Πανεπιστήμιο Κρήτης – Τμήμα Επιστήμης Υπολογιστών

ΗΥ252 – Αντικειμενοστρεφής Προγραμματισμός

Διδάσκων: Ι. Τζίτζικας

Χειμερινό Εξάμηνο 2020-2021

PAYDAY PROJECT

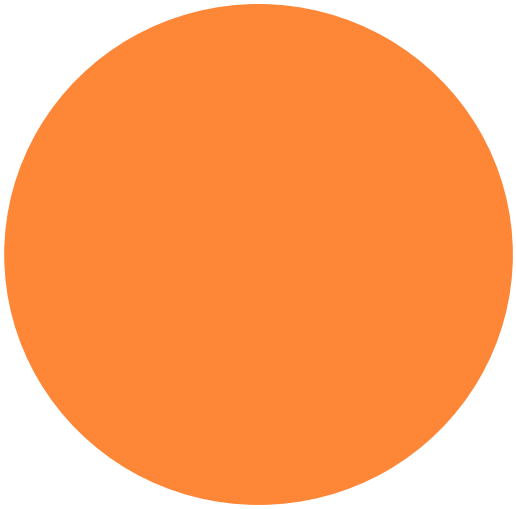
Phase A

Ταρσώ Κελετζή

csd4396

7 Δεκεμβρίου 2021

Περιεχόμενα



1. Εισαγωγή

Η πρώτη ενότητα αφορά το μοντέλο που χρησιμοποιείται για το project είναι το MVC (Model View Controller). Σύμφωνα με το μοντέλο αυτό, το project χωρίζεται σε 3 μέρη. Το πρώτο μέρος είναι το Model το οποίο διαχειρίζεται τα δεδομένα και το πως αυτά συνδέονται μεταξύ τους, δηλαδή την λογική πίσω από αυτά, μέσα σε κλάσσεις. Το δεύτερο κομμάτι είναι το View, το οποίο αφορά την απεικόνιση των δεδομένων αυτών σε γραφικό περιβάλλον και την αλληλεπίδραση με τον χρήστη. Τέλος, το τρίτο και τελευταίο κομμάτι είναι το Controller, το οποίο είναι το κομμάτι που συνδέει τα άλλα δύο κομμάτια, δηλαδή το model και το view, και χρησιμοποιείται για να επικοινωνήσουν μεταξύ τους. Η δεύτερη ενότητα αφορά τον σχεδιασμό και τις κλάσσεις που περιέχονται στο model προγραμματιστικό κομμάτι καθώς και κάποιες επεξηγήσεις όσον αφορά τις ίδιες τις κλάσσεις και τις μεθόδους τους. Έπειτα, ακολουθεί η σχεδίαση και η επεξήγηση των κλάσσεων καθώς και των μεθόδων τους που αφορούν το Controller, δηλαδή του συνδετικού κρίκου μεταξύ model και view. Στη συνέχεια, θα παρουσιαστεί και ο σχεδιασμός του view, δηλαδή του γραφικού κομματιού της εργασίας με τις απαραίτητες επεξηγήσεις. Στην πέμπτη ενότητα θα επισυναφθεί το UML class diagram που απεικονίζει τις σχέσεις εξάρτησης όλων των κλάσσεων και μεθόδων και από τα τρία μέρη του σχεδιαστικού μοντέλου και θα φανεί η μεταξύ τους αλληλεπίδραση. Η έκτη ενότητα αφορά την υλοποίηση όλων όσων αναφέρονται στις ενότητες δύο, τρία και τέσσερα και το πόσο οι στόχοι μπορεσαν να επιτευχθούν και σε τι βαθμό. Τέλος, στην έβδομη ενότητα θα γίνει λόγος για συμπεράσματα που αφορούν την εργασία, τυχόν προβλήματα και ό,τι άλλο κριθεί άξιο να αναφερθεί.

2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model

Η σχεδίαση του πακέτου model αποτελείται από τέσσερα διαφορετικά πακέτα για λόγους οργάνωσης. Το πρώτο πακέτο εν ονόματι Board, αποτελείται από μία κλάση Board. Σε αυτή την κλάση υπάρχουν οι μέθοδοι:

- public ArrayList<Card> getGameCards()

  Η οποία επιστρέφει το πεδίο gameCards της κλάσσης και αντιπροσωπεύει ουσιαστικά όλο το σύνολο των καρτών του παιχνιδιού.

- public void setGameCards (ArrayList<Card> gameCards)

  Η οποία θέτει το πεδίο gameCards της κλάσσης και τοποθετεί ουσιαστικά όλο το σύνολο των καρτών του παιχνιδιού.

- public boolean movePlayer(Player player, int die_num)

  Η οποία χρησιμοποιείται για να μετακινηθεί ο παίκτης τόσες θέσεις όσες και το die_num, δηλαδή ο αριθμός που έχει φέρει στο ζάρι ο παίκτης.

- public void setLastCardPlayed(Card lastCardPlayed)

  Η οποία θέτει την τελευταία κάρτα που έπαιξε ο παίκτης, δηλαδή την τελευταία mailCard/dealCard που χρησιμοποίησε σε περίπτωση που χρειαστεί για το υπόλοιπο του παιχνιδιού.

- public Card getLastCardPlayed()

  Η οποία επιστρέφει την τελευταία κάρτα που πήρε/έδωσε ο παίκτης σε περίπτωση που χρειαστεί για το υπόλοιπο του παιχνιδιού.

- public Board()

  Ο constructor της κλάσσης.

