

LECTURE EAGLE 3: ΤΡΑΠΕΖΙΚΟΙ ΛΟΓΑΡΙΑΣΜΟΙ (BANK ACCOUNTS)



Προθεσμία Υποβολής	10 Απριλίου 2022
Ώρα	23:59
Πλατφόρμα	http://eagle.csd.auth.gr
Λογαριασμοί eagle	To username είναι το AEM και password το email (χωρίς το @csd.auth.gr)

Περιγραφή Άσκησης

Σκοπός της άσκησης είναι να αναπτυχθεί ένα απλό πρόγραμμα διαχείρισης του υπολοίπου ενός λογαριασμού σε ένα τραπεζικό σύστημα. Βασικές λειτουργίες που πρέπει να υποστηρίζονται είναι η ανάληψη (withdraw) και η κατάθεση (deposit) κάποιου ποσού.

Περισσότερες λεπτομέρειες για την υλοποίηση παρουσιάζονται παρακάτω.

Κλάσεις και Συναρτήσεις

Σας δίνετε η δήλωση της κλάσης Account (αρχείο [Account.h](#)), η οποία αναπαριστά έναν τραπεζικό λογαριασμό και προσφέρει τις μεταβλητές:

```
float balance;  
char *description;
```

για την αποθήκευση του υπολοίπου και μιας περιγραφής του λογαριασμού. Επίσης περιέχει τα εξής:

- `Account(char *description)` και `Account(char *description, float balance)` για την αρχικοποίηση αντικειμένων με παραμέτρους την περιγραφή και του αρχικού διαθέσιμου υπολοίπου.
- `bool withdraw(float amount)` για την ανάληψη του ποσού `amount`.
- `bool deposit(float amount)` για την κατάθεση του ποσού `amount`.
- Getters για το υπόλοιπο και την περιγραφή του λογαριασμού.

Με βάση την παραπάνω κλάση, θα πρέπει να υλοποιηθούν τα εξής:

- Η υλοποίηση της κλάσης **Account** (Account.cpp), με βάση την παραπάνω δήλωση (Account.h).
- Μια κλάση **CreditCard** (παράγωγη της **Account**) η οποία να αναπαριστά μια πιστωτική κάρτα. Η **CreditCard** θα πρέπει να έχει ένα πιστωτικό όριο **LIMIT** και να επιτρέπει υπεραναλήψεις (άρα θα εμφανίζεται αρνητικό υπόλοιπο) μέχρι του ποσού του **LIMIT**.
- Μια κλάση **Savings** (παράγωγη της **Account**) η οποία να αναπαριστά ένα λογαριασμό αποταμίευσης, ο οποίος θα επιτρέπει μέχρι 3 μόνο αναλήψεις ανεξαρτήτως ποσού (αρκεί βέβαια να επαρκεί το υπόλοιπο). Φυσικά στις καταθέσεις δεν υπάρχει όριο.

Οι κλάσεις θα πρέπει να λειτουργούν με βάση τη main (αρχείο [lecture_eagle_3_accounts.cpp](#)) στην οποία μπορείτε να δείτε ποιες μεθόδους θα πρέπει να περιέχει κάθε κλάση σας.

Σημειώσεις: Δεν πρέπει να υποβάλετε στο eagle τα αρχεία που σας δίνονται, δηλαδή τα Account.h και lecture_eagle_3_accounts.cpp. Χρειάζεται να υποβάλετε μόνο τα αρχεία του προγράμματος που θα αναπτύξετε εσείς για τις κλάσεις Account, CreditCard και Savings. Ο κώδικας θα πρέπει να είναι δικός σας (ο κώδικας θα ελεγχθεί). Μην χρησιμοποιείτε το eagle ως compiler: αναπτύξτε το πρόγραμμά σας σε κάποιον editor και υποβάλετε τον κώδικά σας αφού ελέγξετε ότι παράγονται τα σωστά αποτελέσματα.

Συνάρτηση main

Στην main δημιουργείται ένας πίνακας 3 θέσεων για την αποθήκευση τριών δεικτών τύπου Account και δημιουργούνται 3 αντικείμενα διαφορετικού τύπου:

```
Account *myAccounts[3];  
myAccounts[0] = new Account((char *)"simple", 100);  
myAccounts[1] = new CreditCard((char *)"visa", 100, 200);  
myAccounts[2] = new Savings((char *)"My savings", 100);
```

Στην συνέχεια, ο χρήστης επιλέγει αν θα κάνει κατάθεση (d), ανάληψη (w) ή να τερματίσει το πρόγραμμα (q). Στην περίπτωση της ανάληψης ή της κατάθεσης, ζητείται το ποσό και η ενέργεια εκτελείται και στους τρεις λογαριασμούς. Η διαδικασία συνεχίζεται, δηλαδή ο χρήστης επιλέγει αν θα κάνει κατάθεση ή ανάληψη μέχρι να επιλέξει τερματισμό, όπου και εκτυπώνονται τα υπόλοιπα των τριών λογαριασμών.

Ενδεικτικές Εκτελέσεις

```
> .\main.exe  
Quit, deposit or withdraw (q, d, w)?:  
d  
Give the amount:  
50  
Quit, deposit or withdraw (q, d, w)?:  
q
```

```
150
150
150
```

```
> .\main.exe
Quit, deposit or withdraw (q, d, w)?:
w
Give the amount:
10
Quit, deposit or withdraw (q, d, w)?:
w
Give the amount:
20
Quit, deposit or withdraw (q, d, w)?:
w
Give the amount:
70
Quit, deposit or withdraw (q, d, w)?:
q
0
0
0
```

```
> .\main.exe
Quit, deposit or withdraw (q, d, w)?:
w
Give the amount:
100
Quit, deposit or withdraw (q, d, w)?:
w
Give the amount:
1
Withdraw denied for simple
Withdraw denied for My savings
Quit, deposit or withdraw (q, d, w)?:
q
0
-1
0
```

```
> .\main.exe
Quit, deposit or withdraw (q, d, w)?:
w
Give the amount:
100
Quit, deposit or withdraw (q, d, w)?:
w
```

```
Give the amount:
1
Withdraw denied for simple
Withdraw denied for My savings
Quit, deposit or withdraw (q, d, w)?:
w
Give the amount:
400
Withdraw denied for simple
Withdraw denied for visa
Withdraw denied for My savings
Quit, deposit or withdraw (q, d, w)?:
q
0
-1
0
```

```
> .\main.exe
Quit, deposit or withdraw (q, d, w)?:
w
Give the amount:
1
Quit, deposit or withdraw (q, d, w)?:
w
Give the amount:
1
Quit, deposit or withdraw (q, d, w)?:
w
Give the amount:
1
Quit, deposit or withdraw (q, d, w)?:
w
Give the amount:
1
Withdraw denied for My savings
Quit, deposit or withdraw (q, d, w)?:
q
96
96
97
```