

## **Μέρος Β: Πρώτη υλοποίηση κώδικα**

---

### **Αναφορά**

Αντικειμενοστραφής Προγραμματισμός

Νικόλαος Πίγκας 9329, [ngpigkas@ece.auth.gr](mailto:ngpigkas@ece.auth.gr)  
Αλέξανδρος Σαχίνης 8906, [alexsach@ece.auth.gr](mailto:alexsach@ece.auth.gr)

## Κλάση Player

```
class Player {  
  
    string name;  
    int age;  
    string gender;  
    string employment;  
    int fatigue;  
    double popularity;  
    double ability;  
    int winTotal;  
  
public:  
  
    Player();  
    Player(string na, int ag, string ge, string em);  
    ~Player();  
  
    void setName(string val);  
    void setAge(int val);  
    void setGender(string val);  
    void setEmployment(string val);  
    void setFatigue(int val);  
    void setPopularity(double val);  
    void setAbility(double val);  
    void setWinTotal(int val);  
  
    string getName();  
    int getAge();  
    string getGender();  
    string getEmployment();  
    int getFatigue();  
    double getPopularity();  
    double getAbility();  
    int getWinTotal();  
  
    void work();  
    void practice();  
    void eat();  
    void sleep();  
    void socialize();  
    void status();  
};
```

## Μεταβλητές

Η κλάση Player περιέχει τις ίδιες μεταβλητές που εξηγήθηκαν στην πρώτη αναφορά και γι' αυτό η χρήση τους είναι προφανής.

## Συναρτήσεις

**getters** και **setters**: Ορίζονται οι getters και οι setters για όλες τις μεταβλητές προκειμένου να παίρνουμε και να αλλάζουμε τις τιμές τους.

**Player()** και **Player(string, int, string, string)**: Συνάρτηση αρχικών συνθηκών με και χωρίς ορίσματα.

**~Player()**: Συνάρτηση τελικών συνθηκών που επιστρέφει μήνυμα καταστροφής για το αντικείμενο.

**void Player::work()** : Αυξάνει την κούραση και την τεχνική κατάρτιση του κάθε παίκτη, φροντίζοντας παράλληλα οι τιμές αυτές να παραμένουν στο διάστημα 0 έως 100.

**void Player::socialize()** η οποία αυξομειώνει τυχαία το ποσοστό δημοτικότητας του κάθε παίκτη, εξασφαλίζοντας ότι παραμένει στο διάστημα 0 έως 100.

**void Player::practice()** η οποία αυξάνει την τεχνική κατάρτιση του παίκτη.

**void Player::sleep()** η οποία μηδενίζει την κούραση του παίκτη.

**void Player::eat()** η οποία δεν έχει κάποια ιδιαίτερη λειτουργία παρά μόνο εμφανίζει σχετικό μήνυμα.

**void Player::status()** η οποία εμφανίζει τα στοιχεία (attributes) του παίκτη.

## Κλάση Team

```
class Team{

    Player members[11];
    int curNuOfMembers;
    string color;
    int numOfMeals;
    string ingredients;
    int winTotal;

public:
    Team();
    Team(string co, int nu, string in, int wi);
    ~Team();

    void setColor(string val);
    void setNumOfMeals(int val);
    void setIngredients(string val);
    void setWinTotal(int val);
    void setCurNuOfMembers(int val);

    string getColor();
    int getNumOfMeals();
    string getIngredients();
    int getWinTotal();
    int getCurNuOfMembers();

    void addMember(Player mem);
    void status();
    void dispMembers();
    void dispPlayer(int number);
    void dispPlayer(string name);

};
```

### Μεταβλητές

Η κλάση Team περιέχει, ως επί το πλείστον, μεταβλητές που εξηγήθηκαν στην πρώτη αναφορά και γι' αυτό η χρήση τους είναι προφανής. Για τις επιπλέον:

**Player members[11]:** Πίνακας 11 θέσεων, σε κάθε κελί του οποίου αποθηκεύεται ένας παίκτης της ομάδας.

**int curNuOfMembers:** Δηλώνει τον αριθμό των παικτών που βρίσκονται κάποια δεδομένη χρονική στιγμή στην ομάδα.

## Συναρτήσεις

**getters** και **setters**: Ορίζονται οι getters και οι setters για όλες τις μεταβλητές προκειμένου να παίρνουμε και να αλλάζουμε τις τιμές τους.

