

# 2<sup>η</sup> Εργασία OOP: MasterChef και C++

Βέλλιος Γεώργιος-Σεραφείμ

Μανούσος Διαγόρας

AEM: 9471

AEM: 9554

velliosg@ece.auth.gr

dmanouso@ece.auth.gr



## **Πρόβλημα προς επίλυση:**

Δημιουργία των κλάσεων *Player* και *Team* ( η δήλωση τους, των μεταβλητών και των μεθόδων τους σε αρχεία *.h*, καθώς και υλοποίηση των μεθόδων αυτών σε αντίστοιχα αρχεία *.cpp*), και μίας *main* στο αρχείο *main.cpp* που θα περιλαμβάνει ένα κεντρικό μενού με τις εξής δυνατότητες:

1. Προσθήκη Παίκτη σε μια Ομάδα
2. Προβολή των Στοιχείων της Ομάδας-
3. Προβολή των στοιχείων ενός Παίκτη μιας Ομάδας (είτε κατ' όνομα, είτε κατά θέση του πίνακα)
4. Να βγει από το πρόγραμμα

Η *main.cpp*, πέραν του κώδικα και των συναρτήσεων που χρειάζονται για τη επίλυση του παραπάνω προβλήματος, περιέχει σε */\* \*/* κώδικα στον οποίο δοκιμάζεται η ορθή λειτουργία όλων των λειτουργιών των δύο προαναφερθέντων κλάσεων (θα γίνει αναφορά σε αυτόν αργότερα)

## **Επίλυση του Προβλήματος:**

- Ο τύπος  $(max - min) * rand() / RAND\_MAX + min$  δίνει ένα τυχαίο αριθμό μεταξύ του *min* και του *max* (Θα χρησιμοποιηθεί αρκετές φορές στον κώδικα).

Αρχικά δημιουργήθηκαν οι κεφαλίδες *Player.h* , *Team.h* όπου δηλώνονται οι δύο κλάσεις, οι μεταβλητές και οι μέθοδοί τους και στην συνέχεια τα αρχεία *Player.cpp*, *Team.cpp* όπου υλοποιούνται οι δηλώσεις που έγιναν στις αντίστοιχες κεφαλίδες. Μεταβλητές της κλάσης *Player* είναι οι:

1. *string name*: Όνομα
2. *string gender*: Φύλο
3. *int age*: Ηλικία
4. *string job*: Επάγγελμα
5. *float technicalTraining*: Τεχνική Κατάρτιση
6. *float exhaustion*: Κούραση
7. *float popularity*: Ποσοστό Δημοφιλίας
8. *int wins* : Νίκες

Και της κλάσης *Team* είναι οι:

1. *int wins*: Νίκες
2. *string colour*: Χρώμα της ομάδας
3. *int portions*: Μέρηδες που δίνονται στην ομάδα
4. *Player players[11]*: Πίνακας των παικτών που περιέχει η ομάδα (Περιέχει στοιχεία τύπου *Player*)

Οι μέθοδοι των δύο κλάσεων καθώς και λεπτομέρειες για τον τρόπο λειτουργίας τους παρατίθενται στα αντίστοιχα αρχεία με την μορφή σχολίων. Εν συντομία, επαναλαμβάνεται ότι η κλάση του παίκτη έχει 8 μεταβλητές (οπότε 8 *setters* και 8 *getters*), 2 *constructors* , με και χωρίς ορίσματα, έναν *destructor* και 8 επιπλέον μεθόδους, ενώ κάθε ομάδα έχει 4 μεταβλητές (οπότε 4 *setters* και 4 *getters*), 2 *constructors*, με και χωρίς ορίσματα, έναν *destructor* και 1 επιπλέον μέθοδο (*showTeam(intnum)*).

