

ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

***Master Chef 2018
Μέρος Γ'
Τετάρτη 09/05/2018***

Μιχαήλ Μηναδάκης
AEM: 8858
Email: thejokergr@hotmail.gr

Άρης-Ελευθέριος Παπαγγέλης
AEM: 8883
Email: aris.papagelis@gmail.com

Στέφανος Παπαδάμ
AEM: 8885
Email: stefanospapadam@gmail.com



Illustration 1: Ετοιμοπόλεμοι οι παίκτες μας!

Πρόβλημα προς επίλυση

Καλούμαστε να αναπτύξουμε ένα πρόγραμμα, το οποίο να μοντελοποιεί τους διάφορους διαγωνισμούς του τηλεπαιχνιδιού MasterChef.

Περιγραφή Υλοποίησης

Κλάση *Competition*

Η συγκεκριμένη κλάση αναλύεται στο εξής αρχείο: *Competition.h*

Competition.h:

Στο αρχείο αυτό προδιαγράφουμε τις εξής μεθόδους (όλες οι μέθοδοι της κλάσης υλοποιούνται σε αυτό το αρχείο, ως inline συναρτήσεις) :

- *Competition()*: συνάρτηση αρχικών συνθηκών χωρίς ορίσματα.
- *Competition(int id, string name)*: συνάρτηση αρχικών συνθηκών με παραμέτρους.
- *~Competition()*: συνάρτηση τελικών συνθηκών.
- *int getId()*: συνάρτηση που επιστρέφει την τιμή της μεταβλητής id.
- *string getName()*: συνάρτηση που επιστρέφει την τιμή της μεταβλητής name.
- *string getWinner()*: συνάρτηση που επιστρέφει την τιμή της μεταβλητής winner.
- *void setId(int id)*: συνάρτηση που θέτει τιμή στη μεταβλητή id.
- *void setName(string name)*: συνάρτηση που θέτει τιμή στη μεταβλητή name.
- *string setWinner (string winner)*: συνάρτηση που θέτει τιμή στη μεταβλητή winner.
- *void status()*: συνάρτηση που εμφανίζει τις τιμές των μεταβλητών της κλάσης μαζί με κατάλληλα μηνύματα.

Κλάση *TeamCompetition*

Η συγκεκριμένη κλάση **κληρονομεί** την κλάση *Competition* και αναλύεται σε δύο αρχεία: *TeamCompetition.h* και *TeamCompetition.cpp*.

TeamCompetition.h:

Στο αρχείο αυτό προδιαγράφουμε τις εξής μεθόδους (οι getters και setters υλοποιούνται κιόλας σε αυτό το αρχείο, ως inline συναρτήσεις) :

- TeamCompetition(): συνάρτηση αρχικών συνθηκών χωρίς ορίσματα.
- TeamCompetition(int id,string name ,FoodAward foodAward): συνάρτηση αρχικών συνθηκών με παραμέτρους.
- ~TeamCompetition(): συνάρτηση τελικών συνθηκών.
- FoodAward getFoodAward():συνάρτηση που επιστρέφει την τιμή του αντικειμένου foodAward.
- Round getRounds(int i): συνάρτηση που επιστρέφει την τιμή του αντικειμένου rounds[i].
- void setFoodAward(FoodAward foodAward): συνάρτηση που θέτει τιμή στο αντικείμενο foodAward.
- void setRounds(Round r,int i): συνάρτηση που θέτει τιμή στο αντικείμενο rounds[i].
- void status(): συνάρτηση που εμφανίζει τις τιμές των μεταβλητών της κλάσης μαζί με κατάλληλα μηνύματα.
- int compete(Team &team1, Team &team2):συνάρτηση που δέχεται ως ορίσματα αναφορές στις δύο ομάδες που συμμετέχουν σε αγώνισμα και επιστρέφει κατάλληλη ακέραια τιμή (0 ή 1)ανάλογα με την ομάδα που έχασε.

TeamCompetition.cpp:

Σε αυτό το αρχείο γίνεται η υλοποίηση των παρακάτω συναρτήσεων:

- TeamCompetition(): Συνάρτηση αρχικών συνθηκών χωρίς ορίσματα η οποία αρχικοποιεί το αντικείμενο foodAward καλώντας την κενή συνάρτηση αρχικών συνθηκών FoodAward() της κλάσης FoodAward. Επίσης αρχικοποιεί τον πίνακα rounds που περιέχει αντικείμενα τύπου Round με μία for τριών επαναλήψεων καλώντας για κάθε θέση την κενή συνάρτηση αρχικών συνθηκών Round() της κλάσης Round.
- TeamCompetition(int id , string name,FoodAward foodAward):
Competition(id,name): Συνάρτηση αρχικών συνθηκών με ορίσματα.Οι μεταβλητές id και name αρχικοποιούνται μέσω του constructor της κλάσης Competition η οποία κληρονομείται. Το αντικείμενο foodAward

αρχικοποιείται μέσω του ορίσματος που δίνεται στη συνάρτηση και ο πίνακας rounds αρχικοποιείται όπως προηγουμένως.

