

Γλώσσα Προγραμματισμού C++

Άσκηση 2019

1. Δημιουργείτε μια κλάση Account ως base class των SavingsAccount και CheckingAccount. Μεταφέρετε σ' αυτήν όλα τα πεδία (μεταβλητές) και τις μεθόδους (συναρτήσεις) που είναι κοινά στις 2 υποκλάσεις. Στα αρχεία των υποκλάσεων (derived classes) δηλώστε πως κληρονομούν την Account (" : public Account "). Το vector στην κλάση Bank θα δέχεται ως elements, δείκτες σε αντικείμενα Account (`vector<Account*> v`)
2. Να αντικατασταθούν τα πεδία string Owner, contact κλπ της κλάσης Account από αντικείμενο της κλάσης Person που είχαμε φτιάξει παλιότερα. Προσοχή: όπου χρησιμοποιούνται οι μεταβλητές που καταργήθηκαν (πχ συνάρτηση show()) να διορθωθεί κατάλληλα ο κώδικας.
3. Όπως είπαμε, οι λογαριασμοί πρέπει να δημιουργούνται στην κλάση Bank και όχι στην main(). Γι αυτό το λόγο δημιουργείτε μια μέθοδο Bank : : createAccount που θα δέχεται ως ορίσματα ένα αντικείμενο τύπου Person, ένα string iban, double amount και ένα int type. Αν το type==1, θα δημιουργείται δυναμικά ένα SavingsAccount με τα στοιχεία που δόθηκαν ενώ αν type==2, ένα CheckingAccount. Παράδειγμα για την δυναμική δημιουργία αντικειμένου:
`Account *a = new SavingsAccount(person1, iban, amnt, 1);`
Μετά τη δημιουργία του object, ο pointer θα εισάγεται στο vector.
4. Οι withdraw(), deposit() θα πρέπει να γίνονται πλέον μέσω του αντικειμένου Bank, αφού στη main() δεν θέλουμε να χειριζόμαστε αντικείμενα Account. Γι αυτό το λόγο θα δημιουργήσουμε στην κλάση Bank τις αντίστοιχες συναρτήσεις. Παράδειγμα:
`void Bank::depositToAccount(string iban, double amount)`
Αυτή συνάρτηση θα καλεί πρώτα την find() ώστε να βρεί το index i, του λογαριασμού μέσα στο vector v και μετά θα εκτελεί την εντολή: `v[i] -> deposit(amount);`
(Προσοχή στο βελάκι -, αφού πλέον το v[i] είναι pointer και όχι αντικείμενο)
5. Δημιουργείτε στην κλάση Bank μια μέθοδο showAll(), χωρίς ορίσματα, που θα διατρέχει όλο το vector v και θα καλεί την show() του κάθε αντικειμένου.
6. Στη main():
Δημιουργείτε (είτε στατικά είτε δυναμικά) ένα αντικείμενο της κλάσης Bank.
Δημιουργείτε στατικά, 2 αντικείμενα Person.
Για κάθε person δημιουργείτε ένα CheckingAccount και ένα SavingsAccount. (καλώντας την createAccount() μέσω του αντικειμένου Bank).
Κάντε συνολικά τουλάχιστον 10 transactions στους λογαριασμούς που δημιουργήσατε.
Καλέστε την calculateCost()
Τέλος, καλέστε την showAll() που θα πρέπει να τυπώσει στην οθόνη τα στοιχεία όλων των λογαριασμών.