
ООР – PROJECT (ΕΡΓΑΣΙΑ ΜΑΘΗΜΑΤΟΣ)

Μεταγλώττιση: *make* (περιέχεται Makefile) ή *g++ -o Project *.cpp*

Εκτέλεση: *./Project* ή *./Project -R* (για εγγραφή Punter)

Στον παραδοτέο φάκελο, υπάρχει ένας φάκελος SystemFiles με τα αρχεία που διαχειρίζεται το πρόγραμμα.

ΜΟΡΦΗ ΑΡΧΕΙΩΝ

➤ hierarchy.dat:

Κάθε γραμμή αποτελεί έναν κόμβο(Node). Ο Node της κάθε γραμμής υποδεικνύεται από 1 έως 5 αριθμούς (όπως στην εκφώνηση), οι οποίοι χωρίζονται με «.». Ο τελευταίος αριθμός ΔΕΝ πρέπει να ακολουθείται από «.» .

Μετά από τους αριθμούς αυτούς (node_id) , μπορεί να υπάρχει παράμετρος «[H]» ή «[Hidden]», για να δηλώσει ότι ο κόμβος είναι κρυφός. Επιπρόσθετα, για κόμβο Selection μπορεί επιπλέον να υπάρχει «[W]», «[L]», «[V]» ή «[Win]», «[Loss]», «[Voided]» αντίστοιχα (ανεξάρτητα αν είναι hidden ή όχι) , για να δηλώσει το αποτέλεσμα του, ενώ για κόμβο Market μπορεί να υπάρχει «[S]» ή «[Settled]» , για να δηλώσει ότι είναι settled.

Ακολουθεί το όνομα του κόμβου (το οποίο μπορεί να περιέχει white spaces). Εάν πρόκειται για Event: μετά το όνομα πρέπει να υπάρχει «|», ακολουθούμενο από ημερομηνία και ώρα για το Event με την μορφή «**dd/mm/yyyy hh:mm**» (day, month, year, hours, minutes). Εάν πρόκειται για Selection: μετά το όνομα πρέπει να υπάρχει «#», ακολουθούμενο από την απόδοσή(price) του σε κλασματική(fractional) αναπαράσταση.

- users.csv: Ακολουθεί πιστά την εκφώνηση.
- bets.csv: Ακολουθεί πιστά την εκφώνηση.
- audit.log: Κάθε γραμμή είναι μία εγγραφή για μία system action.
- profit.txt: Χρησιμοποιείται για να φυλάξει το κέρδος του συστήματος (αλλιώς θα χανόταν σε κάθε εκτέλεση του προγράμματος). Διαθέτει μια μοναδική γραμμή με έναν δεκαδικό αριθμό ο οποίος αναπαριστά το κέρδος.

ΜΟΝΤΕΛΟΠΟΙΗΣΗ

- file manager:
Στα αρχεία file_manager.cpp/file_manager.h υλοποιούνται όλες οι συναρτήσεις που αφορούν την επικοινωνία του προγράμματος με τα αρχεία του Συστήματος. Θα μπορούσαν (πολλές αν όχι όλες) από αυτές να ήταν συναρτήσεις-μέλη της κλάσης system, μιας και αφορούν τα αρχεία του Συστήματος για

να είναι encapsulated στο namespace της κλάσης, αλλά δεν είναι απαραίτητο μιας και δεν είναι (ακριβώς) «συμπεριφορά» του συστήματος ως αντικείμενο και δεν πρόκειται να έχουμε πάνω από ένα αντικείμενο System.

- Σύστημα(class System):

Η class System έχει πεδία για τα χαρακτηριστικά (π.χ. name, profit, ...) του Συστήματος. Επίσης, έχει μια λίστα με τους εγγεγραμμένους χρήστες (RegisteredUser* list) , μια λίστα με όλα τα στοιχήματα στο Σύστημα (bet* list) (ανεξαρτήτως αποτελέσματος) και την ιεραρχία (Hierarchy*). Τέλος, έχει μια FIFO ουρά για να φυλάει εγγραφές system actions για το audit.log. Τα παραπάνω φορτώνονται από τα αρχεία κατά την αρχικοποίηση του Συστήματος (constructor) και σώζονται στα αρχεία όταν γίνεται save στο Σύστημα.

- Ιεραρχία:

Η ιεραρχία αναπαρίσταται στο πρόγραμμα σαν ένα δέντρο.

Κάθε επίπεδο στο δέντρο αυτό, περιέχει κόμβους από ένα επίπεδο της ιεραρχίας: Category, Subcategory, Event, Market, Selection. Κάθε τέτοιος κόμβος (εκτός από το Selection) έχει μια λίστα από (δείκτες σε) κόμβους του επόμενου επιπέδου.

Οι κλάσεις Category, SubCategory, Event, Market, Selection κληρονομούν από μία abstract (pure virtual) κλάση Node στην οποία υπάρχει ένα κοινό interface (και data members) για τις παραπάνω (πιο ειδικές) κλάσεις.

Η κλάση Hierarchy, η οποία δεν είναι Node, αποτελεί τη ρίζα του δέντρου και διαθέτει μία λίστα με τις κατηγορίες (Category* list), δηλαδή το πρώτο επίπεδο του δέντρου.

Σημειώνεται ότι αντί για λίστα με δείκτες σε κόμβους του επόμενου επιπέδου σε κάθε υποκλάση της class Node, θα μπορούσε να υπάρχει μια λίστα με Node * στην κλάση Node, έτσι ώστε να εκμεταλλευτούμε την κληρονομικότητα (και να κάνουμε override όπου χρειαζόταν). Έτσι θα αποφεύγαμε επαναλήψεις σε συναρτήσεις που κάνουν παρόμοιες ενέργειες πάνω στις λίστες. Ωστόσο, όταν το σκεφτήκαμε, τα πλεονεκτήματα μιας τέτοιας αλλαγής στην μοντελοποίηση δεν «άξιζαν» τις αλλαγές που θα απαιτούσε στον κώδικα (ο οποίος ήταν ήδη λειτουργικός).

