



Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών

ΗΥ252– Αντικειμενοστρεφής Προγραμματισμός

Διδάσκων: Ι. Τζίτζικας

Χειμερινό Εξάμηνο 2020-2021

PAY DAY GAME

Ιωάννης Δάβανος

Csd4622

10/12/2021

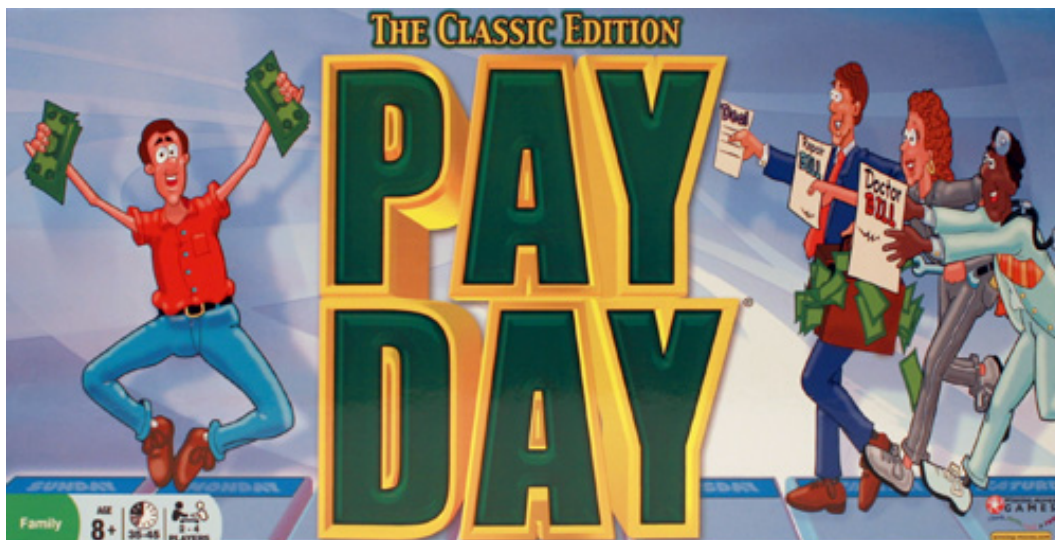
--	--	--

Περιεχόμενα

1. Εισαγωγή	2
2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model	3
3. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller	10
4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View	12
5. Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML	14
6. Λειτουργικότητα (Β Φάση)2	15
7. Συμπεράσματα	15

