

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών
Υπολογιστών

Αντικειμενοστρεφής Προγραμματισμός

2ο Εξάμηνο Σπουδών

Εργασία Α' - Διάγραμμα UML

Μια εργασία των φοιτητών:

Κωφοκώτσου Ηλία

AEM: 9380

Σκαπέτη Χρήστου

AEM: 9378



Ακαδημαϊκό Έτος 2017-2018

Περιγραφή του Προβλήματος

Στην παρούσα εργασία ζητήθηκε από τους φοιτητές να μοντελοποιήσουν το δημοφιλές τηλεοπτικό παιχνίδι- διαγωνισμό μαγειρικής Masterchef. Δόθηκε ένα κείμενο το οποίο περιέγραφε τις βασικές λειτουργίες του παιχνιδιού, από την βασική δομή των ομάδων έως τις διάφορες λειτουργίες που επιτελεί ο κάθε παίκτης κατά την διάρκεια του παιχνιδιού. Επιπλέον, το δοθέν κείμενο περιέγραφε την βασική ιδέα του παιχνιδιού, η οποία περιστρέφεται γύρω από διαγωνισμούς στους οποίους συμμετέχουν οι παίκτες και οι ομάδες τους, προκειμένου μέσα από την διαδικασία διαδοχικών διαγωνισμών να προκύψει κάποιος παίκτης ο οποίος θα αποχωρήσει οριστικά από το παιχνίδι και την επανάληψη της διαδικασίας αυτή εως ότου απομείνει ένας μοναδικός παίκτης, ο οποίος θα στεφθεί και ο νικητής του παιχνιδιού. Το ζητούμενο της άσκησης είναι να δημιουργηθεί ένα διάγραμμα κλάσεων με την χρήση της γλώσσας UML, το οποίο να περιγράφει κατά το δυνατόν ακριβέστερα το παιχνίδι αυτό, αφού θα περιέχει όλες τις απαιτούμενες ενέργειες και χαρακτηριστικά των κλάσεων.

Διάγραμμα UML Masterchef 2018



Λίστα Βασικών Οντοτήτων

Κλάση

Player

Χαρακτηριστικά

- Name: String
- Gender: String {Gender = "Male" OR "Female"}
- Age: Unsigned Integer
- Occupation: String
- Skill: Unsigned Integer {Skill < = 80}
- Fatigue: Unsigned Integer {Fatigue < = 100}
- Popularity: Unsigned Integer {Popularity < = 100}
- Victories: Unsigned Integer

Μέθοδοι

- Player()
- Work (Skill: Unsigned Integer, Fatigue: Unsigned Integer)
- Socialize(Popularity: Unsigned Integer)
- EndOfWeekActivities(Skill: Unsigned Integer, Fatigue: Unsigned Integer)
- ParticipateInCompetition(Fatigue: Unsigned Integer, Type: String)
- Leave()

Σχόλια

Δημιουργήσαμε μια κλάση Player, σε στιγμιότυπα της οποίας θα αποθηκεύονται όλα τα στοιχεία των παικτών, δηλαδή το όνομα (*Name*), το φύλο (*Gender*), που μπορεί να είναι αρσενικό ή θηλυκό (*Male / Female*), την ηλικία (*Age*), το επάγγελμα εκτός διαγωνισμού (*Occupation*), την ικανότητα (*Skill*) που είναι ένας αριθμός - ποσοστό από το 0 έως το 80, την κούραση (*Fatigue*) που είναι ένας αριθμός - ποσοστό από το 0 έως το 100, την δημοφιλία (*Popularity*) που είναι ένας αριθμός - ποσοστό από το 0 έως το 100, και τέλος τις νίκες (*Victories*).

Επιπλέον, στην κλάση βάλαμε και έναν αριθμό μεθόδων. Αρχικά, βάλαμε μια συνάρτηση αρχικών συνθηκών (*Player*) η οποία θα αρχικοποιεί όλα τα στοιχεία του κάθε παίκτη που αναφέραμε παραπάνω στις τιμές που καθορίζονται από τα δεδομένα του προβλήματος. Επιπρόσθετα, βάλαμε μεθόδους οι οποίες θα επιτελούν λειτουργίες όπως η εξάσκηση κατά την διάρκεια της εβδομάδας (*Work*),

η συναναστροφή με τους υπόλοιπους παίκτες (*Socialize*), η επιλογή της απασχόλησης του καθενός για την μια ελεύθερη μέρα στο τέλος κάθε εβδομάδας (*EndOfWeekActivities*), η συμμετοχή σε διαγωνισμούς (*ParticipateInCompetition*) και η αποχώρηση εξ ολοκλήρου από το παιχνίδι (*Leave*), με κάποια ορίσματα τα οποία θεωρήσαμε απαραίτητα για την λειτουργία της κάθε μεθόδου, χωρίς αυτό να είναι αποκλειστικό.

