

## Μέρος Δ

---

### Ημέρα ψηφοφορίας του Masterchef

### 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)

[Προσομοίωση του παιχνιδιού Masterchef-  
Μοντελοποίηση ημέρας ψηφοφορίας]

## Περιεχόμενα

Περιγραφή προβλήματος .....	3
Ανάλυση Κλάσεων .....	3
Κλάση Vote .....	3
Μεταβλητές .....	3
Συναρτήσεις .....	3
Κλάση Voting .....	4
Μεταβλητές .....	4
Συναρτήσεις .....	4

## Περιγραφή προβλήματος

Το ζητούμενο σε αυτό το παραδοτέο είναι να μπορέσουμε να μοντελοποιήσουμε την ημέρα ψηφοφορίας του δημοφιλούς παιχνιδιού MasterChef. Μια τέτοια ημέρα περιλαμβάνει αρχικά τη συμμετοχή των μελών της ομάδας σε ένα διαγωνισμό ασυλίας, από όπου προκύπτει ένας νικητής ο οποίος έχει, εκτός από ασυλία, πλεονέκτημα και στον αριθμό των ψήφων που μπορεί να χρησιμοποιήσει κατά τη διαδικασία της ψηφοφορίας. Κατά την ψηφοφορία, κάθε παίκτης μπορεί να ψηφίσει οποιονδήποτε από τους συμπαίκτες του, εκτός από αυτόν που έχει ασυλία. Στο τέλος αυτής, τα αποτελέσματα συγκεντρώνονται και τίθενται ως υποψήφιοι οι δύο παίκτες με τις περισσότερες ψήφους. Από αυτούς, αποχωρεί εκείνος με τη μικρότερη τεχνική κατάρτιση.

Για την πραγματοποίηση της παραπάνω προσομοίωσης καλούμαστε να υλοποιήσουμε τις κλάσεις **Vote** και **Voting**. Η κλάση **Vote** προσομοιώνει μία ψήφο ενώ η **Voting** τη συνολική διαδικασία.

## Ανάλυση Κλάσεων

### Κλάση Vote

Η κλάση αυτή υλοποιείται στο .h file καθώς η υλοποίηση της λειτουργικότητας της είναι αρκετά σύντομη και τετριμμένη. Όλες οι μεταβλητές έχουν οριστεί **private**, ενώ όλες οι συναρτήσεις **public**.

### Μεταβλητές

- **string voted**: Το όνομα του υποψηφίου για αποχώρηση.
- **string reason**: Ο λόγος για τον οποίον ψηφίστηκε αυτός ο παίκτης.

### Συναρτήσεις

- **Vote()**: Ο χωρίς ορίσματα constructor. Αρχικοποιεί τις μεταβλητές voted και reason ίσα με ''.
- **Vote(string v , string r)**: Δέχεται ως ορίσματα το όνομα του υποψηφίου και τον λόγο αποχώρησης και με βάση αυτά θέτει τις τιμές στις αντίστοιχες μεταβλητές.
- **string getVoted()**: Επιστρέφει το όνομα του υποψηφίου.
- **string getReason()**: Επιστρέφει τον λόγο αποχώρησής του.
- **void setVoted(string v)**: Θέτει το όνομα του υποψηφίου ίσο με v .
- **void setReason(string r)**: Θέτει τον λόγο αποχώρησής του ίσο με r.

- **void status():** Εμφανίζει τα στοιχεία της ψήφου, δηλαδή το όνομα του υποψηφίου και τον λόγο για τον οποίο ψηφίστηκε.

## Κλάση Voting

Όλες οι μεταβλητές και οι συναρτήσεις αυτής της κλάσης ορίζονται ως **static**. Επίσης το scope όλων είναι **private**, εκτός από αυτό της συνάρτησης **votingProcess(Team &team)** που είναι **public**. Οι συναρτήσεις στο σύνολό τους έχουν οριστεί στο **.h file** και έχουν υλοποιηθεί στο **.cpp file**.

### Μεταβλητές

- **vector<Vote> votes:** Οι ψήφοι που έχουν συγκεντρωθεί στην τρέχουσα διαδικασία ψηφοφορίας.
- **map<string,int> results:** Ο πίνακας που διαθέτει το όνομα του παίκτη και τις ψήφους που έχει συγκεντρώσει στην τρέχουσα διαδικασία ψηφοφορίας.