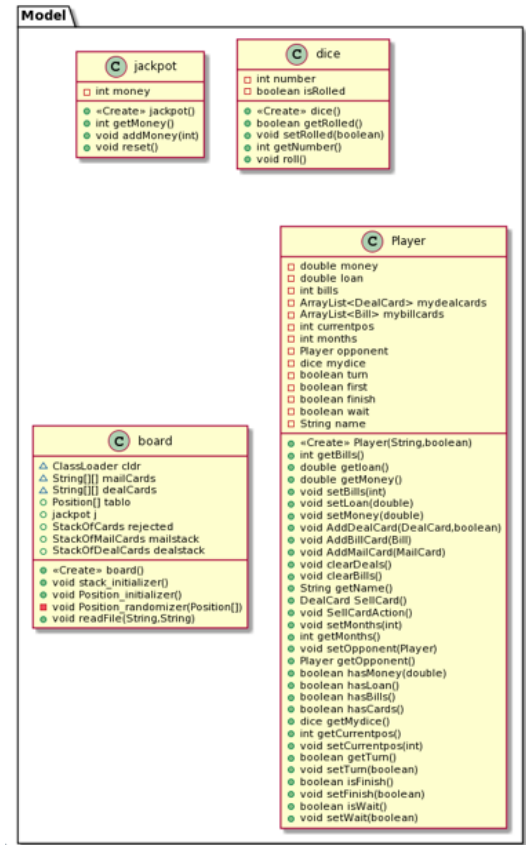
1. Εισαγωγή

Το Project σχεδιάστηκε με το σκεπτικό του MVC. Έχουμε χωρίσει δηλαδή το παιχνίδι σε 3 packages, στο Model, το View και το Controller. Στο Model περιέχονται όλα τα classes που σχετίζονται με τα 'πραγματα/οντοτητες' του παιχνιδιού (για παράδειγμα το ζαρι, ο παίκτης κτλ). Το View περιέχει τις classes που χρησιμοποιούνται για να εμφανίσουμε τις πληροφορίες μας σε ένα γραφικό περιβάλλον. Τέλος, έχουμε το Controller που περιέχει την κλαση controller και είναι αυτή που συντονίζει το παιχνίδι, χρησιμοποιεί τόσο το model όσο και το view ώστε να τρέχει το παιχνίδι. Δέχεται input από τον χρήστη και κάνει τις αλλαγές στο data (model) ενώ παράλληλα χρησιμοποιεί το view για να εμφανίζει γραφικά το τι γίνεται στο παιχνίδι. Με άλλα λόγια το controller αποτελεί τον εγκέφαλο του παιχνιδιού, δέχεται πληροφορίες από το περιβάλλον και κάνει τις κατάλληλες αλλαγές.



2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model

Στο πακέτο Model έχουμε υποπακέτα τα οποία περιέχουν κλάσεις και υποκλάσεις.



Board



Πρώτα έχουμε το board, το οποίο έχει τα πεδία :

1. public Position[] table , οι θέσεις του επιτραπέζιου.

2. `public jackpot j`, η θέση jackpot που υπάρχουν λεφτά.
3. `public StackOfCards rejected`, οι κάρτες που έχουν απορριφθεί.
4. `public StackOfMailCards mails`, η στοίβα με τις κάρτες μηνυμάτων.
5. `public StackOfDealCards deals`, η στοίβα με τις κάρτες συμφωνιών.

Ουσιαστικά, η κλάση `board` έχει την πληροφορία για το επιτραπέζιο όπως θα το έβλεπε κανείς χωρίς τους παίκτες. Ο constructor κάνει την αρχικοποίησή των παραπάνω. Οι μέθοδοι που περιέχει είναι οι εξής:

- `public void stack_initializer()`

Αρχικοποιεί τις δύο στοίβες (μηνυμάτων και deal). Θα χρησιμοποιηθεί τόσο στην δημιουργία του `board` όσο και αν όλες οι κάρτες έχουν απορριφθεί.

- `void Position_initializer()`

Δημιουργεί έναν πίνακα με `Position`, τοποθετώντας κάθε φορά τυχαίες θέσεις. Χρησιμοποιείται στην αρχή του παιχνιδιού.

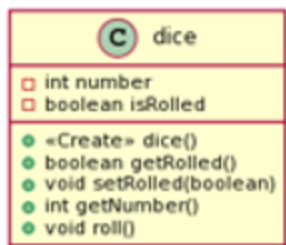
- `void Position_randomizer(Position[] table)`

Χρησιμοποιείται από την παραπάνω μέθοδο. Αλλάζει τις θέσεις του πίνακα `randomly`.

- `void readFile(String path, String type)`

Διαβάζει ένα csv αρχείο και παίρνει τις απαραίτητες πληροφορίες για να δημιουργήσει τις κάρτες μας. Θα χρησιμοποιηθεί μια φορά, στην δημιουργία των καρτών.

Dice



Επόμενη κλάση είναι το `Dice`, το οποίο έχει 2 πεδία:

- `int number`, δηλαδή τον αριθμό του ζαριού
- `boolean isRolled`, δηλαδή αν το έχουμε ρίξει ή όχι.

Οι μέθοδοι του είναι πολύ απλοί: Getters για τα πεδία του `isRolled` και `number`. Έναν setter για το `isRolled` και μια μέθοδο `roll` που αλλάζει τον αριθμό του ζαριού με έναν αριθμό από 1 έως 6. Ο constructor αρχικοποιεί το τον αριθμό με 0 και το `isRolled` με `false` σαν να μην έχουμε ρίξει ζάρι.

Jackpot



Μετά έχουμε την κλάση `jackpot`, η οποία έχει ένα πεδίο `int money`, δηλαδή τα λεφτα που κερδίζει κανείς από αυτο.