Το επόμενο πακέτο του model είναι το Cards, το οποίο περιέχει την abstract class Card και όλες τις άλλες κλάσσεις που την κληρονομούν-κάνουν extend, δηλαδή οι MailCard και DealCard.

Η κλάση Card αποτελείται από τις εξής μεθόδους:

- public Icon getIcon()

  Η οποία θα επιστρέφει το εικονίδιο που θα βρίσκεται πάνω στο κουμπί μιας κάρτας.

- public void setIcon(Icon icon)

  Η οποία θα θέτει το εικονίδιο πάνω στο κουμπί μιας κάρτας.

- public abstract void fillArray(Icon icon)

  Η οποία θα γεμίζει τον πίνακα Icon[] της κλάσσης που κληρονομεί την κλάση Card, και ο οποίος θα έχει μέγεθος τόσο όσος ο αριθμός των καρτών που είναι διαθέσιμες για το παιχνίδι. Επειδή ο αριθμός θα διαφέρει ανάλογα με το είδος της κάρτας, η κλάση αυτή θα υλοποιείται διαφορετικά σε κάθε κλάση και έχει δηλωθεί ως abstract.

- public int getNumber()

  Η οποία επιστρέφει το πεδίο number της κλάσσης που αφορά τον αριθμό των καρτών.

- public void setNumber(int number)

  Η οποία θέτει το πεδίο number της κλάσσης που αφορά τον αριθμό των καρτών.

- public abstract void performAction()

  Η υλοποίηση αυτής της μεθόδου θα διαφέρει στις κλάσσεις που κληρονομούν την κλάση Card. Για αυτό τον λόγο, επειδή οι ενέγειες που θα γίνουν θα είναι διαφορετικές, είναι δηλωμένη ως abstract.

- public Card()

  Ο constructor της κλάσσης.

Τα attributes της κλάσσης Card:

Ο αριθμός των καρτών.

- private static int number

Το εικονίδιο της κάθε κάρτας.

- protected static Icon icon

Ο πίνακας από εικονίδια για το κάθε είδος κάρτας.

- protected static Icon[] icons;

Η MailCard αποτελείται από:

- public void fillArray(Icon icon)

  Η οποία θα γεμίζει τον πίνακα Icon[] της κλάσσης και ο οποίος θα έχει μέγεθος τόσο όσος ο αριθμός των συγκεκριμένων καρτών που είναι διαθέσιμες για το παιχνίδι.

- Card getCard(Stack<MailCard> mailCardStack, Player p)

Η οποία θα επιστρέφει μια κάρτα από την στοίβα με τις mail cards, επιτρέποντας στον χρήστη να “τραβήξει” μία κάρτα όταν χρειαστεί.

- public void performAction()

  Η οποία αφορά τις ενέργειες που θα γίνουν όταν ο παίκτης τραβήξει μια mail card.

- public MailCard()

  Ο constructor της κλάσσης.

Και η DealCard αποτελείται από τις εξής μεθόδους:

- public void setCost_buy(int cost_buy)

  Η οποία θέτει το κόστος αγοράς μιας deal card.

- public int getCost_buy()

  Η οποία επιστρέφει το κόστος αγοράς μιας deal card.

- public void setCost_sell(int cost_sell)

  Η οποία θέτει το κόστος πώλησης μιας deal card.

- public int getCost_sell()

  Η οποία επστρέφει το κόστος πώλησης μιας deal card.

- public void fillArray(Icon icon)

  Η οποία θα γεμίζει τον πίνακα Icon[] της κλάσσης και ο οποίος θα έχει μέγεθος τόσο όσος ο αριθμός των καρτών που είναι διαθέσιμες για το παιχνίδι.

- Card getCard(Stack<DealCard> dealCardStack, Player p)

  Η οποία θα επιστρέφει μια κάρτα από την στοίβα με τις deal cards, επιτρέποντας στον χρήστη να “τραβήξει” μία κάρτα όταν χρειαστεί.

- public static boolean wantsTheCard(Player p)

  Η οποία επιστρέφει αληθές όταν ο παίκτης όντως θέλει να πάρει την deal card.

- public void performAction()

  Η οποία αφορά τις ενέργειες που θα γίνουν όταν ο παίκτης τραβήξει μια deal card και θέλει να την κρατήσει.

- public DealCard()

  Ο constructor της κλάσσης.

Έχουμε την κλάση Player στο πακέτο Player. Η συγκεκριμένη κλάσση έχει τις εξής μεθόδους:

- public ArrayList<MailCard> getMailCards()

  Η οποία επιστρέφει τις mail cards του παίκτη.

- public ArrayList<DealCard> getDealCards()

  Η οποία επιστρέφει τις deal cards του παίκτη.

- public int getScore()

  Η οποία επιστρέεφει το score του παίκτη στο τέλος κάθε παιχνιδιού/γύρου.

- public int getAvailableMoney()

  Η οποία επιστρέφει τα διαθέσιμα χρήματα που βρίσκονται στην κατοχή του παίκτη.

- public int getDays()

  Η οποία επιστρέφει τις μέρες που απομένουν για να τελειώσει το παιχνίδι ο παίκτης.

- public int getLoans()

  Η οποία επιστρέφει τα δάνεια του παίκτη. Με τον όρο δάνεια εννοειται το χρηματικό ποσό που χρωστά.

- public Position getPosition()

  Η οποία επιστρέφει την θέση στο ταμπλό που βρίσκεται ο παίκτης την συγκεκριμένη στιγμή.

- public void initPlayer()

  Η οποία αρχικοποιεί τα πεδία του παίκτη όταν πρόκειται να ξεκινήσει το παιχνίδι.

