



ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ OOP- Master Chef

ΕΡΓΑΣΙΑ 2

Υλοποίηση βασικών κλάσεων



Καρακώστας Κωνσταντίνος 7892 karakosk@ece.auth.gr

Κοσεόγλου Σωκράτης 8837 Sokrkose@ece.auth.gr

25/04/2018

Στην παρούσα εργασία μας ζητήθηκε να υλοποιήσουμε τις δύο βασικές κλάσεις του προγράμματος, κλάση Player και Team. Επίσης υλοποιήθηκε η συνάρτηση main η οποία αποτελεί την διασύνδεση του χρήστη με την εφαρμογή.

Κλάση Team

Η κλάση team δημιουργεί τις ομάδες της πλατφόρμας, συγκεκριμένα την κόκκινη και μπλε ομάδα. Το αρχείο με κατάληξη .cpp περιέχει τον κώδικα που αρχικοποιεί τις μεταβλητές και ορίζει της λειτουργία των συναρτήσεων. Από την άλλη το αρχείο με κατάληξη .h είναι ο header της κλάσης Team και περιέχει μόνο την μέθοδο του constructor και τις μεθόδους get/set. Φυσικά ορίζονται και οι τύποι των μεταβλητών που θα χρησιμοποιήσει η κλάση.

Team.cpp

Οι μεταβλητές που χρησιμοποιήθηκαν και ο τύπος τους είναι οι: `playersNumber:int`, `teamsWinsCounter:int`, `color:string`, `numberOfSupplies:int` και ο πίνακας `players[11]:Player`. Η αρχική κατάσταση για τις τύπου int είναι 0 και Blue για την color.

Ο αναγνώστης διαβάζοντας τον κώδικα θα συναντήσει ως πρώτη μέθοδο τον constructor ο οποίος καλεί την συνάρτηση `checkColor(string)`. Ο ρόλος της παραπάνω συνάρτησης είναι να ελέγχει αν έγινε σωστή εισαγωγή του χρώματος της ομάδας, δηλαδή είτε Red είτε Blue.

Επείτα ακολουθούν οι μέθοδοι get/set της κλάσης με την παρακάτω σύνταξη

```
int Team::getPlayersNumber() {
    return playersNumber;
};
void Team::setPlayersNumber(int playersNumber) {
    this->playersNumber = playersNumber;
}
```

Συγκεκριμένα για τους getters/setters του πίνακα players καλείται η συνάρτηση `checkNumber` ώστε να ελεγχθεί αν έγινε σωστή εισαγωγή του αριθμού του παίχτη. Γνωρίζουμε ότι οι παίχτες σε μία ομάδα είναι το πολύ 11 άρα η συνάρτηση ελέγχει αν ο πίνακας player δεν ξεπερνά τις 11 θέσεις.

Τέλος υπάρχει η συνάρτηση `teamsWins()` η οποία για κάθε νίκη της ομάδας αυξάνει την τιμή της μεταβλητής `teamsWinsCounter` κατά ένα.

Κλάση Player

Για την κλάση player η λογική σχεδίασης της είναι ακριβώς η ίδια με την λογική σχεδίασης της κλάσης Team όπως παρουσιάστηκε παραπάνω.

Player.cpp

Εδώ οι μεταβλητές που χρησιμοποιήθηκαν είναι οι: `name:string`, `age:int`, `sex:string`, `occupation:string`, `technicalExperience:float`, `fatigue:float`, `popularity:float`, `winsNumber:int`.

Αρχική κατάσταση για τις string είναι “ ”, για την `age` είναι -1, `fatigue=0`, `winsNumber=0`, `popularity=50` και για την `technicalExperience` μία τυχαία τιμή όπως προκύπτει από την $((\text{float})\text{rand}() / (\text{float})(\text{RAND_MAX})) * 100$.

Διαβάζοντας τον κώδικα παρατηρούμε τον constructor της κλάσης. Εντός αυτού καλούνται οι συναρτήσεις `checkRange` για την `technicalExperience` όπου ελέγχει αν ο αριθμός που δημιουργήθηκε από την συνάρτηση `rand()` είναι εντός των τιμών 0 και 100. Καλείται επίσης η συνάρτηση `checkAge` για τον έλεγχο της ηλικίας μιας και πρέπει να εισαχθεί τιμή μεγαλύτερη του 18. Τέλος καλείται η συνάρτηση `checkSex` για να ελέγξει ότι έγινε σωστή εισαγωγή των φύλων, Male or Female. [trigger warning] (σε περίπτωση λάθους επιστρέφει την τιμή “Male”).

Στην συνέχεια του κώδικα ακολουθούν οι μέθοδοι `get/set` όπως ακριβώς και στην κλάση Team.

Τέλος ακολουθούν οι συναρτήσεις της κλάσης player όπως ορίστηκαν στο διάγραμμα UML.