- *~TeamCompetition()*: συνάρτηση τελικών συνθηκών η οποία εμφανίζει κατάλληλο μήνυμα όταν το αντικείμενο καταστρέφεται.
- *status()*: συνάρτηση η οποία εμφανίζει τις τιμές των μεταβλητών της κλάσης καλώντας τη συνάρτηση `Competition::status()` για τις μεταβλητές της βασικής κλάσης `Competition`, τη συνάρτηση `foodAward.status()` για τις μεταβλητές του αντικειμένου `foodAward`, το οποίο περιλαμβάνεται στην κλάση `TeamCompetition`, και τη συνάρτηση `rounds[i].status()` για κάθε θέση του πίνακα `rounds`.
- *compete(Team &team1,Team &team2)*: Αρχικά μηδενίζουμε τις μεταβλητές `wins1` και `wins2` που αντιπροσωπεύουν τις νίκες της κάθε ομάδας. Χρησιμοποιούμε μία `for` που επαναλαμβάνεται τρεις φορές (μία για κάθε γύρο του διαγωνισμού). Μέσα στην επανάληψη ορίζουμε ένα πίνακα τύπου `bool` (`selected1`) με θέσεις όσοι είναι οι παίκτες της πρώτης ομάδας και τον αρχικοποιούμε με τιμή `false` για κάθε θέση του. Η τιμή μίας θέσης του πίνακα γίνεται `true` όταν επιλέγεται να διαγωνιστεί ο παίκτης της συγκεκριμένης θέσης του πίνακα. Μηδενίζουμε τις μεταβλητές `meanTechnique1` και `meanFatigue1` οι οποίες θα μας δώσουν την μέση τεχνική και την μέση κόυραση των 5 παικτών που θα επιλεγθούν. Χρησιμοποιούμε μία `while` που τερματίζει όταν επιλεγθούν 5 τυχαίοι παίκτες της πρώτης ομάδας. Αρχικά επιλέγουμε έναν τυχαίο αριθμό ο οποίος είναι μικρότερος από τον αριθμό των παικτών της πρώτης ομάδας, ελέγχουμε αν έχει επιλεγθεί προηγουμένως ο συγκεκριμένος αριθμός, και αν όχι, επιλέγουμε τον παίκτη που βρίσκεται σε αυτή τη θέση για να διαγωνιστεί. Κάνουμε `true` την τιμή στην αντίστοιχη θέση του πίνακα `selected1`, προσθέτουμε την τεχνική του παίκτη στην `meanTechnique1` και την κόυραση του στην `meanFatigue1`. Τέλος, διαιρούμε δια του 5 τις `meanTechnique1` και `meanFatigue1` ώστε να βρούμε την μέση τεχνική και κόυραση των παιχτών τις ομάδας. Αντίστοιχα πράττουμε και για την δεύτερη ομάδα. Η ομάδα με το μεγαλύτερο `meanTechnique` είναι νικήτρια και αυξάνει τις νίκες της για το συγκεκριμένο `TeamCompetition` κατα 1 ενώ σε περίπτωση ισότητας των `meanTechnique` νικήτρια είναι η ομάδα με το μικρότερο `meanFatigue`. Αρχικοποιείται επίσης αντικείμενο τύπου `round` στην αντίστοιχη θέση `i` του πίνακα `rounds[i]`, με τα κατάλληλα ορίσματα. Μετά το πέρας της `for` η ομάδα που έχει τουλάχιστον 2 νίκες αυξάνει κατα 1 τις συνολικές τις νίκες της και αυξάνει τις προμήθειες της σύμφωνα με την κλάση `foodAward`. Τέλος, εκτυπώνουμε τα στοιχεία του διαγωνισμού και της νικητριάς ομάδας.

Κλάση *ImmunityCompetition*

Η συγκεκριμένη κλάση **κληρονομεί** την κλάση *Competition* και αναλύεται σε δύο αρχεία: *ImmunityCompetition.h* και *ImmunityCompetition.cpp*.