**Team()** και **Team(string, int, string, int)**: Συνάρτηση αρχικών συνθηκών με και χωρίς ορίσματα.

**~Team()**: Συνάρτηση τελικών συνθηκών που επιστρέφει μήνυμα καταστροφής για το αντικείμενο.

**void addMember(Player mem)**: Προσθέτει ένα παίκτη στην ομάδα.

**void Team::status()**: Εμφανίζει τα στοιχεία της ομάδας, συμπεριλαμβανομένων και των παικτών, ονομαστικά.

**void Team::dispPlayer(int number)**: Η οποία δοθέντος ενός αριθμού εμφανίζει τα στοιχεία του αντίστοιχου παίκτη με βάση τη θέση του στον πίνακα των members. Γίνεται έλεγχος τιμών για την είσοδο ώστε να αποφευχθούν τα σφάλματα.

**void Team::dispPlayer(string name)**: Η οποία δοθέντος του ονόματος ενός παίκτη εμφανίζει τα στοιχεία του, εφόσον βρεθεί ότι υπάρχει στην ομάδα. Σε διαφορετική περίπτωση επιστρέφεται μήνυμα σφάλματος.

## Αρχείο main.cpp

### Μεταβλητές

**Team blue**: αντικείμενο που ορίζεται έξω από τη main, προκειμένου να αποθηκεύεται οποιαδήποτε αλλαγή γίνεται πάνω σε αυτό μέσα σε εκείνη και προσομοιώνει την μπλε ομάδα του παιχνιδιού.

**Team red**: αντικείμενο που ορίζεται έξω από τη main, προκειμένου να αποθηκεύεται οποιαδήποτε αλλαγή γίνεται πάνω σε αυτό μέσα σε εκείνη και προσομοιώνει την κόκκινη ομάδα του παιχνιδιού.

## Συναρτήσεις

**int main():** Παρουσιάζει στον χρήστη ένα μενού προσφέροντας τις εξής λειτουργίες

- a) Προσθήκη παίκτη σε ομάδα, πατώντας το 1
- b) Προβολή των στοιχείων μιας ομάδας, πατώντας το 2
- c) Προβολή των στοιχείων ενός παίκτη μιας ομάδας, είτε κατ' όνομα, είτε κατά θέση του πίνακα, πατώντας το 3
- d) Έξοδο από το πρόγραμμα, πατώντας οποιοδήποτε άλλο πλήκτρο

**void addPlayerToTeam():** Η συνάρτηση αυτή έχει σκοπό την εισαγωγή ενός παίκτη σε μία ομάδα, με στοιχεία που θα δοθούν από το χρήστη του προγράμματος. Με την κλήση της ελέγχει αν και οι δύο ομάδες είναι γεμάτες, οπότε και επιστρέφει μήνυμα σφάλματος. Σε διαφορετική περίπτωση ζητά από το χρήστη τα στοιχεία του παίκτη που θέλει να δημιουργήσει. Όταν έρθει η ώρα να εισαχθεί ο παίκτης σε μία ομάδα, τότε αν η μία από τις 2 ομάδες είναι γεμάτη, ο παίκτης εισάγεται αυτόματα στην άλλη. Σε διαφορετική περίπτωση εισάγεται στην ομάδα που ο χρήστης επιλέγει.

**void displayTeamDetails():** Η συνάρτηση αυτή παρουσιάζει της λεπτομέρειες (attributes) της ομάδας που ο χρήστης έχει επιλέξει προς εμφάνιση.

**void displayPlayer():** Η συνάρτηση αυτή ζητά από το χρήστη να επιλέξει από ποια ομάδα θέλει να ψάξει έναν παίκτη προς εμφάνιση των στοιχείων του. Αφού επιλέξει, παρουσιάζει τα στοιχεία της ομάδας και ζητά από το χρήστη να επιλέξει αν θέλει να ψάξει έναν παίκτη με βάση το όνομα ή τον αριθμό του. Αν επιλέξει κάτι από τα δύο, το σύστημα ψάχνει τον παίκτη και επιστρέφει κατάλληλα μηνύματα σε περίπτωση που βρεθεί ή όχι. Διαφορετικά γίνεται έξοδος από τη συγκεκριμένη λειτουργία.

**int insertAge():** Η συνάρτηση αυτή ζητά επαναληπτικά από το χρήστη να εισάγει την ηλικία του παίκτη μέχρις ότου δώσει ηλικία μεγαλύτερη ή ίση του 18.

**string insertGender():** Η συνάρτηση αυτή ζητά επαναληπτικά από το χρήστη να εισάγει το φύλο του παίκτη, μέχρις ότου δώσει φύλο “male” ή “female”.

**string insertTeam():** Η συνάρτηση αυτή ζητά επαναληπτικά από το χρήστη να εισάγει την ομάδα του παίκτη, μέχρις ότου δώσει ομάδα “blue” ή “red”.