Η συνάρτηση `work()` αυξάνει την μεταβλητή `technicalExperience` κατά 5% και την `fatigue` κατά ένα τυχαίο ποσοστό από το 20% έως 40%. Οι συνάρτησεις `eat()` και `compete()` προς το παρόν δεν κάνουν κάποιες ενέργειες. Η συνάρτηση `sleep()` θέτει την μεταβλητή `fatigue` ίσο με μηδέν. Η συνάρτηση `socialize()` μεταβάλλει την τιμή της μεταβλητής κατά ένα τυχαίο τρόπο από -10% έως 10%. Τέλος, η συνάρτηση `lastDayOfTheWeek()` επιλέγει κατά τυχαίο τρόπο (50-50 οι πιθανότητες) αν ένας παίχτης θα εξασκηθεί μόνος του ώστε να αυξήσει την μεταβλητή `technicalExperience` κατά 5% ή αν θα αποφασίσει να κοιμηθεί ώστε να καλέσει την συνάρτηση `sleep()`.

Συνάρτηση main

Στο αρχείο `main.cpp` βρίσκεται η συνάρτηση `main` η οποία ουσιαστικά «στήνει» και διαχειρίζεται το παιχνίδι. Αποτελεί την “διασύνδεση” του χρήστη με τον κώδικα. Οι λειτουργίες της `main` είναι οι εξής:

- Εισαγωγή παίχτη σε κάποια ομάδα
- Προβολή στοιχείων κάποιας ομάδας
- Προβολή στοιχείων κάποιου παίχτη
- Έξοδος από το πρόγραμμα

Θα λεγαμε με κάποιο τρόπο ότι αποτελεί το UI από το οποίο ο χρήστης του προγράμματος μπορεί να διαχειριστεί το παιχνίδι.

Ξεκινώντας η main εμφανίζει στο χρήστη τις 4 παραπάνω επιλογές. Ο χρήστης κάνει την επιλογή του πατώντας τον αριθμό που του ζητείται.

Προσθήκη παίχτη

Ο χρήστης πατά τον αριθμό 1. Αμέσως εμφανίζεται στο περιβάλλον διεπαφής το μήνυμα όπου τον ρωτά σε ποιά ομάδα θέλει να προσθέσει τον καινούριο παίχτη. Ο χρήστης επιλέγει την ομάδα και εκτελείται η συνάρτηση `addPlayer(string)`. Αυτή η συνάρτηση ζητά από τον χρήστη να εισάγει ένα-ένα τα στοιχεία του καινούριου παίχτη. Εφόσον γίνει αυτό, τα στοιχεία που δόθηκαν περνούν στο αντικείμενο τύπου `player` ώστε να αποκτήσει υπόσταση ο παίχτης και αυξάνεται ο α/α των παιχτων κατά ένα ώστε ο επόμενος παίχτης να μην εισαχθεί στην ίδια θέση του πίνακα.

Προβολή ομάδας

Ο χρήστης πατά τον αριθμό 2. Εμφανίζεται μήνυμα στην οθόνη που ζητά την επιλογή της ομάδας για την οποία θα προβληθούν τα στοιχεία της. Εφόσον επιλέξει ο χρήστης εκτελείται η συνάρτηση `teamPrinter(string)`. Η συνάρτηση αυτή εκτυπώνει τα στοιχεία της ομάδας, αριθμό παιχτών, νίκες και προμήθειες. Παράλληλα εκτυπώνει και τους παίχτες της ομάδας μαζί με τα στοιχεία τους.

Προβολή παίχτη

Ο χρήστης πατά τον αριθμό 3. Εμφανίζεται μήνυμα στην οθόνη όπου ζητά την επιλογή της ομάδας για την οποία θέλει ο χρήστης να προβάλει τα στοιχεία των μελών της. Εφόσον δοθεί εκτελείται η συνάρτηση `PlayersPrinter(string)`. Η συνάρτηση αυτή ρωτά τον χρήστη με ποιον τρόπο θέλει να τυπώσει τα στοιχεία των παιχτών. Οι επιλογές είναι, είτε με αλφαβητική σειρά, είτε με την θέση που έχουν στον πίνακα (ουσιαστικά με ποια σειρά έγινε η αρχική εγγραφή τους). Αν ο χρήστης επιλέξει το δεύτερο τότε τυπώνει σειριάκα του παίχτες μαζί με τα στοιχεία τους, αν επιλέξει το 2^ο τότε η συνάρτηση τοποθετεί ΜΟΝΟ τα ονόματα των παιχτών στην λίστα `NameList[int]`. Έπειτα εκτελείται η συνάρτηση `bubbleSort(Team*, string)` που είναι μία τυπική ταξινόμηση φυσαλίδας. Πλέον η `NameList` περιέχει τα ονόματα των παιχτών σε αλφαβητική σειρά. Προφανώς ο πίνακας `Players` δεν έχει επηρεαστεί. Τέλος για να εκτυπώσει τα στοιχεία των παιχτών ψάχνει να βρει το όνομα του παίχτη που βρίσκεται στην `NameList` στην θέση που έχει στον πίνακα `Players`. Εφόσον το βρει τυπώνει τα στοιχεία του.

Έξοδος

Ο χρήστης πατά τον αριθμό 4. Η εφαρμογή τερματίζει.