- public int getID()

  Η οποία επιστρέφει το id του παίκτη το οποίο μπορεί να είναι είτε 1 είτε 2, καθώς το παιχνίδι παίζεται από 2 παίκτες.

- public void setID(int id) throws Exception

  Η οποία θέτει το id του παίκτη το οποίο μπορεί να είναι είτε 1 είτε 2, καθώς το παιχνίδι παίζεται από 2 παίκτες.

- public void setAvailableMoney(int available_money)

  Η οποία θέτει τα διαθέσιμα χρήματα του παίκτη.

- public void setPosition(Position player_position)

  Η οποία θέτει την θέση του παίκτη πάνω στο ταμπλό.

- public void setInfo()

  Η οποία θέτει τις πληροφορίες του χρήστη έτσι ώστε να μπορέσουν να τοποθετηθούν στο ταμπλό.

- public void setDealCards(ArrayList<DealCard> c)

  Η οποία θέτει τις deal cards του παίκτη εαν αυτό χρειαστεί.

- public void setMailCards(ArrayList<MailCard> c)

  Η οποία θέτει τις mail cards του παίκτη εαν αυτό χρειαστεί.

- public boolean gotTheDeal()

  Η οποία επιστρέφει το έαν ο χρήστης εχει αποφασίσει να δεχτεί τη συμφωνία και να πάρει την κάρτα συμφωνίας.

- public boolean hasPlayed()

  Η οποία επιστρέφει το έαν ο χρήστης εχει παίξει εάν ήταν η σειρά του να παίξει.

- public void setHasPlayed(boolean has_played)

  Η οποία θέτει το πεδίο has_played του χρήστη, εαν εχει παίξει και ήταν η σειρά του να παίξει.

- public boolean hasFinished()

  Η οποία ελέγχει εάν ο χρήστης έχει τελειώσει το παιχνίδι/τον γύρο.

- public boolean hasStarted()

  Η οποία ελέγχει εάν ο χρήστης έχει ξεκινήσει το παιχνίδι/τον γύρο.

- public void setHasStarted(boolean hasStarted)

  Η οποία θέτει το πεδίο has_started του παίκτη εάν έχει ξεκινήσει το παιχνίδι/τον γύρο.

- public void setHasFinished(boolean hasFinished)

  Η οποία θέτει το πεδίο has_finished του παίκτη εάν έχει τελειώσει το παιχνίδι/τον γύρο.

- public void reduceMoney(int money)

  Η οποία μειώνει τα διαθέσιμο χρήματα του παίκτη όταν αυτός πρέπει να πληρώσει δάνειο/λογαριασμούς ή να δώσει χρήματα στον άλλο παίκτη.

- public void upgradeMoney(int money)

  Η οποία αυξάνει τα διαθέσιμο χρήματα του παίκτη όταν αυτός κερδίζει χρήματα λόγω κάποιας θέσης στην οποία έχει βρεθεί, καποιας κάρτας που έχει πάρει ή χρημάτων που έχει πάρει απο τον άλλο παίκτη.

- public void getLoan(int money)

  Η οποία αυξάνει τα διαθέσιμα χρήματα του παίκτη καθώς και το πεδίο που αφορά τα δάνεια του παίκτη όταν αυτός παίρνει δάνειο.

- public Player()

  Και ο constructor της κλάσσης.

Τα attributes του Player είναι:

Τα βασικά στοιχεία του, όπως το σκορ, οι μέρες, τα δάνεια, οι λογαριασμοί και τα διαθέσιμα χρήματα.

- private int score
- private int days
- private int loan
- private int bills
- private int available_money

Οι πληροφορίες του.

- public String info

Κάποιες βασικές booleans για την ομαλή ροή του παιχνιδιού.

- private boolean has_finished
- private boolean has_played
- private boolean has_started
- private boolean turn

Η “ταυτότητά” του.

- private int id //can be either 1 or 2 (we have only 2 players)

Η θέση στην οποία βρίσκεται.

- private Position player_position

Οι κάρτες του.

- private ArrayList<DealCard> deal_cards = new ArrayList<>()
- private ArrayList<MailCard> mail_cards = new ArrayList<>()

Το πιόνι του.

- public Icon pawn

Επίσης, υπάρχει και το πακέτο Positions, το οποίο περιέχει την abstract κλάση Position και τις κλάσσεις Buyer, Casino, DicePosition, Lottery, PayDay, Radio, SundayFootballGame, Sweepstake, ThursdayRise, Yard που την κληρονομούν.

Η κλάση Position έχει τις εξής μεθόδους:

- public Icon getIcon()

  Η οποία επιστρέφει το icon του κουμπιού της κάθε θέσης.

- public void setIcon(Icon icon)

  Η οποία θέτει το icon του κουμπιού της κάθε θέσης.

- public abstract void fillArrayPos(Icon icon)

  Η οποία γεμίζει το array Icon[] της θέσεις με τόσες εικόνες (αν και θα είναι η ίδια εικόνα για κάθε θέση) όσες και οι διαθέσιμες θέσεις στο ταμπλό.

- public abstract void performActionPos()

  Η οποία ειναι η μέθοδος που καθορίζει τις ενέργειες που θα ακολουθησουν όταν ο παίκτης βρεθεί σε κάποια θέση, και όταν βρεθεί θα υλοποιηθεί η overriden μέθοδος της συγκεκριμένης θέσης.

- public Position()

  Ο constructor.

Τα attributes της κλάσσης Position:

Το εικονίδιο που έχει η κάθε θέση στο ταμπλό.

- protected static Icon icon

Ο πίνακας από εικονίδια όσα και οι θέσεις στο ταμπλό.

