

# HY-150 2023

2<sup>η</sup> Σειρά Ασκήσεων  
10/03/2023 - 24/03/2023

## Θέμα 1ο: Player [25%]

Υλοποιήστε την κλάση player. Η κλάση player θα έχει τις εξής μεταβλητές

- Name: το όνομα του παίχτη
- Alive: εάν είναι ακόμα στο παιχνίδι ή όχι
- Role: τον ρόλο του, γκάνγκστερ, γιατρός ή πολίτης
- CurrentVotes: τον αριθμό των ψήφων που έχει στην τρέχουσα ψηφοφορία

Για όλες τις παραπάνω μεταβλητές υλοποιήστε setters και getters.

Επίσης, υλοποιήστε μία συνάρτηση vote, η οποία θα εκπληρώνει την διαδικασία ψηφοφορίας αυτού του παίχτη.

Τέλος, υλοποιήστε δύο constructors με υπογραφές

- Player(string Name , bool isAlive, string Role)
- Player()

*Σημείωση: τα παραπάνω είναι τα απαραίτητα που ζητάμε, μπορείτε να εμπλουτίσετε αυτή την κλάση όσο θέλετε.*

## Θέμα 2ο: Rounds [15%]

Υλοποιήστε μία κλάση Round η οποία θα κρατάει πληροφορία για τον εκάστοτε γύρο. Συγκεκριμένα πρέπει να περιέχει:

- Ποιος παίχτη αποχώρησε λόγο του γκανγκστερ, κενό εάν δεν αποχώρησε κανείς.
- Τον αριθμό του γύρου (1ος, 2ος κτλπ)
- Ποιος παίχτη αποχώρησε λόγο της ψηφοφορίας, κενό εάν δεν αποχώρησε κανείς

Στο τέλος του κάθε γύρου θα πρέπει να αποθηκεύεται τις παραπάνω πληροφορίες σε ένα καινούριο στιγμιότυπο (instance) της κλάσης round.

*Σημείωση: τα παραπάνω είναι τα απαραίτητα που ζητάμε, μπορείτε να εμπλουτίσετε αυτή την κλάση όσο θέλετε.*

### Θέμα 3ο: Player & Rounds Usage [25%]

Αναβαθμίστε το πρόγραμμα χρησιμοποιώντας τις καινούριες κλάσεις.

Μερικά παραδείγματα,

- Όπου εκτυπώνεται το ID του κάθε παίχτη θα πρέπει να εκτυπώνεται και το username
- Αντί να χρησιμοποιείτε `vector<int>` με το ID του κάθε παίχτη χρησιμοποιείτε `vector<Player>`
- Χρησιμοποιείτε την συνάρτηση `vote` για την ψήφο ενός παίχτη.

Τα παραπάνω είναι μερικά παραδείγματα. Ο στόχος σας σε αυτό το ερώτημα είναι να ανιχνεύσετε όσο περισσότερες εφαρμογές έχουν οι παραπάνω κλάσεις στο πρόγραμμά σας και να τις χρησιμοποιήσετε.

### Θέμα 4ο: Multiple Files [10%]

Υλοποιήστε τις κλάσεις `Player` και `Round` σε ξεχωριστά αρχεία `.cpp`. Οι δηλώσεις τους μπορεί να είναι σε ένα συγκεντρωτικό `.h` ή σε ξεχωριστά `.h`, αλλά η υλοποίηση τους πρέπει να είναι σε ξεχωριστά αρχεία `.cpp`.

### Θέμα 5ο: Input [15%]

Το πρόγραμμά σας θα πρέπει να γίνεται initialize από αρχείο. Στην αρχή του παιχνιδιού θα πρέπει να ανοίγετε ένα αρχείο με όνομα `"Players.txt"` στο οποίο σε κάθε γραμμή υπάρχει το username του κάθε παίχτη (`player1`, `IamnotTheGangster`, `Winner99` κτλ...), και ο ρόλος του, χωρισμένα από ένα κενό. Στην συνέχεια θα πρέπει να αρχικοποιείτε το πρόγραμμά σας με την πληροφορία από αυτό το αρχείο. Θεωρήστε ότι το αρχείο δεν έχει διπλότυπα usernames και έχει πάντα ακριβώς έναν γκάνγκστερ και έναν γιατρό.

### Θέμα 6ο: Output [10%]

Στο τέλος του παιχνιδιού θα πρέπει να δημιουργείτε ένα καινούριο αρχείο `"TownOfSalem_output.txt"` στο οποίο θα γράφετε όλη την πληροφορία που αποθηκεύσατε στα στιγμιότυπα της κλάσης `rounds`.

### Παραδοτέα

Παραδώστε όλα τα αρχεία `.h` που χρησιμοποιήσατε, **τρία** αρχεία `.cpp` (ενδεικτικά ονόματα: `Player.cpp`, `Round.cpp`, `Assignment2.cpp`) και το `CMakeLists.txt` που παράγει το τελικό εκτελέσιμο. Προσθέστε όλα τα παραπάνω αρχεία σε ένα `.zip` archive με όνομα `Assignment2_AMXXXX.zip`.