### Συναρτήσεις

- **void votingProcess(Team &team):** Είναι η συνάρτηση που προσομοιώνει τη διαδικασία. Χρησιμοποιεί όλες τις βοηθητικές συναρτήσεις που αναλύονται παρακάτω. Αρχικά ελέγχεται το αν υπάρχει μόνο ένας ή δύο παίκτες διαθέσιμοι στην ομάδα που ψηφίζει. Αν αυτό συμβαίνει, τότε η **specialConditions()** θα έχει ήδη διαχειριστεί την κατάσταση και άρα η **votingProcess()** επιστρέφει. Σε διαφορετική περίπτωση, καλείται η **playerVote()** για κάθε έναν από τους παίκτες που έχουν απομείνει στην ομάδα. Αμέσως μετά, εμφανίζονται οι λεπτομέρειες κάθε ψήφου που δόθηκε και αφού ολοκληρωθεί η συμπλήρωση του **results** παρουσιάζεται η τελική του μορφή. Έπειτα, καλείται δύο φορές η **setCandidate()** προκειμένου να αποθηκευτούν οι δύο υποψήφιοι προς αποχώρηση, ενώ τέλος καλείται η **kickPlayer()** προκειμένου να γίνει η αποχώρηση και σβήνονται όλα τα στοιχεία των **votes** και **results** ώστε να μπορούν να χρησιμοποιηθούν για επόμενη ψηφοφορία.
- **bool specialConditions(Team &team):** Δέχεται ως όρισμα την ομάδα που ψηφίζει και εξετάζει τον αριθμό των παικτών της. Αν η ομάδα διαθέτει έναν ή δύο παίκτες επιστρέφει **true**. Στην πρώτη περίπτωση, πριν επιστρέψει, εμφανίζει μήνυμα σφάλματος, ώστε να ενημερώσει ότι δε μπορεί ο τελευταίος παίκτης να αποχωρίσει από την ομάδα. Στη δεύτερη περίπτωση, πριν επιστρέψει, διώχνει από το παιχνίδι εκείνον τον παίκτη, από τους δύο που έχουν μείνει, που έχει χάσει την ασυλία. Αν η ομάδα διαθέτει περισσότερους από δύο παίκτες τότε η συνάρτηση απλά επιστρέφει **false**.

- **void playerVote(Player\* p, int i):** Δέχεται ως όρισμα την αναφορά στο πρώτο στοιχείο του πίνακα των παικτών της ομάδας που ψηφίζει και τη θέση στον παραπάνω πίνακα του παίκτη που αυτή τη στιγμή καλείται να καταθέσει την ψήφο του. Ο συγκεκριμένος παίκτης ψηφίζει τόσες φορές, όσες και οι ψήφοι που διαθέτει, ενώ κάθε φορά που το κάνει, επιλέγει να ψηφίσει έναν συμπαίκτη του που δε διαθέτει ασυλία, βασιζόμενος σε έναν από τους παρακάτω λόγους:
  - α) Ο παίκτης που ψηφίζει είναι πολύ καλός και άρα απειλή για τον ίδιο στο παιχνίδι.
  - β) Ο παίκτης που ψηφίζει είναι πολύ κακός και άρα βάρος για την ομάδα του στο παιχνίδι.
  - γ) Ο παίκτης που ψηφίζει δε του είναι συμπαθής.
 Η επιλογή μεταξύ των τριών γίνεται τυχαία και ανάλογα με το αν είναι η α), β) ή γ) καλείται αντίστοιχα η συνάρτηση **goodPlayer()**, **badPlayer()** ή **notPopular()**. Ο παίκτης που ψηφίστηκε και ο λόγος για τον οποίο ψηφίστηκε αποθηκεύονται σε δύο τοπικές μεταβλητές οι οποίες θα περαστούν ως ορίσματα στον constructor του αντικειμένου **Vote** που δημιουργείται. Αυτό το αντικείμενο αντιγράφεται και προστίθεται στο vector **votes** της κλάσης. Να σημειωθεί ότι έχει προστεθεί και ένας επιπλέον (περιττός σε αυτή τη φάση) έλεγχος, ώστε να αποφευχθεί η εισαγωγή μιας ψήφου κενού ονόματος έπειτα από λανθασμένο χειρισμό των παραπάνω συναρτήσεων.
- **void kickPlayer(Team &team, string c1, string c2):** Δέχεται ως όρισμα την ομάδα που ψηφίζει καθώς και τα ονόματα των δύο παικτών που έχουν τεθεί ως υποψήφιοι προς αποχώρηση. Βρίσκει τα αντικείμενα τύπου **Player** που αντιστοιχούν σε αυτούς τους υποψηφίους και συγκρίνει την τεχνική τους κατάρτιση. Ο παίκτης με τη μικρότερη τεχνική κατάρτιση αποχωρεί από το παιχνίδι.
- **string getCandidate():** Η συνάρτηση αυτή διατρέχει τον πίνακα **results** ώστε να βρει το όνομα του παίκτη που έχει συγκεντρώσει τις περισσότερες ψήφους. Μόλις τον βρει, αποθηκεύει το όνομά του, τον διαγράφει από τον πίνακα **results** και επιστρέφει το όνομα του. Η διαγραφή επιλέχθηκε ώστε με μία δεύτερη κλήση να μπορεί να επιστρέψει τον δεύτερο παίκτη με τις περισσότερες ψήφους. Αν δύο ή περισσότεροι παίκτες έχουν τον ίδιο αριθμό ψήφων, επιλέγεται τυχαία ένας ως υποψήφιος.
- **string notPopular(Player\* p, int i):** Δέχεται ως όρισμα την αναφορά στο πρώτο στοιχείο του πίνακα με τους παίκτες της ομάδας που ψηφίζει καθώς και τη θέση σε αυτόν τον πίνακα του παίκτη που αυτή τη στιγμή καλείται να καταθέσει την ψήφο του. Προσομοιώνει στην ουσία την ψήφο του i παίκτη, εξασφαλίζοντας ότι θα ψηφίσει εκείνον τον συμπαίκτη του που είναι λιγότερο δημοφιλής και