- protected static Icon[] iconPos

Η κλάση Buyer έχει τις εξής μεθόδους:

- public void fillArrayPos(Icon icon)

  Η οποία γεμίζει το array Icon[] της θέσεις με την εικόνα της θέσης Buyer τόσες φορές όσες και οι διαθέσιμες θέσεις στο ταμπλό.

- public void performActionPos()

  Η οποία ειναι η μέθοδος που καθορίζει τις ενέργειες που θα ακολουθησουν όταν ο παίκτης βρεθεί στην θέση του αγοραστή.

- public Buyer()

  Ο constructor της κλάσσης.

Η κλάση Casino έχει τις εξής μεθόδους:

- public void fillArrayPos(Icon icon)

  Η οποία γεμίζει το array Icon[] της θέσεις με την εικόνα της θέσης Casino τόσες φορές όσες και οι διαθέσιμες θέσεις στο ταμπλό.

- public int chooseNumber(Player p)

  Επιτρέπει στον player p να διαλέξει έναν αριθμό (1-6).

- public int rollTheDie(Player p)

  Επιτρέπει στο player p να ρίξει το ζάρι.

- public void performActionPos()

  Η οποία ειναι η μέθοδος που καθορίζει τις ενέργειες που θα ακολουθήσουν όταν ο παίκτης βρεθεί στην θέση του FamilyCasino, δηλαδή μόλις επιτρέψει στους χρήστες να διαλέξουν απο έναν αριθμό, όταν κάποια ζαριά φέρει 1 από τους 2 αυτούς αριθμούς πρώτη, εκείνος ο παίκτης κερδίζει το ποσό.

- public Casino()

  Ο constructor της κλάσσης.

Η κλάση DicePosition έχει τις εξής μεθόδους:

- public int getMoney()

  Η οποία επιστρέφει τα χρήματα που μπορεί να κερδίσει ο παίκτης.

- public int getDieNumber()

  Η οποία επιστρέφει τον αριθμό που έφερε ο παίκτης στο ζάρι.

- public void setMoney(int money)

  Η οποία θέτει τα χρήματα που μπορεί να κερδίσει ο παίκτης αν τα κερδίσει.

- public void setDieNumber(int dieNumber)

  Η οποία θέτει τον αριθμό ζαριού που έφερε ο παίκτης.

- public void performAction(Player p, int dieNumber)

  Η οποία μετακινεί τον παίκτη τόσες θέσεις όσο έφερε στο ζάρι.

- public void fillArrayPos(Icon icon)

  Η οποία θέτει το array Icon[] με τις εικόνες του ζαριού. Το μέγεθος του array είναι όσες και οι θέσεις του ζαριού (1-6).

- public DicePosition()

  Ο constructor της κλάσσης.

Η κλάση Lottery:

- public void fillArrayPos(Icon icon)

  Η οποία θέτει το array Icon[] με την εικόνα για το Lottery. Το μέγεθος του array είναι όσες και οι θέσεις στο ταμπλό.

- public int rollTheDie()

  Η οποια επιτρέπει στον παίκτη να ρίξει το ζάρι και επιστρέφει το αποτέλεσμα που έφερε.

- public void performActionPos()

  Η οποία υλοποιεί τις ενέργειες που πρέπει να γίνουν όταν ένας παικτης βρεθεί σε αυτή τη θέση, δηλαδή δίνει χρήματα ή κερδίζει ανάλογα με το τι ζαριά έφερε, δηλαδή το rollTheDie().

- public Lottery()

  Ο constructor της κλάσσης.

Η κλάση PayDay:

- public void getPaid(Player p, int money)

  Η οποία πληρώνει τον χρήστη 3500 λόγω της θέσης στην οποία βρίσκεται.

- public boolean checkForBills(Player p)

  Η οποία ελέγχει εάν ο χρήστης έχει λογαριασμούς που δεν έχει ξεπληρώσει.

- public boolean checkForLoans(Player p)

  Η οποία ελέγχει εάν ο χρήστης έχει δάνεια που δεν έχει ξεπληρώσει.

- public void payTheBills(Player p)

  Η οποία χρησιμεύει στο να πληρώσει ο παίκτης τους λογαριασμούς.

- public void ignoreCards(Player p)

  Η οποία χρησιμεύει στο να μπορέσει ο παίκτης να απορρίψει τις κάρτες εάν είναι ο τελευταίος μήνας του παιχνιδιού.

- public boolean hasEnoughMoney(Player p)

  Η οποία ελέγχει εάν ο χρήστης έχει αρκετά χρήματα έτσι ώστε να μπορέσει να πληρώσει τους λογαριασμούς.

- public static boolean isTheLastMonth()

  Η οποία ελέγχει εάν ο τρέχον μήνας είναι ο τελευταίος του παιχνιδιού.

- public void fillArrayPos(Icon icon)

  Η οποία θέτει το array Icon[] με το κατάλληλο εικονίδιο και έχεις μέγεθος όσες οι θέσεις στο ταμπλο(στη συγκεκριμένη περίπτωση 1).

- public void performActionPos()

  Η οποία υλοποιεί τις ενέργειες που πρέπει να γίνουν όταν ο παίκτης βρεθεί σε αυτήν την θέση. Τα χρήματα του παίκτη αυξάνονται κατα 3500, πληρώνει 10% του δανείου(εάν αυτό υπάρχει) επομένως μειώνονται τόσο τα χρήματα του όσο και το δάνειο, πληρώνει τους λογαριασμούς του και εάν είναι ο τελευταίος μήνας αγνοεί τις κάρτες αλλιώς το position του γινεται η αρχή.

- public PayDay()

  Ο constructor της κλάσσης.