ImmunityCompetition.h:

Στο αρχείο αυτό προδιαγράφουμε τις εξής μεθόδους (οι getters και setters υλοποιούνται κιόλας σε αυτό το αρχείο, ως inline συναρτήσεις) :

- *ImmunityCompetition()*: συνάρτηση αρχικών συνθηκών χωρίς ορίσματα.
- *ImmunityCompetition(int id,string name ,ImmunityAward immunityAward)*: συνάρτηση αρχικών συνθηκών με παραμέτρους.
- *~ImmunityCompetition()*: συνάρτηση τελικών συνθηκών.
- *ImmunityAward getImmunityAward()*: συνάρτηση που επιστρέφει την τιμή του αντικειμένου *immunityAward*.
- *void setImmunityAward(ImmunityAward immunityAward)*: συνάρτηση που θέτει τιμή στο αντικείμενο *immunityAward*.
- *void status()*: συνάρτηση που εμφανίζει τις τιμές των μεταβλητών της κλάσης μαζί με κατάλληλα μηνύματα.
- *void compete(Team &team)*: συνάρτηση που δέχεται ως ορίσμα αναφορά της ομάδας που συμμετέχει στο αγώνισμα.

ImmunityCompetition.cpp:

Σε αυτό το αρχείο γίνεται η υλοποίηση των παρακάτω συναρτήσεων:

- *ImmunityCompetition()*: Συνάρτηση αρχικών συνθηκών χωρίς ορίσματα η οποία αρχικοποιεί το αντικείμενο *immunityAward* καλώντας την κενή συνάρτηση αρχικών συνθηκών *ImmunityAward()* της κλάσης *ImmunityAward*.
- *ImmunityCompetition(int id , string name,ImmunityAward immunityAward):Competition(id, name)*: Συνάρτηση αρχικών συνθηκών με ορίσματα. Οι μεταβλητές *id* και *name* αρχικοποιούνται μέσω του constructor της κλάσης *Competition* η οποία κληρονομείται. Το αντικείμενο *immunityAward* αρχικοποιείται μέσω του ορίσματος που δίνεται στη συνάρτηση.
- *~ImmunityCompetition()*: συνάρτηση τελικών συνθηκών η οποία εμφανίζει κατάλληλο μήνυμα όταν το αντικείμενο καταστρέφεται.

- status(): συνάρτηση η οποία εμφανίζει τις τιμές των μεταβλητών της κλάσης καλώντας τη συνάρτηση `Competition::status()` για τις μεταβλητές της βασικής κλάσης `Competition`, τη συνάρτηση `immunityAward.status()` για τις μεταβλητές του αντικειμένου `immunityAward` το οποίο περιλαμβάνεται στην κλάση `ImmunityCompetition`.
- compete(*Team &team*): Αρχικά μηδενίζουμε τη μεταβλητή `best` που αντιπροσωπεύει τον καλύτερο συνδυασμό τεχνικής κατάρτισης και μη κούρασης και θέτουμε -1 τη μεταβλητή `index` η οποία αντιπροσωπεύει τη θέση ενός στοιχείου του πίνακα. Στη συνέχεια με μία `for` ελέγχουμε ποιός παίκτης έχει τον καλύτερο συνδυασμό και αλλάζουμε αντίστοιχα τις τιμές των `best` και `index`. Όταν τελειώσει η `for` αυξάνουμε τις νίκες του παίκτη που κέρδισε κατά 1 και εκτυπώνουμε τα στοιχεία του νικητή και του διαγωνισμού.

Κλάση CreativityCompetition

Η συγκεκριμένη κλάση **κληρονομεί** την κλάση `Competition` και αναλύεται σε δύο αρχεία: `CreativityCompetition.h` και `CreativityCompetition.cpp`.

CreativityCompetition.h:

Στο αρχείο αυτό προδιαγράφουμε τις εξής μεθόδους (οι `getters` και `setters` υλοποιούνται κιόλας σε αυτό το αρχείο, ως `inline` συναρτήσεις) :

- *CreativityCompetition()*: συνάρτηση αρχικών συνθηκών χωρίς ορίσματα.
- *CreativityCompetition(int id, string name, ExcursionAward excursionAward)*: συνάρτηση αρχικών συνθηκών με παραμέτρους.
- *~CreativityCompetition()*: συνάρτηση τελικών συνθηκών.
- *ExcursionAward getExcursionAward()*: συνάρτηση που επιστρέφει την τιμή του αντικειμένου `excursionAward`.
- *void setExcursionAward(ExcursionAward excursionAward)*: συνάρτηση που θέτει τιμή στο αντικείμενο `excursionAward`.
- *void status()*: συνάρτηση που εμφανίζει τις τιμές των μεταβλητών της κλάσης μαζί με κατάλληλα μηνύματα.
- *void compete(Team &team1, Team &team2)*: συνάρτηση που δέχεται ως ορίσματα αναφορές στις δύο ομάδες που συμμετέχουν σε αγώνισμα και υπολογίζει το νικητή του διαγωνισμού.
- *static string ingredients[10]*: στατικός πίνακας 10 στοιχείων που περιέχει τα υλικά που χρησιμοποιούνται στον `CreativityCompetition`.