Κλάση

Team

Χαρακτηριστικά

- Colour: String {Colour = "Blue" OR "Red"}
- PlayerNum: Unsigned Integer
- Players[PlayerNum]: Array of Player
- Victories: Unsigned Integer

Μέθοδοι

- Team()
- ParticipateInCompetition (Comp: Competition)
- ParticipateInVoting (Players[PlayerNum]:Array of Player)

Σχόλια

Σύμφωνα με το κείμενο, δημιουργήσαμε μια κλάση η οποία καλύπτει τις προγραμματιστικές ανάγκες κάθε ομάδας (*Team*). Κάθε ομάδα πρέπει να έχει ένα χρώμα (*Colour*), μπλε ή κόκκινο (*Red / Blue*), αποτελείται από ένα θετικό αριθμό παικτών (*PlayerNum*), τα στοιχεία των οποίων καταγράφονται σε έναν πίνακα (*Players*) με στοιχεία αντικείμενα της κλάσης *Player*. Επιπλέον, κάθε ομάδα έχει έναν αριθμό από νίκες (*Victories*) σε διαγωνισμούς του παιχνιδιού.

Κάθε ομάδα μπορεί να συμμετάσχει σε έναν ομαδικό διαγωνισμό (*ParticipateInCompetition*). Επιπλέον, στην περίπτωση που η ομάδα ηττηθεί σε ομαδικό διαγωνισμό, προβαίνει σε μια ψηφοφορία (*ParticipateInVoting*), μέσα από την οποία προκύπτουν τα μέλη της που θα διαγωνιστούν στον διαγωνισμό

αποχώρησης (*Disqualification Competition*), που αναφέρεται παρακάτω. Επιπλέον, βάλαμε και μια συνάρτηση αρχικών συνθηκών, η οποία θα αρχικοποιεί τα χαρακτηριστικά της κλάσης.

Κλάση

Trophy

Χαρακτήριστικά

- Name: String
- Personal: Boolean
- Type: String {Type = "Immunity" OR "Food" OR "Excursion" }

Μέθοδοι

-

Σχόλια

Δημιουργήσαμε μια κλάση (*Trophy*) που να καλύπτει τις ανάγκες του κάθε βραβείου που θα έχει ως έπαθλο ο κάθε διαγωνισμός. Έτσι, κάθε βραβείο έχει ένα όνομα (*Name*), μπορεί να είναι ατομικό ή όχι (*Personal*), ενώ έχει και έναν τύπο, ο οποίος μπορεί να είναι ασυλία (*Immunity*), κάποιο φαγητό (*Food*) ή κάποια εκδρομή (*Excursion*).

Κλάση

Competition

Χαρακτηριστικά

- Code: Unsigned Integer

- Type: String {Type = " Team" OR "Immunity" OR "Disqualification"}

Μέθοδοι

- Competition()

Σχόλια

Δημιουργήσαμε μια κλάση για τους διαγωνισμούς (*Competition*), η οποία μέσω κληρονομικότητας μεταδίδει τα χαρακτηριστικά της, δηλαδή έναν αύξοντα κωδικό (*Code*) και έναν τύπο (*Type*), ο οποίος μπορεί να είναι Ομαδικός, Ασυλίας ή Αποχώρησης (*Team / Immunity / Disqualification*), σε τρεις κλάσεις, οι οποίες παρουσιάζονται παρακάτω, και οι οποίες περιγράφουν κάθε τύπο διαγωνισμού ξεχωριστά.

Επιπλέον, βάλαμε και μια συνάρτηση αρχικών συνθηκών, ώστε να αρχικοποιούνται τα παραπάνω χαρακτηριστικά.