Επειτα η κλάση Radio ακολουθεί με τις μεθόδους:

- public int rollTheDie()

  Η οποια επιτρέπει στον παίκτη να ρίξει το ζάρι και επιστρέφει το αποτέλεσμα που έφερε.

- public void fillArrayPos(Icon icon)

  Η οποία θέτει το array Icon[] με το κατάλληλο εικονίδιο και έχει μέγεθος όσες οι θέσεις στο ταμπλο.

- public void performActionPos(Player p1, Player p2)

  Η οποία υλοποιεί τις ενέργειες που πρέπει να γίνουν όταν ο παίκτης βρεθεί σε αυτήν την θέση, δηλαδή δίνει 1000 ευρώ στον παίκτη με την μεγαλύτερη ζαριά αφού έχουν ρίξει και οι δύο παίκτες.

- public Radio()

  Και ο constructor.

Στην κλάσση SundayFootballGame υπάρχουν οι μέθοδοι:

- public static boolean wantsToBet(Player p)

  Η οποία ελέγχει εάν ο παίκτης θέλει να στοιχηματίσει ή όχι.

- public boolean hasWonTheBet(Player p)

  Η οποία κάνει την κλήρωση και επιστρέφει εάν ο παίκτης έχει κερδίσει το στοίχημα ή όχι.

- public int moneyWon()

  Η οποία προσαρμόζει το ποσό που κέρδισε ο παίκτης και επιστρέφει το κέρδος εάν έχει στοιχιματίσει και έχει κερδίσει τον αγώνα ή αν χάνει τα χρήματά του.

- public void fillArrayPos(Icon icon)

  Η οποία δεν κάνει κάτι απλά πρέπει να γίνει override λόγω του ότι είναι δηλωμένη ως abstract στην κλάσση Position που κληρονομείται από την SundayFootballGame.

- public void performActionPos()

  Η οποία είναι η κληρονομημένη μέθοδος και εκτελεί τις ενέργειες που πρέπει να γίνουν όταν ο παίκτης βρεθεί σε αυτήν την θέση ενώ στο τέλος adjusts the προσαρμόζει τα χρήματα του παίκτη κατάλληλα.

- public SundayFootballGame()

  Ο constructor.

Στην συνέχεια έχουμε την κλάσση Sweepstake, με τις εξής μεθόδους:

- public void fillArrayPos(Icon icon)

  Η οποία θέτει το array Icon[] με το κατάλληλο εικονίδιο και έχει μέγεθος όσες οι θέσεις στο ταμπλο.

- public int rollTheDie(Player p)

  Η οποία επιστρέφει το αποτέλεσμα μετά την ζαριά που υποχρεούται να ρίξει ο παίκτης.

- public void performActionPos()

  Η οποία είναι η κληρονομημένη μέθοδος και εκτελεί τις ενέργειες που πρέπει να γίνουν όταν ο παίκτης βρεθεί σε αυτήν την θέση, δηλαδή αυξάνει τα χρήματά του κατά 1000*rollTheDie(player).

- public Sweepstake()

Ο constructor.

Ακολουθεί η κλάση ThursdayRise, η οποία έχει τις μεθόδους:

- public static boolean wantsToBet(Player p)

  Η οποία ελέγχει εάν ο παίκτης θέλει να στοιχηματίσει ή όχι.

- public void fillArrayPos(Icon icon)

  Η οποία δεν κάνει κάτι απλά πρέπει να γίνει override λόγω του ότι είναι δηλωμένη ως abstract στην κλάση Position που κληρονομείται από την ThursdayRise.

- public int moneyWon()

  Η οποία προσαρμόζει το ποσό που κέρδισε ο παίκτης και επιστρέφει το κέρδος εάν έχει στοιχιματίσει και κερδίσει ή αν έχει χάσει τα χρήματά του.

- public boolean hasWonTheBet(Player p)

  Επιστρέφει το εάν ο παίκτης κέρδισε ή όχι.

- public void performActionPos()

  Η οποία είναι η κληρονομημένη μέθοδος και εκτελεί τις ενέργειες που πρέπει να γίνουν όταν ο παίκτης βρεθεί σε αυτήν την θέση και προσαρμόζει τα χρήματα του παίκτη.

- public ThursdayRise()

  Ο απαραίτητος constructor.

Τέλος, υπάρχει και η κλάση Yard η οποία έχει τις μεθόδους:

- public void fillArrayPos(Icon icon)

  Η οποία θέτει το array Icon[] με το κατάλληλο εικονίδιο και έχεις μέγεθος όσες οι θέσεις στο ταμπλο.

- public int rollTheDie()

  Η οποια επιτρέπει στον παίκτη να ρίξει το ζάρι και επιστρέφει το αποτέλεσμα που έφερε.

- public void performActionPos()

  Η οποία είναι η κληρονομημένη μέθοδος και εκτελεί τις ενέργειες που πρέπει να γίνουν όταν ο παίκτης βρεθεί σε αυτήν την θέση, δηλαδή μειώνει τα χρήματα του παίκτη κατα 100*rollTheDie() και τον αναγκάζει να πάρει μια κάρτα συμφωνίας χωρίς περαιτέρω πληρωμή.

- public Yard()

  Ο constructor της κλάσσης.

3. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller

Στο πακέτο Controller υπάρχει μία μόνο κλάση, η Controller, η οποία κάνει implement το ActionListener και extend την κλάση GUI. Στην κλάση αυτή υπάρχουν οι μέθοδοι:

- public void startGame()

  Η οποία χρησιμοποείται για την αρχικοποίηση του παιχνιδιού και συνδέει το model με το view καλώντας την μέθοδο initComponents() και initButtons() από το GUI.java που αφορά το view κομμάτι του σχεδιαστικού μοντέλου, καθώς και σεττάροντας τις κάρτες και τις θέσεις του παιχνιδιού που βρίσκονται στο πακέτο model και καλώντας την initPlayer() για κάθε παίκτη του παιχνιδιού.

