέσω `switch-case` δομής.
- Η κάθε ενέργεια βασίζεται σε αλληλεπίδραση με τοπικά `.txt` αρχεία που περιέχουν τα στοιχεία κάθε λογαριασμού.

- Υπάρχουν σημαντικοί αλγόριθμοι ή δομές δεδομένων;

Η βασική δομή δεδομένων είναι τα αρχεία `.txt` — δεν χρησιμοποιούνται `arrays`, `structs` ή `classes`, καθώς από το `project` θέλαμε απλή και κατανοητή διαχείριση.

Ωστόσο, χρησιμοποιήθηκε:

- `ifstream / ofstream` για `file streams`.
- `stringstream` για επεξεργασία κειμένου (π.χ. αλλαγή συγκεκριμένων γραμμών στα αρχεία).
- `getline()` + `substr()` για ανάγνωση και ανάλυση γραμμών.
- `SetConsoleTextAttribute()` για προσθήκη χρωμάτων στο `output` του τερματικού.

Ο έλεγχος του PIN υλοποιείται με απλή ανάγνωση της γραμμής PIN: από το αρχείο και σύγκριση με την είσοδο του χρήστη — ένας πρακτικός αλλά κατανοητός αλγόριθμος για αυτή τη χρήση.

- Αποτελέσματα & Demo

Το σύστημα `CoreBank` υλοποιήθηκε με επιτυχία, επιτυγχάνοντας όλους τους βασικούς του στόχους. Η λειτουργικότητα είναι πλήρης, σταθερή και απόλυτα κατανοητή προς το χρήστη .

Τι πετύχαμε:

- Δημιουργία και επεξεργασία λογαριασμών
- Καταθέσεις, αναλήψεις και μεταφορές με ενημερωμένο υπόλοιπο
- Έλεγχος πρόσβασης μέσω PIN
- Καταγραφή όλων των συναλλαγών με ημερομηνία και ώρα
- Προβολή όλων των στοιχείων με καθαρή μορφοποίηση
- Χρήση χρωμάτων στο τερματικό για βελτιωμένη εμπειρία χρήστη

Demo Screenshots

- Ξεκινάμε με την εισαγωγή στο terminal:

```
=====
Welcome to CoreBank
=====

==== CoreBank Menu ====
1. Create
2. Delete
3. Deposit
4. Withdraw
5. Transfer
6. View
7. Edit
0. Exit
Choice: |
```

-Δημιουργία λογαριασμού:

```
Welcome to CoreBank
=====

==== CoreBank Menu ====
1. Create
2. Delete
3. Deposit
4. Withdraw
5. Transfer
6. View
7. Edit
0. Exit
Choice: 1
Enter Account ID: 123
Enter Full Name: Giannhs Papastergiou
Enter 4-digit PIN: 7890
Initial Deposit: 1000
Account created successfully!
```

-Προσθήκη χρημάτων:

```
==== CoreBank Menu ====
1. Create
2. Delete
3. Deposit
4. Withdraw
5. Transfer
6. View
7. Edit
0. Exit
Choice: 3
Enter ID: 123
Enter PIN: 7890
Amount to deposit: 200
Deposit successful.
```

-Ανάλυση χρημάτων:

```
==== CoreBank Menu ====
1. Create
2. Delete
3. Deposit
4. Withdraw
5. Transfer
6. View
7. Edit
0. Exit
Choice: 4
Enter ID: 123
Enter PIN: 7890
Amount to withdraw: 110
Withdrawal successful.
```

-Δημιουργία 2^{ου} λογαριασμού για να δούμε και την επιλογή transfer:

```
==== CoreBank Menu ====
1. Create
2. Delete
3. Deposit
4. Withdraw
5. Transfer
6. View
7. Edit
0. Exit
Choice: 1
Enter Account ID: 234
Enter Full Name: George Euaggelopoulos
Enter 4-digit PIN: 1234
Initial Deposit: 1000
Account created successfully!
```

-Τα στοιχεία ενός από τους 2 λογαριασμούς για να δούμε και την επιλογή 6:

```
==== CoreBank Menu ====
1. Create
2. Delete
3. Deposit
4. Withdraw
5. Transfer
6. View
7. Edit
0. Exit
Choice: 6
Enter ID to view: 123
===== ACCOUNT INFO =====
Name: Giannhs Papastergiou
ID: 123
Balance: 690.00
PIN: 7890
Created: 05/20/25 16:50:05
=====

[Deposit] +$200 at 05/20/25 16:52:15
[Withdraw] -$110 at 05/20/25 16:53:29
[Transfer OUT] -$400 to ID 234 at 05/20/25 16:56:04
```