Κλάση

TeamCompetition

Χαρακτήριστικά

- ChallengeCount: Unsigned Integer
- WinnerTeam: Team

Μέθοδοι

- TeamCompetition()

Σχόλια

Δημιουργήσαμε μια κλάση για τους ομαδικούς διαγωνισμούς (*TeamCompetition*), οι οποίοι περιλαμβάνουν έναν αριθμό από δοκιμασίες (*ChallengeCount*) και θα έχουν στο τέλος μια νικήτρια ομάδα (*WinnerTeam*).

Επιπλέον, δημιουργήσαμε μια συνάρτηση αρχικών συνθηκών προκειμένου να αρχικοποιείται ο αριθμός των δοκιμασιών του διαγωνισμού, οι οποίες περιγράφονται από την κλάση που ακολουθεί.

Κλάση

TeamChallenge

Χαρακτηριστικά

- ParticipantsNum: Unsigned Integer {Participants < = 5}
- BlueParticipants[ParticipantsNum]: Array of Player
- RedParticipants[ParticipantsNum]: Array of Player
- Duration: Unsigned Integer

Μέθοδοι

- TeamChallenge()

Σχόλια

Δημιουργήσαμε μια κλάση για τις ανάγκες κάθε δοκιμασίας των ομαδικών διαγωνισμών (*Team Challenge*). Κάθε δοκιμασία έχει έναν αριθμό από συμμετέχοντες (*ParticipantsNum*), μικρότερο ή ίσο από 5, για κάθε ομάδα ξεχωριστά, οι οποίοι καταγράφονται σε πίνακες στιγμιotypών της κλάσης *Player* (*BlueParticipants/RedParticipants*).

Τέλος, βάλαμε και μια συνάρτηση αρχικών συνθηκών, για να αρχικοποιεί αυτά τα στοιχεία.

Κλάση

ImmunityCompetition

Χαρακτηριστικά

- ChallengeCount: Unsigned Integer
- Winner: Player

Μέθοδοι

- ImmunityCompetition()

Σχόλια

Δημιουργήσαμε μια κλάση για τους διαγωνισμούς ασυλίας(*ImmunityCompetition*), οι οποίοι περιλαμβάνουν έναν αριθμό από δοκιμασίες (*ChallengeCount*) και θα έχουν στο τέλος έναν νικητή (*Winner*).

Επιπλέον, δημιουργήσαμε μια συνάρτηση αρχικών συνθηκών προκειμένου να αρχικοποιείται ο αριθμός των δοκιμασιών του διαγωνισμού, οι οποίες περιγράφονται από την κλάση που ακολουθεί.

Κλάση

ImmunityChallenge

Χαρακτηριστικά

- ParticipantsNum: Unsigned Integer {Participants < = 5}
- Participants[ParticipantsNum]: Array of Player
- Duration: Unsigned Integer

Μέθοδοι

- ImmunityChallenge()

Σχόλια

Δημιουργήσαμε μια κλάση για τις ανάγκες κάθε δοκιμασίας των διαγωνισμών (*ImmunityChallenge*). Κάθε δοκιμασία έχει έναν αριθμό από συμμετέχοντες (*ParticipantsNum*), μικρότερο ή ίσο από 3, που αποθηκεύονται σε έναν πίνακα με στοιχεία στιγμιότυπα της κλάσης *Player*. Επιπλέον, κάθε δοκιμασία έχει από μια διάρκεια (*Duration*), ενώ δεν θεωρήσαμε σωστό το ότι κάθε δοκιμασία, μόνη της έχει έναν νικητή, καθώς το όλο νόημα του διαγωνισμού ασυλίας είναι η ανάδειξη ενός μοναδικού νικητή στο τέλος του διαγωνισμού αυτού.

Τέλος, βάλαμε και μια συνάρτηση αρχικών συνθηκών, για να αρχικοποιεί τα στοιχεία της κάθε δοκιμασίας.

Κλάση

`DisqualificationCompetition`

Χαρακτηριστικά

- `ChallengeCount`: Unsigned Integer
- `Loser`: *Player*

Μέθοδοι

- `DisqualificationCompetition()`

Σχόλια

Δημιουργήσαμε μια κλάση για τους διαγωνισμούς αποχώρησης (*DisqualificationCompetition*), οι οποίοι περιλαμβάνουν έναν αριθμό από δοκιμασίες (*ChallengeCount*), οι οποίες περιγράφονται από την αντίστοιχη κλάση που αναφέρεται παρακάτω, και θα έχουν στο τέλος έναν χαμένο (*Loser*), ο οποίος και θα αποχωρήσει από το παιχνίδι.