- public String Score()

  Η οποία επιστρέφει το μέχρι τώρα score των παικτών χρησιμοποιώντας μεθόδους απο την κλάση Player του model, το οποίο θα τοποθετηθεί στο infoBox του γραφικού κομματιού του παιχνιδιού.

- public void reduceJackpot(int money)

  Η οποία μειώνει το χρηματικό ποσό που έχει συγκεντρωθεί στο κουμπί jackpot όταν κάποιος παίκτης παίρνει τα χρήματα.

- public void upgradeJackpot(int money)

  Η οποία αυξάνει το χρηματικό ποσό που έχει συγκεντρωθεί στο κουμπί jackpot όταν κρίνεται αναγκαίο κατά τη διαρκεια του παιχνιδιού.

- public Player getTurn()

  Η οποία επιστρέφει τον παίκτη του οποιου είναι η σειρά να παίξει.

- public boolean moveIsCorrect(int days, int new_days, int new_die_num)

  Η οποία ελέγχει εάν η κίνηση ενός εκ των δύο παικτών είναι σωστή.

- public static boolean gameHasFinished()

  Η οποία ελέγχει εάν το παιχνίδι έχει τελειώσει.

- public int isTheWinner()

  Η οποία επιστρέφει το id του παίκτη που κέρδισε το παιχνίδι(αν επιστρέψει 0 τότε έχουμε ισοπαλία).

- public void shuffleCards(Stack<Card> c)

Η οποία ανακατεύει τις κάρτες όταν χρειάζεται.

- public void actionPerformed(ActionEvent actionEvent)

  Η οποία καθορίζει τις ενέργειες που θα ακολουθήσουν όταν πατιούνται συγκεκριμένα κουμπιά στο ταμπλό.

- public Controller()

  Ο απαραίτητος constructor.

4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View

Στο πακέτο view, υπάρχει η κλάσση GUI. Σε αυτήν την κλάσση υπάρχουν οι εξής μέθοδοι:

- private static void initComponents()

  Στην οποία αρχικοποιούνται τα components που θα χρειαστούν για το γραφικό περιβάλλον του παιχνιδιού.

- private static void updateInfo()

  Στην οποία ενημερώνεται το Info Box που αφορά τις πληροφορίες του παιχνιδιού.

- public static void initButtons()

  Σε αυτήν την κλάσση γίνονται initialize τα κουμπιά με την έννοια ότι τους προστίθονται τίτλοι και εικόνες και τοποθετούνται στο ταμπλό.

Καθώς και ο constructor της στον οποίο θα γίνονται οι απαραίτητες αρχικοποιήσεις - (κλήσεις των παραπάνω μεθόδων).

- public GUI()

Τα attributes της συγκεκριμένης κλάσσης είναι:

Η εικόνα η οποία θα τοποθετηθεί στο background.

- private Image image

  Το παράθυρο του παιχνιδιού.

- public JFrame game

  Το gridlayout στο οποίο θα βρίσκονται οι θέσεις του παιχνιδιού.

- public GridLayout board

Ο πίνακας από Strings που είναι οι τίτλοι στο κάθε κουμπί του GridLayout (του ταμπλο).

- public String[] buttonTitles = new String[32]

Τα κουμπιά που ακολουθούν είναι 1 κουμπί για το logo του παιχνιδιού, 1 για το κουμπί start που σηματοδοτεί την έναρξη του παιχνιδιού και τα υπολοιπα αφορούν τις μέρες του μήνα.

- public JButton logo
- public JButton startButton
- public JButton Button1
- public JButton Button2
- public JButton Button3
- public JButton Button4
- public JButton Button5
- public JButton Button6
- public JButton Button7
- public JButton Button8
- public JButton Button9
- public JButton Button10
- public JButton Button11
- public JButton Button12
- public JButton Button13
- public JButton Button14
- public JButton Button15
- public JButton Button16
- public JButton Button17
- public JButton Button18
- public JButton Button19
- public JButton Button20

- public JButton Button21
- public JButton Button22
- public JButton Button23
- public JButton Button24
- public JButton Button25
- public JButton Button26
- public JButton Button27
- public JButton Button28
- public JButton Button29
- public JButton Button30
- public JButton Button31
- public JButton ButtonJackpot
- public JButton dealCards
- public JButton mailCards

Ακολουθούν κουμπιά που αφορούν το ταμπλό του παίκτη 1.

- public JButton RollDice1
- public JButton MyDealCards1
- public JButton GetLoan1
- public JButton EndTurn1
- public JButton die1

Ακολουθούν κουμπιά που αφορούν το ταμπλό του παίκτη 2.

- public JButton RollDice2
- public JButton MyDealCards2
- public JButton GetLoan2
- public JButton EndTurn2
- public JButton die2

Το infoBox αφορά τις πληροφορίες που αφορούν το παιχνίδι.

