

# Αντικειμενοστραφής Προγραμματισμός με την C++

## Τμήμα Πληροφορικής και Τηλεπικοινωνιών, Πανεπιστήμιο Ιωαννίνων

---

### Ενδεικτικά θέματα για την πρόοδο (Γκόγκος Χρήστος)

---

#### Θέμα 1

Κατασκευάστε μια κλάση `account` (λογαριασμός τράπεζας) που να περιέχει τα ιδιωτικά δεδομένα: `owner` (όνομα ιδιοκτήτη), `balance` (υπόλοιπο) και `type` (τύπος λογαριασμού). Ο τύπος λογαριασμού να είναι `enum class` με όνομα `ACCOUNT_TYPE` και 3 πιθανές τιμές `CHECKING`, `SAVINGS` και `MONEY_MARKET`.

1. Κατασκευάστε `getters` και `setters` για τα ιδιωτικά μέλη δεδομένων της κλάσης `owner`.
2. Ορίστε μια στατική μεταβλητή με όνομα `nr_of_accounts` που μετρά πόσα αντικείμενα έχουν κατασκευαστεί.
3. Κατασκευάστε έναν `constructor` που εισάγει τις εξής τιμές: `ιδιοκτήτης=Noname`, `υπόλοιπο=0€`, `τύπος λογαριασμού=CHECKING`
4. Κατασκευάστε έναν `constructor` που να δέχεται 3 παραμέτρους και να αρχικοποιεί με αυτές και τα 3 πεδία του προς κατασκευή αντικειμένου.
5. Κατασκευάστε έναν `copy constructor` που να δημιουργεί ένα αντίγραφο του λογαριασμού αλλά με υπόλοιπο μηδέν.
6. Ορίστε τη συνάρτηση `deposit(double)` (κατάθεση) που προσθέτει στον λογαριασμό ένα ποσό.
7. Ορίστε τη συνάρτηση `withdrawal(double)` (ανάληψη) που αφαιρεί από τον λογαριασμό ένα ποσό. Αν το ποσό είναι μεγαλύτερο από το υπόλοιπο τότε το υπόλοιπο να μηδενίζεται.
8. Υπερφορτώστε τον τελεστή `<` έτσι ώστε να διατάσσει δύο αντικείμενα `account` σύμφωνα με το υπόλοιπό τους.
9. Προσθέστε κατάλληλη συνάρτηση μέλος έτσι ώστε να μετατρέπει ένα αντικείμενο `account` σε `string`.
10. Κατασκευάστε `main` που:
  - I. Δημιουργεί έναν λογαριασμό `acc1` με τον `default constructor`.
  - II. Δημιουργεί έναν λογαριασμό `acc2` με τις τιμές `ιδιοκτήτης=Ken Lay`, `υπόλοιπο=100.000€`, `τύπος λογαριασμού=MONEY_MARKET`
  - III. Δημιουργεί χρησιμοποιώντας τον `copy constructor` έναν τρίτο λογαριασμό `acc3` χρησιμοποιώντας τον λογαριασμό `acc2`.
  - IV. Εμφανίστε το πλήθος των λογαριασμών που έχουν δημιουργηθεί χρησιμοποιώντας την στατική μεταβλητή.
  - V. Πραγματοποιήστε ανάληψη ενός ποσού που θα δίνει ο χρήστης από τον λογαριασμό `acc2` και κατάθεσή του κατά 80% στο λογαριασμό `acc3` και κατά 20% στο λογαριασμό `acc1`.
  - VI. Εμφανίζει ποιος από τους δύο λογαριασμούς του Ken Lay έχει το μικρότερο υπόλοιπο.

#### Θέμα 2

Να κατασκευάσετε κλάση με όνομα `vehicle` (όχημα) και τις υποκλάσεις αυτής `car` (αυτοκίνητο) και `truck` (φορτηγό). Για την κλάση `Vehicle` ορίστε τα προστατευμένα μέλη δεδομένων `registration_number` (αριθμός άδειας) και `owner_name` (όνομα ιδιοκτήτη) και `cc` (κυβικά εκατοστά). Για την κλάση `car` ορίστε το μέλος δεδομένων `number_of_doors` (αριθμός πορτών) ενώ για την κλάση `Truck` ορίστε το μέλος δεδομένων `max_weight` (μέγιστο βάρος φόρτωσης).