Αφού δημιουργήθηκαν οι απαραίτητες κλάσεις προχωράμε στην δημιουργία της *main* στο *main.cpp*, καθώς και των απαραίτητων συναρτήσεων που χρειάζονται για την λειτουργία της. Πριν την *main* δηλώνουμε 4 καθολικές μεταβλητές *pBlue*(=0), *pRed*(=0), *portionsBlue*(=14), *portionsRed*(=14) . Οι *pBlue* και *pRed* δείχνουν σε ποιον παίκτη σε καθεμία από τις δύο έχουμε μείνει. Πχ. αν προσθέσουμε έναν παίκτη στη μπλε ομάδα ο *pBlue* από 0 γίνεται 1. Οι *portions* ξεκινάνε από 0 για κάθε ομάδα και κάθε φορά που προστίθεται ένας παίκτης αυξάνονται κατά 14 μέσω της συνάρτησης *addPlayer(string)*. Ακόμη δημιουργούνται 2 κλάσεις *Team*, μία με *colourBlue* και *default* τα άλλα ορίσματα και μία με *colourRed* και *default* τα υπόλοιπα. Στην *main* ξεκινάει μια γεννήτρια ψευδοτυχαίων αριθμών (Για τις μεταβλητές *technicalTraining*, *exhaustion*, *popularity* του *Player* και τις συναρτήσεις που τις επηρεάζουν).

## **Κύριο Μέρος:**

Το κύριο μενού του προγράμματος αποτελείται από μία *do-while* που σταματάει όταν η *choice1==4*. Η *do-while* περιέχει μια *switch* με όρισμα *intchoice1* που δίνεται από τον χρήστη μέσω πληκτρολογίου και 4 *cases*(εκτελούν τα 4 ζητούμενα της άσκησης) (+1 *default* που εκτυπώνει ένα μήνυμα λάθος επιλογής).

### **Case 1:**

Ζητείται από τον χρήστη να πληκτρολογήσει σε *string* το χρώμα της ομάδας που θέλει να τοποθετήσει τον παίκτη. Κατόπιν καλείται η συνάρτηση *PlayeraddPlayer(strings)* που τοποθετεί στοιχεία σε ένα *Playerp* που αρχικοποιείται κάθε φορά που εκτελείται η *do-while* και στη συνέχεια ελέγχεται το *string* για να τοποθετηθεί ο *player* στην σωστή ομάδα.

```
player = addPlayer(s);
if(s=="Blue" && pBlue < 11){
    blueTeam.setPlayers(pBlue, player);
    blueTeam.setPortions(portionsBlue);
    pBlue++;
    portionsBlue+=14;
}
else if(s=="Red" && pRed < 11){
    redTeam.setPlayers(pRed, player);
    redTeam.setPortions(portionsRed);
    pRed++;
    portionsRed+=14;
}
else
;
```

### Εικόνα 1: Τοποθέτηση του παίκτη στην σωστή Ομάδα

Οι μεταβλητές `rBlue`, `rRed` λένε στο πρόγραμμα σε ποια θέση του πίνακα να τοποθετηθεί ο καινούργιος παίκτης και αυξάνονται κάθε φορά που αυτό συμβαίνει. Επίσης τοποθετούνται και οι `portions` στην αντίστοιχη ομάδα (14 για τον κάθε παίκτη που προστίθεται) και αυξάνεται το `portionsBlue` ή `portionsRed`, ώστε την επόμενη φορά που θα κληθεί, να τεθεί η προηγούμενη τιμή +14. (Αν έχουν τοποθετηθεί ήδη 11 παίκτες τότε δεν τοποθετείται και άλλος παρά εμφανίζεται ένα μήνυμα λάθους)

#### Case 2:

Καλείται η συνάρτηση `void ektiposiOmadas(Team,Team)`, η οποία εκτυπώνει τα στοιχεία της ομάδας που επιλέγεται (Wins, Portions, colour, Namesofplayers).

#### Case 3:

Καλείται η συνάρτηση `void ektiposiPaikti(Team,Team)`, η οποία εκτυπώνει τα στοιχεία του παίκτη που επιλέγεται.

#### Case 4:

Εκτυπώνεται το μήνυμα "Antio" και σταματάει να εκτελείται η *do-while*.