CreativityCompetition.cpp:

Σε αυτό το αρχείο γίνεται η υλοποίηση των παρακάτω συναρτήσεων:

- CreativityCompetition(): Συνάρτηση αρχικών συνθηκών χωρίς ορίσματα η οποία αρχικοποιεί το αντικείμενο excursionAward καλώντας την κενή συνάρτηση αρχικών συνθηκών ExcursionAward() της κλάσης ExcursionAward.
- CreativityCompetition(int id , string name,ExcursionAward excursionAward):Competition(id, name) Συνάρτηση αρχικών συνθηκών με ορίσματα. Οι μεταβλητές id και name αρχικοποιούνται μέσω του constructor της κλάσης Competition η οποία κληρονομείται. Το αντικείμενο excursionAward αρχικοποιείται μέσω του ορίσματος που δίνεται στη συνάρτηση.
- ~ExcursionCompetition(): συνάρτηση τελικών συνθηκών η οποία εμφανίζει κατάλληλο μήνυμα όταν το αντικείμενο καταστρέφεται.
- status(): συνάρτηση η οποία εμφανίζει τις τιμές των μεταβλητών της κλάσης καλώντας τη συνάρτηση Competition::status() για τις μεταβλητές της βασικής κλάσης Competition και τη συνάρτηση excursionAward.status() για τις μεταβλητές του αντικειμένου excursionAward το οποίο περιλαμβάνεται στην κλάση CreativityCompetition.
- compete(Team &team1,Team &team2): Αρχικά ορίζουμε έναν πίνακα τύπου Player μεγέθους total (συνολικός αριθμός των παικτών και των δύο ομάδων). Συμπληρώνουμε το πρώτο μισό του πίνακα με τους παίκτες της πρώτης ομάδας και το άλλο μισό με τους παίκτες της δεύτερης. Ύστερα, βρίσκουμε ποιός από τους παίκτες αυτούς έχει την καλύτερη τεχνική, καθώς και την θέση που αυτός βρίσκεται στον πίνακα. Ο παίκτης αυτός χρίζεται νικητής, εκτυπώνονται τα στοιχεία του και μεταβάλλονται η τεχνική του και η δημοφιλία του με βάση την εκφώνηση. Τέλος, εκτυπώνονται και τα στοιχεία του διαγωνισμού.

MAIN

main.cpp:

Σε αυτό το αρχείο γίνεται η υλοποίηση των παρακάτω συναρτήσεων:

- int main(): Απλά αρχικοποιούμε τον random number generator και καλούμε την menu()
- void menu(): Ζητείται από τον χρήστη να επιλέξει τι είδους μέρα είναι η σημερινή στο παιχνίδι, κανονική, ημέρα ομαδικού διαγωνισμού, ημέρα διαγωνισμού ασυλίας ή ημέρα διαγωνισμού δημιουργικότητας, και ανάλογα καλείται η αντίστοιχη συνάρτηση. Υπάρχει επίσης επιλογή εξόδου από το παιχνίδι

- `void normalDay()`: Υλοποιούνται οι καθημερινές δράσεις για τις δύο ομάδες, δηλαδή δουλειά, πρωινό, μεσημεριανό, συζήτηση, ύπνος

```
1.Normal Day.
2.Team Competition Day.
3.Immunity Competition Day.
4.Creativity Competition Day.
0.Quit
1

This is a normal day in the Master Chef Game.

Team Red now works.
Team Red now eats.
Team Red now eats.
Team Red now socializes.
Team Red now sleeps.
Team Blue now works.
Team Blue now eats.
Team Blue now eats.
Team Blue now socializes.
Team Blue now sleeps.

1.Normal Day.
2.Team Competition Day.
3.Immunity Competition Day.
4.Creativity Competition Day.
0.Quit
```