-Το .txt αρχείο που δημιουργήθηκε στην αρχή και οι αλλαγές που έγιναν στο εσωτερικό του 1^{ου} λογαριασμού, όσο δοκιμάζαμε τις επιλογές του προγράμματος :

```
===== ACCOUNT INFO =====  
Name: Giannhs Papastergiou  
ID: 123  
Balance: 690.00  
PIN: 7890  
Created: 05/20/25 16:50:05  
=====
```



```
[Deposit] +$200 at 05/20/25 16:52:15  
[Withdraw] -$110 at 05/20/25 16:53:29  
[Transfer OUT] -$400 to ID 234 at 05/20/25 16:56:04
```

-Εδώ και το εσωτερικό του 2^{ου} λογαριασμού:

```
===== ACCOUNT INFO =====  
Name: George Euaggelopoulos  
ID: 234  
Balance: 1400.00  
PIN: 1234  
Created: 05/20/25 16:55:07  
=====
```



```
[Transfer IN] +$400 from ID 123 at 05/20/25 16:56:04
```

-Και εδω θα δούμε κάποια errors που εμφανίζει το πρόγραμμα στο χρήστη ανάλογα με τις επιλογές που έχει αποφασίσει μέσω του πληκτρολογίου

```
==== CoreBank Menu ====
```

- 1. Create
- 2. Delete
- 3. Deposit
- 4. Withdraw
- 5. Transfer
- 6. View
- 7. Edit
- 0. Exit

Choice: 6

Enter ID to view: 456

Not found.

```
==== CoreBank Menu ====
```

- 1. Create
- 2. Delete
- 3. Deposit
- 4. Withdraw
- 5. Transfer
- 6. View
- 7. Edit
- 0. Exit

Choice: 1

Enter Account ID: 123

Account already exists!

```
==== CoreBank Menu ====
```

- 1. Create
- 2. Delete
- 3. Deposit
- 4. Withdraw
- 5. Transfer
- 6. View
- 7. Edit
- 0. Exit

Choice: 4

Enter ID: 123

Enter PIN: 7890

Amount to withdraw: 2000

Insufficient funds.

```
==== CoreBank Menu ====
```

- 1. Create
- 2. Delete
- 3. Deposit
- 4. Withdraw
- 5. Transfer
- 6. View
- 7. Edit
- 0. Exit

Choice: 5

From ID: 234

Enter PIN: 1234

To ID: 123

Amount: 10000

Insufficient funds.

-Και την διαγραφή ενός λογαριασμού:

```
==== CoreBank Menu ====
```

- 1. Create
- 2. Delete
- 3. Deposit
- 4. Withdraw
- 5. Transfer
- 6. View
- 7. Edit
- 0. Exit

Choice: 2

Enter Account ID: 123

Enter PIN: 7890

Account deleted.

- Comparison with AI generated code

Κώδικας AI	CoreBank (τελικό αποτέλεσμα)
Σχεδόν λειτουργικός με κάποια μικρά σφάλματα, αλλά γενικός	Εξατομικευμένος, 100% ελεγμένος
Συνήθως χωρίς validation	Έλεγχοι PIN, ID, σφάλματα
Δεν γράφει έγγραφα, οδηγίες	Δημιουργήθηκαν με δική μας κατανόηση, και ιδέες
Δεν εστιάζει σε UX	Χρώματα, μηνύματα, ασφάλεια

Συμπερασματικά:

Το AI δεν αντικατέστησε την ανθρώπινη κατανόηση — την υποστήριξε. . Το τελικό αποτέλεσμα είναι προϊόν συνεργασίας ανθρώπου και τεχνητής νοημοσύνης , και αυτό είναι που το κάνει μοναδικό.

-Conclusions & Lessons learned

Η ανάπτυξη του CoreBank αποτέλεσε μια πλήρη και ουσιαστική εμπειρία, τόσο από τεχνικής όσο και από μαθησιακής άποψης. Δεν επρόκειτο απλώς για συγγραφή κώδικα, αλλά για σχεδιασμό, οργάνωση και εφαρμογή ενός ολοκληρωμένου συστήματος που λειτουργεί όπως ένας πραγματικός οργανισμός.

Τι μάθαμε στην πράξη:

- Πώς να οργανώνουμε μεγάλα προβλήματα σε μικρές συναρτήσεις.
- Πώς να χρησιμοποιούμε τα αρχεία .txt σαν βάση δεδομένων.
- Πώς να ελέγχουμε καταστάσεις, να διαχειριζόμαστε είσοδο/έξοδο, και να προστατεύουμε δεδομένα.
- Πόσο σημαντικό είναι το User Experience(UX) , ακόμα και σε ένα απλό terminal πρόγραμμα.

Επιπλέον, μάθαμε πώς να αξιοποιούμε έξυπνα την τεχνητή νοημοσύνη χωρίς να χάνουμε την προσωπική κατανόηση και δημιουργικότητα.

Τι θα μπορούσε να προστεθεί στο μέλλον:

- Διασύνδεση με γραφικό περιβάλλον (GUI) για πιο οπτική εμπειρία
- Σύνδεση με βάση δεδομένων
- Υποστήριξη πολλών χρηστών ταυτόχρονα ή login με session