- public JTextField infoBox
- public JLayeredPane paydayPane

Επίσης υπάρχει η κλάση myDesktopPane στο ίδιο αρχείο και η βασική της μέθοδος είναι η:

- public void paintComponent(Graphics g)

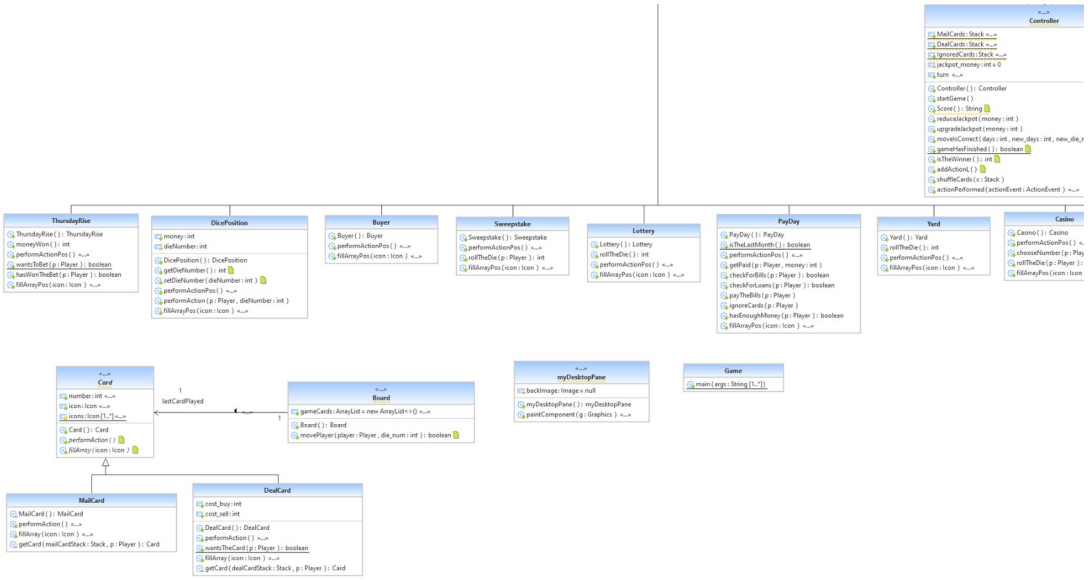
  Η οποία βάζει στο background του ταμπλό μια εικόνα.

5. Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML

Έχω σπάσει το διάγραμμα σε 2 κομμάτια λόγω μεγέθους. Επισυνάπτεται και μία 3η εικόνα με την ολοκληρωμένη απεικόνιση του διαγράμματος. Όμως παρόλα αυτά φαίνονται οι εξαρτήσεις των κλάσσεων και των μεθόδων τους και η συνοχή τους.

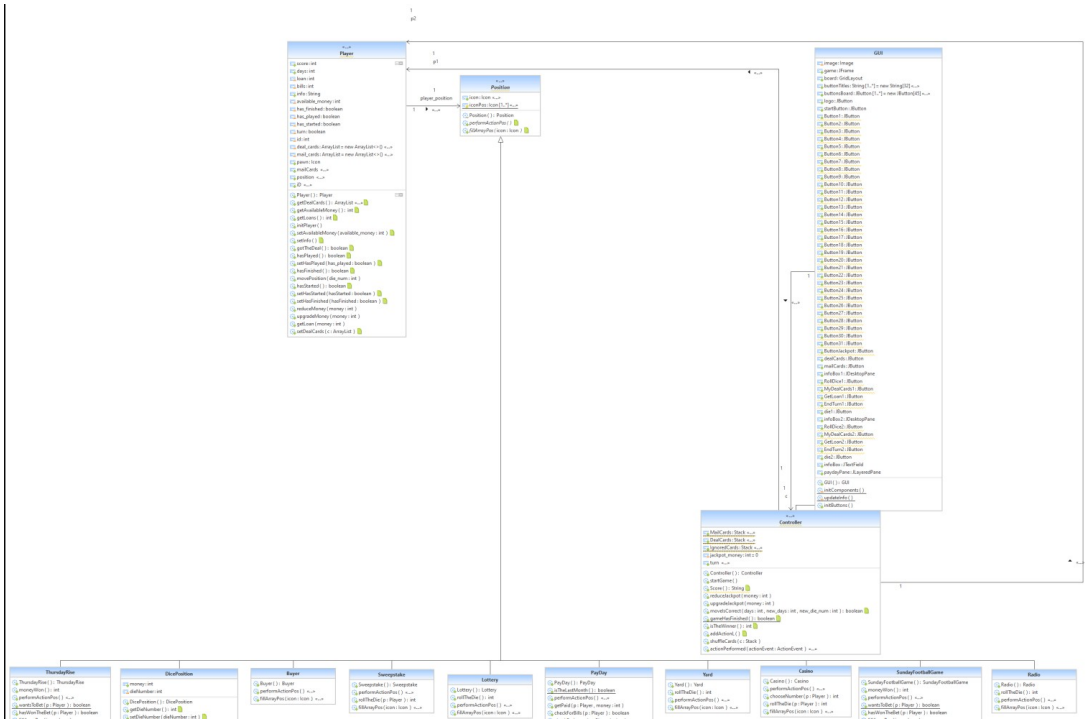
1η εικόνα

Οι εξαρτήσεις των κλάσσεων που κληρονομούν την Position (η οποία φαίνεται στην 2η εικόνα) και έχουν να υλοποιήσουν την μέθοδο performAction() ανάλογα με το συγκεκριμένο Position στο οποίο βρίσκονται. Επίσης φαίνεται η εξάρτηση των κλάσσεων MailCard και DealCard από την κλάση Card, καθώς επίσης και η σύνδεση της με την κλάση Board.