Illustration 2: Κανονική ημέρα

- `void teamCompetitionDay()`: Υλοποιούνται οι καθημερινές δράσεις για τις δύο ομάδες, ενώ δημιουργούνται και αντικείμενα τύπου FoodAward και TeamCompetition με κατάλληλα ορίσματα, καλείται η συνάρτηση compete της TeamCompetition και αποθηκεύεται ο ηττημένος σε καθολική μεταβλητή με όνομα loser.


```

1.Normal Day.
2.Team Competition Day.
3.Immunity Competition Day.
4.Creativity Competition Day.
0.Quit
2

This is a Team Competition day in the Master Chef Game.

Team Red now works.
Team Red now eats.
Team Blue now works.
Team Blue now eats.

WINNER INFO
Color: Blue
Wins: 1
Supplies: 140

COMPETITION INFO
Id: 0
Competition name: Lobster cooking
Winner: Blue
Name: Lentils
Solo: false
Supplies: 19

Round: 1 Duration: 10800 Winner: Red
Round: 2 Duration: 10800 Winner: Blue
Round: 3 Duration: 10800 Winner: Blue

Team Red now eats.
Team Red now socializes.
Team Red now sleeps.
Team Blue now eats.
Team Blue now socializes.
Team Blue now sleeps.
Team competition Lobster cooking is destroyed

1.Normal Day.
2.Team Competition Day.
3.Immunity Competition Day.
4.Creativity Competition Day.
0.Quit

```

Illustration 3: Ημέρα ομαδικού διαγωνισμού

- `void immunityCompetitionDay()`: Υλοποιούνται οι καθημερινές δράσεις για τις δύο ομάδες, ενώ δημιουργούνται και αντικείμενα τύπου `ImmunityAward` και `ImmunityCompetition`. Ύστερα, καλείται η συνάρτηση `compete` της `ImmunityCompetition` με όρισμα την ομάδα που έχασε στο `TeamCompetition`, αλλά μόνο σε περίπτωση που έχει ήδη πραγματοποιηθεί το `TeamCompetition`.

```

1.Normal Day.
2.Team Competition Day.
3.Immunity Competition Day.
4.Creativity Competition Day.
0.Quit
3

This is a Immunity Competition day in the Master Chef Game.

Team Red now works.
Team Red now eats.
Team Blue now works.
Team Blue now eats.

WINNER INFO
Name: Tsikilis
Gender: Male
Age: 34
Job: Cook
Technique: 77.5609
Fatigue: 64.4372
Popularity: 55.0105
Candidate: false

COMPETITION INFO
Id: 1
Competition name: Roast Beef cooking
Winner: Tsikilis
Name: Immunity
Solo: true
Votes: 2

Team Red now eats.
Team Red now socializes.
Team Red now sleeps.
Team Blue now eats.
Team Blue now socializes.
Team Blue now sleeps.
Immunity competition Roast Beef cooking is destroyed

```

Illustration 4: Ημέρα διαγωνισμού ασυλίας

- `void creativityCompetitionDay()`: Υλοποιούνται οι καθημερινές δράσεις για τις δύο ομάδες, ενώ δημιουργούνται και αντικείμενα τύπου `ExcursionAward` και `CreativityCompetition` με κατάλληλα ορίσματα και καλείται η συνάρτηση `compete` της `CreativityCompetition`.

```
This is a Creativity Competition day in the Master Chef Game.

Team Red now works.
Team Red now eats.
Team Blue now works.
Team Blue now eats.

WINNER INFO
Name: Argyris
Gender: Male
Age: 25
Job: Cook
Technique: 91.163
Fatigue: 57.864
Popularity: 56.7901
Candidate: false

COMPETITION INFO
Id: 2
Competition name: Spicy guacamole chicken cooking
Winner: Argyris
Name: Trip to Botrini's restaurant
Solo: true
Technique Bonus: 5 Popularity Penalty: -10

Ingredients:
    Onions
    Avocados
    Salt
    Pepper
    Lobster
    Rice
    Chicken
    Tabasco
    Roquefort
    Caviar

Player Argroudi is destroyed.
Player Seferidis is destroyed.
Player Christina is destroyed.
Player Symeonidis is destroyed.
Player Maia is destroyed.
Player Argyris is destroyed.
Player Magky is destroyed.
Player Vasilis is destroyed.
Player Timoleon is destroyed.
Player Nikol is destroyed.
Player Selim is destroyed.
Player Tzortzis is destroyed.
Player Savvas is destroyed.
Player Glossidis is destroyed.
Player Ilektra is destroyed.
Player Giorgos is destroyed.
Player Konstantinos is destroyed.
Player Gogo is destroyed.
Player Chambos is destroyed.
Player Nikoleta is destroyed.
Player Tsikilis is destroyed.
Player Marios is destroyed.
Team Red now eats.
Team Red now socializes.
Team Red now sleeps.
Team Blue now eats.
Team Blue now socializes.
Team Blue now sleeps.
Creativity competition Spicy guacamole chicken cooking is destroyed
```

Illustration 5: Ημέρα διαγωνισμού δημιουργικότητας