- Χρήστες(Users):

Οι χρήστες χωρίζονται σε guest και εγγεγραμμένους χρήστες (RegisteredUsers).

Ο guest έχει μόνο δυνατότητες πλοήγησης και γι' αυτό δεν υλοποιείται ως ξεχωριστή κλάση. Η λειτουργικότητα για έναν guest είναι κοινή για όλους τους χρήστες και υλοποιείται απευθείας στην main().

Η κλάση RegisteredUser είναι abstract (pure virtual) και αποτελεί κοινό interface – είναι υπερκλάση - για όλους τους εγγεγραμμένους χρήστες του συστήματος (Punter, Trader, Director): ο Punter και ο Trader κληρονομούν από τον RegisteredUser (σχέση «is-a»), ενώ ο Director κληρονομεί από τον Trader.

ΠΑΡΑΔΟΧΕΣ ΥΛΟΠΟΙΗΣΗΣ

- Τα αρχεία του συστήματος ενημερώνονται αυτόματα κατά την έξοδο του χρήστη από το πρόγραμμα. Αυτό δεν ισχύει στην περίπτωση του Director, ο οποίος οφείλει να εκτελέσει την εντολή Save για το σκοπό αυτό.
- Η ιεραρχία ορίζεται ως consistent όταν κάθε ορατός της κόμβος (εκτός του Selection) περιέχει τουλάχιστον 1 ορατό υπο-κόμβο και κάθε ορατό Market περιέχει τουλάχιστον 2 ορατά Selections. Απαγορεύεται οποιαδήποτε λειτουργία θα καθιστούσε την ιεραρχία inconsistent.
- Τα voided Selections αντιμετωπίζονται ως hidden, όσον αφορά την consistency της hierarchy.
- Για λόγους consistency, κάθε κόμβος που παράγεται από εντολή New κάποιου Director είναι από προεπιλογή κρυφός (hidden). Εξαίρεση αποτελεί η δημιουργία κάποιου Selection το οποίο - εφόσον υπάρχει ήδη τουλάχιστον ακόμα ένα ορατό Selection - θα είναι κι αυτό ορατό.
- Κατά τη δημιουργία Event, ο Director εισάγει πάντα ημερομηνία και ώρα για αυτό, ενώ αντίστοιχα για τη δημιουργία Selection εισάγει την απόδοσή του.
- Καθώς οι Punters δεν μπορούν να δουν hidden Selections για να στοιχηματίσουν επί αυτών, ούτε οι Traders για να τα κάνουν Settle/Void, θεωρούμε ότι δεν είναι δυνατή η λειτουργία Settle/Void για hidden Selections, ούτε από Directors.
- Όταν γίνεται Delete κάποιου κόμβου διαγράφονται όλα τα bets σε Selections της υπο-ιεραρχίας που διαγράφηκε και οι χρήστες που τα τοποθέτησαν παίρνουν πίσω τα λεφτά που είχαν στοιχηματίσει.

- Μετά από κάθε Delete, διορθώνονται τα id των κόμβων της ιεραρχίας όπου αυτά εμφανίζονται, προκειμένου να καλυφθούν τυχόν κενά που δημιουργήθηκαν.
- Το μέγιστο όριο κουπονιών για κάθε παίκτη (Punter) είναι 26.
- Δεν επιτρέπεται να δημιουργηθεί Selection σε ένα Market που έχει γίνει settled.
- Αν γίνει Voided μια Selection, τότε για όλα τα στοιχήματα σε αυτή θα επιστραφούν τα χρήματα στοιχηματισμού(stake) στο wallet(balance) των Punters, ακόμα κι αν προέρχονταν από freebet.

ΕΠΙΠΛΕΟΝ ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ

- ❖ Εντολή «**Ba (Back)**»: Χρησιμοποιείται κατά την πλοήγηση σε οποιοδήποτε επίπεδο (εκτός του Home) και μεταφέρει τον χρήστη στο αμέσως προηγούμενο (ανώτερο) επίπεδο. Μπορεί να χρησιμοποιηθεί από οποιονδήποτε χρήστη.
- ❖ Η Εντολή «**Vi (Visibility)**» δίνει τη δυνατότητα στο χρήστη, εφόσον το επιθυμεί, εκτός από την ορατότητα του επιλεγμένου κόμβου να αλλάξει επίσης και την ορατότητα της υπο-ιεραρχίας του. Σε κάθε περίπτωση, γίνεται έλεγχος για διατήρηση της consistency της ιεραρχίας.
- ❖ Εντολή «**SP (System_profits)**»: Τυπώνει τα κέρδη του συστήματος. Μπορεί να χρησιμοποιηθεί μόνο από Director.
- ❖ Όπως αναφέρθηκε και στη «Μορφή Αρχείων», το κέρδος του συστήματος διατηρείται σε ξεχωριστό αρχείο *profits.txt*.

ΕΠΙΠΛΕΟΝ ΣΧΟΛΙΑ

Υπάρχει και η εντολή «**PP (Pretty_print)**», που εκτυπώνει ολόκληρη την ιεραρχία στην οθόνη αλλά δεν εμφανίζεται στις επιλογές (κρυφή). Χρησιμοποιήθηκε κυρίως για debugging και είναι προσβάσιμη μόνο από Director (δεν αποτελεί «επίσημη» λειτουργία/υπηρεσία του Συστήματος).