### Έλεγχος όλων των δυνατοτήτων των μεθόδων των κλάσεων:

Οι παρακάτω γραμμές κώδικα υπάρχουν στην `main` ως σχόλια. Αν βγουν από τα σχόλια εκτελούνται μετά την έξοδο από την `switch` (επιλογή του 4) και ελέγχουν αν λειτουργούν σωστά όλες οι μέθοδοι και τα constructors του κώδικα. Για απόδειξη αυτού θα εκτελεστεί μία φορά και τα αποτελέσματα θα παρατεθούν από κάτω.

```
Player player1("Giannis", "Male", 19, "Odontiatros", 2);
cout<<"Player 1"<<endl;
player1.showPlayer();
Player player2("Dimitris", "male", 30, "Hlektrologos");
cout<<"Player 2"<<endl;
player2.showPlayer();
Player player3;
cout<<"Player 3"<<endl;
player3.showPlayer();
Team team1("Kokkino");
cout<<"Team Kokkino"<<endl;
team1.showTeam();
Team team2("Portokali", 3);
cout<<"Team Portokali"<<endl;
team2.showTeam();
Team team3("Kafe", 53, 234);
cout<<"Team Kafe"<<endl;
team3.showTeam();
Team team4;
cout<<"Team Default"<<endl;
team4.showTeam();
cout<<"Player 1"<<endl;
player1.showPlayer();
cout<<"Player 1 after work()"<<endl;
player1.work();
player1.showPlayer();
cout<<"Player 1 after communicate()"<<endl;
player1.communicate();
player1.showPlayer();
cout<<"Player 1 after chooseSleepOrWork()"<<endl;
player1.chooseSleepOrWork();
player1.showPlayer();
cout<<"Player 1 after participateInContest"<<endl;
player1.participateInContest();
player1.showPlayer();
```

Εικόνα 2: Έλεγχος όλων των δυνατοτήτων των μεθόδων των κλάσεων

Δημιουργούνται οι 3 παίκτες όπως θα έπρεπε (με τα ορίσματα που τους δίνονται και τα αυτόματα ορίσματα όπου δεν δίνονται από τον χρήστη):

- Player player1("Giannis", "Male", 19, "Odontiatros", 2);
- Player player2("Dimitris", "male", 30, "Hlektrologos");
- Player player3;

```
Player 1
Player:
Name: Giannis
Gender: Male
Age: 19
Job: Odontiatros
Wins: 2
Technical Training: 74.5408%
Exhaustion: 0%
Popularity: 50%

Player 2
Player:
Name: Dimitris
Gender: male
Age: 30
Job: Hlektrologos
Wins: 0
Technical Training: 9.16532%
Exhaustion: 0%
Popularity: 50%

Player 3
Player:
Name:
Gender:
Age: 18
Job:
Wins: 0
Technical Training: 35.5455%
Exhaustion: 0%
Popularity: 50%
```

Εικόνα 3: Δημιουργία των τριών Player και εκτύπωση τους

Δημιουργούνται οι 4 ομάδες όπως θα έπρεπε (με τα ορίσματα που τους δίνονται και τα αυτόματα ορίσματα όπου δεν δίνονται από τον χρήστη). Καμία ομάδα δεν έχει παίκτες (ή καλύτερα έχει 11 παίκτες στον πίνακα players[11] αλλά και οι 11 είναι default οπότε η εκτύπωση των ονομάτων τους θα είναι 11 κενές γραμμές ,αφού παίρνουν τον κενό χαρακτήρα για default όνομα) και για αυτό δεν επιλέγεται (θα γινόταν εφικτό, αν μέσα στη συνάρτηση showTeam(intnum=0)βάζαμε για num το 11).

```
Team Kokkino
Team colour: Kokkino
Wins: 0
Portions: 0
Player Names:

Team Portokali
Team colour: Portokali
Wins: 3
Portions: 0
Player Names:

Team Kafe
Team colour: Kafe
Wins: 53
Portions: 234
Player Names:

Team Default
Team colour:
Wins: 0
Portions: 0
Player Names:
```