Επιπλέον, δημιουργήσαμε μια συνάρτηση αρχικών συνθηκών προκειμένου να αρχικοποιείται ο αριθμός των δοκιμασιών του διαγωνισμού.

Κλάση

DisqualificationChallenge

Χαρακτηριστικά

- ParticipantsNum: Unsigned Integer {Participants < = 5}
- Participants[ParticipantsNum]: Array of Player
- Duration: Unsigned Integer

Μέθοδοι

- DisqualificationChallenge()

Σχόλια

Δημιουργήσαμε μια κλάση για τις ανάγκες κάθε δοκιμασίας των διαγωνισμών αποχώρησης (*DisqualificationChallenge*). Κάθε δοκιμασία έχει έναν αριθμό από συμμετέχοντες (*ParticipantsNum*), μικρότερο ή ίσο από 3, οι οποίοι καταγράφονται σε έναν πίνακα στιγμιότυπων της κλάσης *Player* (*Participants*). Επιπλέον, κάθε δοκιμασία έχει από μια διάρκεια (*Duration*), ενώ δεν θεωρήσαμε ότι κάθε δοκιμασία θα έπρεπε να έχει και από έναν χαμένο, καθώς το όλο νόημα του διαγωνισμού ασυλίας είναι να προκύψει ένας μοναδικός χαμένος, ο οποίος έπειτα θα αποχωρήσει από το παιχνίδι.

Τέλος, βάλαμε και μια συνάρτηση αρχικών συνθηκών, για να αρχικοποιεί τα στοιχεία της κάθε δοκιμασίας.

Κλάση

Vote

Χαρακτηριστικά

- Voted: Player
- Reason: String

Μέθοδοι

-

Σχόλια

Θεωρήσαμε σωστό να δημιουργήσουμε και μια κλάση η οποία θα περιέγραφε τις ψήφους του καθενός (*Vote*). Κάθε ψήφος περιέχει έναν παίκτη ο οποίος ψηφίζεται (*Voted*), καθώς και τον λόγο για τον οποίο ψηφίζεται (*Reason*).

Κλάση

Judge

Χαρακτηριστικά

-

Μέθοδοι

- `AnnounceVoteResults()`
- `ChooseDisqualified()`

Σχόλια

Τέλος, θεωρήσαμε σωστό να δημιουργήσουμε μια κλάση, η οποία να καλύπτει τις ανάγκες των κριτών (*Judge*). Από τα δεδομένα του προβλήματος, δίδεται έμφαση στα πρόσωπα των κριτών στις περιπτώσεις που πρέπει να ανακοινώσουν τα αποτελέσματα της ψηφοφορίας (*AnnounceVoteResults*) και πρέπει να αποφασίσουν ποιος θα αποχωρήσει από το παιχνίδι (*ChooseDisqualified*).

Επιπλέον Παρατηρήσεις

- Στο ανωτέρω διάγραμμα UML, έχουμε προσθέσει επίσης κάποιες συσχετίσεις, οι οποίες, με τις αναγραφόμενες πληθικότητες και τα ρήματα, αναδεικνύουν και άλλους περιορισμούς, καθώς και λειτουργίες - μεθόδους αναφορικά με τις προαναφερθείσες κλάσεις. Αυτές δεν καλύφθηκαν στις παραπάνω περιγραφές, γιατί κατά την κρίση μας δεν ήταν απαιτούμενο από την εκφώνηση του προβλήματος. Είναι όμως ένα σημαντικό κομμάτι της μοντελοποίησης του προβλήματος και για αυτό θεωρούμε ότι θα έπρεπε να δοθεί η ίδια σημασία στις συσχετίσεις αυτές με ό,τι γράφτηκε παραπάνω.
- Από την εκφώνηση του προβλήματος ζητήθηκε ρητά να καλυφθούν στο διάγραμμα UML μόνο οι περιπτώσεις που καλύπτονταν στο δοθέν κείμενο. Θα θέλαμε να σημειώσουμε ότι είναι πιθανόν η θέληση μας να ακολουθήσουμε αυτήν την οδηγία να μας οδήγησε σε κάποιες παραλείψεις, αν και θεωρούμε πως κάτι τέτοιο δεν συνέβη.