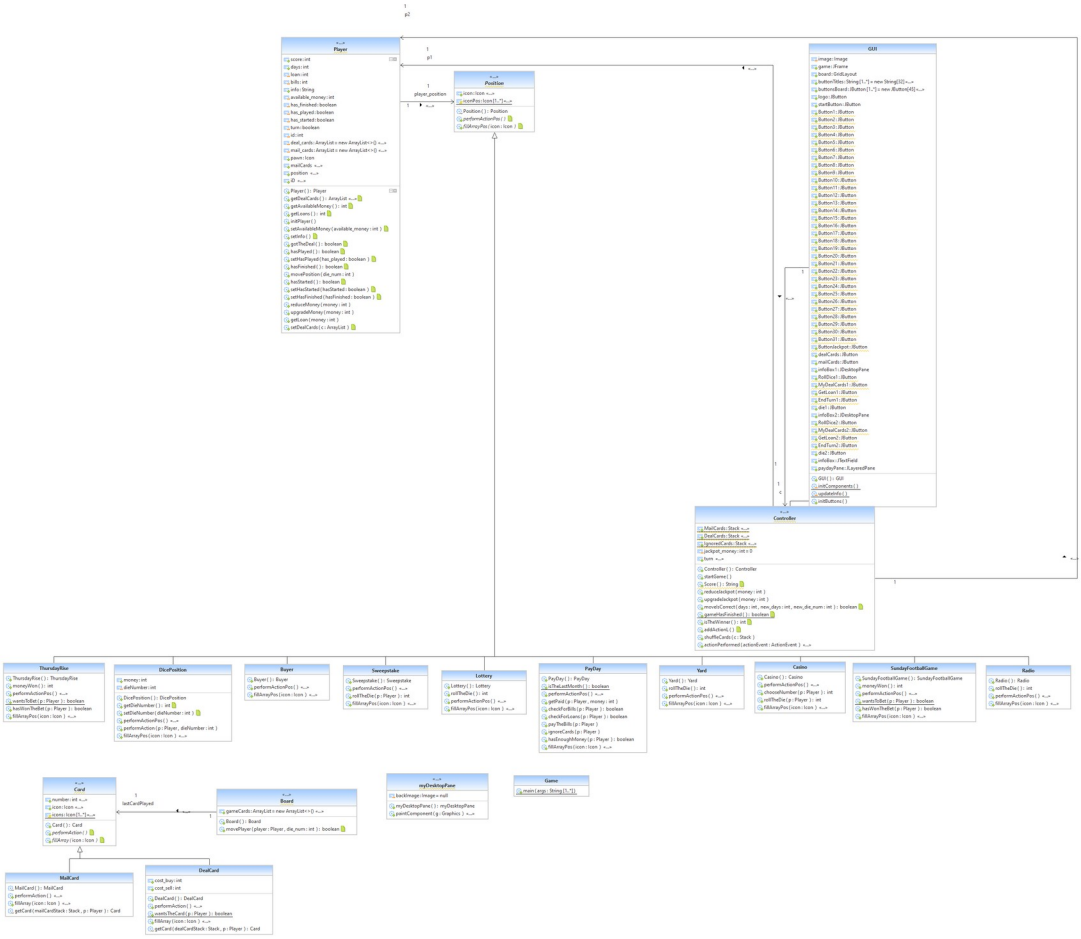
2η εικόνα

Φαίνεται η κλάση Player του πακέτου model καθώς και η αλληλεπίδραση της με το πακέτο controller και την κλάση Position. Επίσης γίνεται αντιληπτό και ότι ο Controller επικοινωνεί τόσο με το GUI όπως και με το model.



3η εικόνα

Η ολοκληρωμένη απεικόνιση του διαγράμματος UML.



6. Λειτουργικότητα (Β Φάση)

Σε αυτήν την ενότητα θα γράψετε στη Β φάση ποια ερωτήματα καταφέρατε να υλοποιήσετε είτε επιτυχώς είτε εν μέρει (και ενδεχομένως ποια όχι).

7. Συμπεράσματα

Τα συμπεράσματα της πρώτης φάσης είναι ότι οι κλάσεις μπορούν να ομαδοποιηθούν συμφωνα με τα πεδία και τις λειτουργίες τους. Επίσης η δημιουργία abstract κλάσεων με τις κατάλληλες μεθόδους μπορεί να υπάρξει πολύ αποδοτική όσον αφορά το κομμάτι των κλάσεων που τις κάνουν extend και υλοποιούν συγκεκριμένες μεθόδους. Βέβαια, μπορεί να είναι και σωστό τα κουμπιά που αφορούν τον κάθε παίκτη ή το JTextField του καθενος επειδή ακριβώς ανήκει στον παίκτη να βρίσκεται στην κλάση Player. Ωστόσο αυτό αντικρούεται με το σχεδιαστικό μοντέλο MVC (model view controller), καθώς σύμφωνα με αυτό επειδή αφορά το γραφικό κομμάτι πρέπει να βρίσκεται στο view πακέτο και όχι στο model. Λόγω της δικής μου προσέγγισης, υπάρχει και μέθοδος που πρέπει να γίνει override ακόμα και αν δεν χρειάζεται να υλοποιηθεί. Στην περιγραφή των μεθόδων και πριν τα attributes (σε όσες κλάσσεις υπάρχουν) αναφέρω και τον constructor της καθεμιάς παρόλο που δεν ειναι μέθοδος, απλώς θεωρησα ότι είναι καλό να τον αναφέρω. Τέλος, δεν έκρινα απαραίτητο η κλάση Player να είναι abstract, καθώς θα έχουμε μόνο δύο στιγμιότυπα των δύο παικτών. Τέλος δεν γίνεται λόγος στην κλάση Game η οποία θα περιλαμβάνει την μέθοδο main για την υλοποίηση του παιχνιδιού καθώς είναι κομμάτι της Β' φάσης της εργασίας.