Εικόνα 4: Δημιουργία των τεσσάρων Team και εκτύπωσή τους

Δοκιμάζεται η επίδραση κάποιων μεθόδων πάνω στον παίκτη και εμφανίζονται τα στοιχεία του μετά από αυτήν. (Τα αποτελέσματα είναι αναμενόμενα και περιγράφονται λεπτομερέστερα στο αρχείο Player.cpp πάνω από την κάθε συνάρτηση.

```
Player 1
Player:
Name: Giannis
Gender: Male
Age: 19
Job: Odontiatros
Wins: 2
Technical Training: 74.5408%
Exhaustion: 0%
Popularity: 50%

Player 1 after work()
Player:
Name: Giannis
Gender: Male
Age: 19
Job: Odontiatros
Wins: 2
Technical Training: 74.5408%
Exhaustion: 21.9068%
Popularity: 50%

Player 1 after communicate()
Player:
Name: Giannis
Gender: Male
Age: 19
Job: Odontiatros
Wins: 2
Technical Training: 74.5408%
Exhaustion: 21.9068%
Popularity: 51.7959%

Player 1 after chooseSleepOrWork()
O paiktis exaskitai
Player:
Name: Giannis
Gender: Male
Age: 19
Job: Odontiatros
Wins: 2
Technical Training: 79.5408%
Exhaustion: 21.9068%
Popularity: 51.7959%

Player 1 after participateInContest
O paiktis simmetixe se diagonismo kai kourastike
Player:
Name: Giannis
Gender: Male
Age: 19
Job: Odontiatros
Wins: 2
Technical Training: 79.5408%
Exhaustion: 24.2083%
Popularity: 51.7959%
```

Εικόνα 6: Δοκιμή των μεθόδων του Player

## **Συναρτήσεις της main.cpp:**

### **Player addPlayer(string s):**

Αποτελείται από μία δομή *if-else*. Σε κάθε μια από τις 2 περιπτώσεις ( $s == \text{"Blue"}$  or  $s == \text{"Red"}$ ) ελέγχεται αν ο πίνακας των παικτών έχει γεμίσει και, αν ναι, εκτυπώνει μήνυμα λάθους και επιστρέφει ένα defaultPlayer, ο οποίος δεν μπαίνει σε καμία ομάδα αφού υπάρχει έλεγχος και στη main, σχετικά με τον γεμάτο πίνακα. Αν ο πίνακας δεν είναι γεμάτος τότε ζητάει από τον χρήστη να πληκτρολογήσει τις μεταβλητές του παίκτη και τις τοποθετεί σε αυτόν με την χρήση των *setters*. Τέλος, επιστρέφει τον παίκτη, ο οποίος τοποθετείται από την main στην κατάλληλη ομάδα και θέση.

### **Void ektiposiOmadas():**

Ζητείται από τον χρήστη να πληκτρολογήσει την ομάδα, της οποίας τα στοιχεία θα εκτυπωθούν και με μία *if-else* και την κατάλληλη μέθοδο `showTeam(int)` εκτυπώνονται τα στοιχεία της ομάδας.

### **Void ektiposiPaikti():**

Αποτελείται από μία κύρια *if-else* όπου η επιλογή γίνεται με βάση το χρώμα της ομάδας (πληκτρολογείται από τον χρήστη) και μέσα σε καθεμία επιλογή υπάρχει μια *switch* όπου επιλέγεται με βάση την προτίμηση εύρεσης του παίκτη (κατά όνομα ή κατά θέση) που δίνεται από τον χρήστη.

#### *Κατά Όνομα:*

Λειτουργεί μία *for* η οποία κάνει τόσες επαναλήψεις όσο είναι η μεταβλητή `pBlue` or `pRed` και συγκρίνει το `string` που δίνεται από τον χρήστη με τα ονόματα των παικτών. Αν ταιριάζουν τότε εκτυπώνει τον παίκτη με την εντολή `...Team.getPlayers(i).showPlayer();` όπου η θέση του στον πίνακα.

#### *Κατά Θέση:*

Ζητείται η θέση και ελέγχεται με την συνθήκη ( $pos < 1 \ || \ pos > 11 \ || \ pos > pRed \ || \ pos > pBlue$ ), αν υπάρχει ο παίκτης. Αν υπάρχει εκτυπώνεται όπως παραπάνω, αλλιώς εμφανίζεται ένα μήνυμα λάθους.