1. Κατασκευάστε constructors και για τις τρεις κλάσεις έτσι ώστε να αρχικοποιούν όλα τα μέλη δεδομένων που διαθέτει η κάθε μια με τιμές παραμέτρων που θα δέχονται.
2. Ορίστε τη virtual συνάρτηση μέλος `traffic_tax` (τέλη κυκλοφορίας) στην κλάση `vehicle`.
3. Ορίστε την κλάση `vehicle` ως abstract τροποποιώντας κατάλληλα τη συνάρτηση `traffic_charges`.
4. Ορίστε τη συνάρτηση `traffic_tax` στην κλάση `car` έτσι ώστε να υπολογίζει τα τέλη κυκλοφορίας ως εξής: για αυτοκίνητα μέχρι και 1000 κυβικά εκατοστά τα τέλη είναι 140€ ενώ για πάνω από 1000 κυβικά εκατοστά για κάθε συμπληρωμένα πλήρως 100 κυβικά εκατοστά άνω των 1000 κυβικών εκατοστών υπάρχει επιπλέον χρέωση 10€ (π.χ. για 1532 κυβ.εκ. η χρέωση είναι  $140€ + 5 * 10€ = 190€$ )
5. Ορίστε τη συνάρτηση `traffic_tax` στην κλάση `truck` έτσι ώστε να υπολογίζει τα τέλη κυκλοφορίας ως εξής: για φορτηγά με μέγιστο βάρος φόρτωσης μέχρι και 3000 κιλά η χρέωση είναι 300€, για φορτηγά με μέγιστο βάρος φόρτωσης μέχρι και 6000 κιλά η χρέωση είναι 400€ και για ακόμα μεγαλύτερα φορτηγά η χρέωση είναι 600€.
6. Υπερφορτώστε τον τελεστή `<<` έτσι ώστε να εμφανίζει τα πλήρη στοιχεία οχήματος για την κλάση `car`.
7. Υπερφορτώστε τον τελεστή `<<` έτσι ώστε να εμφανίζει τα πλήρη στοιχεία οχήματος για την κλάση `truck`.
8. Γράψτε συνάρτηση `total_tax` που να δέχεται έναν πίνακα 5 θέσεων με δείκτες προς αντικείμενα είτε `car` είτε `truck` και να επιστρέφει το σύνολο τελών κυκλοφορίας που πρέπει να πληρωθούν από όλα τα οχήματα.
9. Γράψτε `main` που:
  - I. Δηλώνει πίνακα 5 θέσεων που θα έχει τη δυνατότητα να δεχθεί δείκτες προς αντικείμενα είτε `car` είτε `truck`.
  - II. Για κάθε ένα από 5 οχήματα να δέχεται τιμές από τον χρήστη έτσι ώστε ο χρήστης να επιλέγει το εάν το όχημα που εισάγει είναι αυτοκίνητο ή φορτηγό, εν συνεχεία να δέχεται τα στοιχεία του οχήματος και να το τοποθετεί στον πίνακα.
  - III. Να καλεί τη συνάρτηση `total_tax` και να εμφανίζει το αποτέλεσμα που επιστρέφει.

### Θέμα 3

Κατασκευάστε μια templated συνάρτηση που να δέχεται ένα διάνυσμα και ένα στοιχείο και να επιστρέφει πόσες φορές υπάρχει το στοιχείο μέσα στο διάνυσμα. Να κληθεί η συνάρτηση:

1. Για ένα διάνυσμα ακεραίων.
2. Για ένα διάνυσμα πραγματικών.
3. Για ένα διάνυσμα με εγγραφές σημείων (`struct Point`) όπου κάθε σημείο προσδιορίζεται από 2 αέριες τιμές.

Οι απαντήσεις θα ανακοινωθούν στις 12/4/2019 στο <https://github.com/chgogos/oop>