ταυτόχρονα δεν έχει ασυλία. Στο τέλος προστίθεται και ένας έλεγχος (περιττός σε αυτή τη φάση) ώστε να αποφευχθεί το να βρεθούμε εκτός ορίων του πίνακα που περιέχει τους παίκτες της ομάδας, σε περίπτωση που κάποιος καλέσει αυτή τη συνάρτηση σε λάθος σημείο του προγράμματος.

- **string badPlayer(Player\* p, int i):** Δέχεται ως όρισμα την αναφορά στο πρώτο στοιχείο του πίνακα με τους παίκτες της ομάδας που ψηφίζει καθώς και τη θέση σε αυτόν τον πίνακα του παίκτη που αυτή τη στιγμή καλείται να καταθέσει την ψήφο του. Προσομοιώνει στην ουσία την ψήφο του *i* παίκτη, εξασφαλίζοντας ότι θα ψηφίσει εκείνον τον συμπαίκτη του που είναι **λιγότερο ικανός** και ταυτόχρονα δεν έχει ασυλία. Στο τέλος προστίθεται και ένας έλεγχος (περιττός σε αυτή τη φάση) ώστε να αποφευχθεί το να βρεθούμε εκτός ορίων του πίνακα που περιέχει τους παίκτες της ομάδας, σε περίπτωση που κάποιος καλέσει αυτή τη συνάρτηση σε λάθος σημείο του προγράμματος.
- **string goodPlayer(Player\* p, int i):** Δέχεται ως όρισμα την αναφορά στο πρώτο στοιχείο του πίνακα με τους παίκτες της ομάδας που ψηφίζει καθώς και τη θέση σε αυτόν τον πίνακα του παίκτη που αυτή τη στιγμή καλείται να καταθέσει την ψήφο του. Προσομοιώνει στην ουσία την ψήφο του *i* παίκτη, εξασφαλίζοντας ότι θα ψηφίσει εκείνον τον συμπαίκτη του που είναι **περισσότερο ικανός** και ταυτόχρονα δεν έχει ασυλία. Στο τέλος προστίθεται και ένας έλεγχος (περιττός σε αυτή τη φάση) ώστε να αποφευχθεί το να βρεθούμε εκτός ορίων του πίνακα που περιέχει τους παίκτες της ομάδας, σε περίπτωση που κάποιος καλέσει αυτή τη συνάρτηση σε λάθος σημείο του προγράμματος.