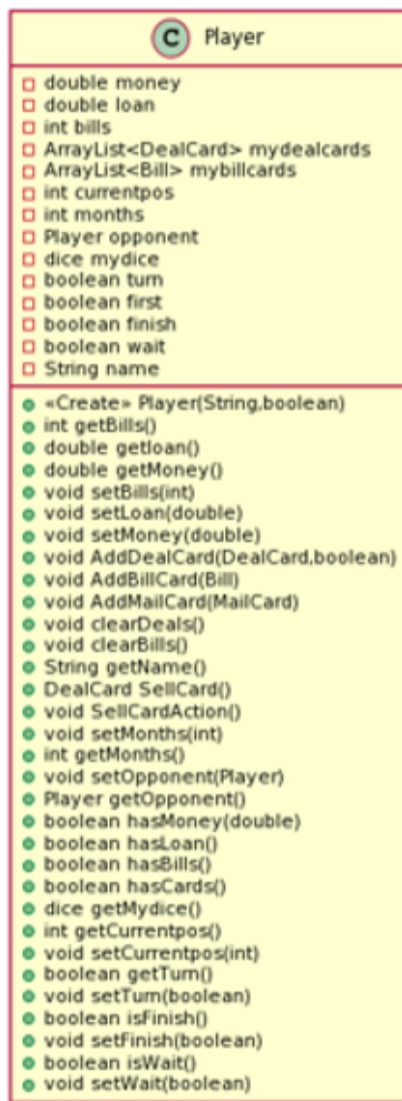
Ο constructor αρχικοποιει τα `int money=0`. Υπάρχουν οι εξής μεθοδοι:

`int getMoney`, getter δηλαδή για να παρουμε τα λεφτα,

`void reset`, να κάνουμε reset δηλαδή για να μηδενίσουμε τα λεφτά

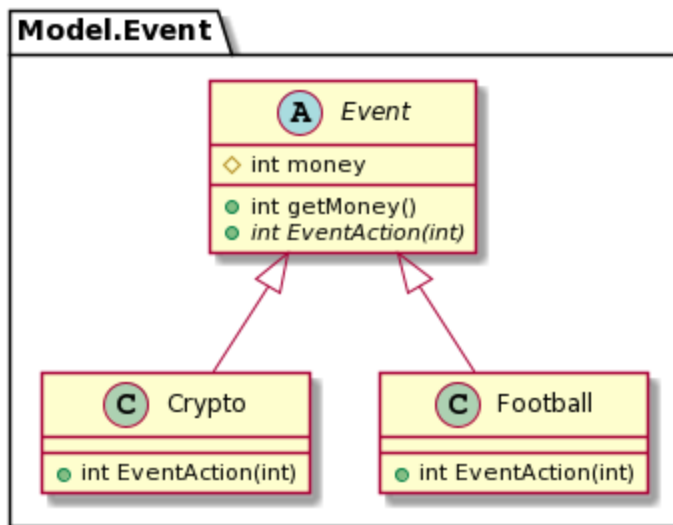
`void addMoney(int)`, για να προσθεσουμε λεφτα στα ήδη υπάρχοντα λεφτα.

Player



Επόμενη είναι μια σημαντική κλάση, η κλάση του Player. Τα πεδία του είναι οτιδήποτε έχει ένας παίκτης, δηλαδή λεφτα, δάνεια, λογαριασμούς, θέση, αντίπαλο, ζάρι, deal κάρτες, Bill κάρτες, σημαίνει αν είναι η σειρά του, αν πρέπει να περιμένει, αν τερμάτισε, και τέλος σε ποιον μήνα βρίσκεται και το όνομα του. Ο constructor αρχικοποιεί τα πεδία αυτά και δίνει ένα όνομα στον παίκτη (η παραμετρος του constructor). Υπάρχουν setters and getters για τα εξής πεδία Bills, Loan, Money, months, opponent, position, turn, finnish και wait. Ενώ για το dice έχουμε μόνο getter γιατί είναι final. Τέλος έχουμε μεθόδους για να αφαιρούμε τις κάρτες του παίκτη, να προσθέτουμε κάρτες και να τις αφαιρούμε αλλά και να τις πουλάμε.

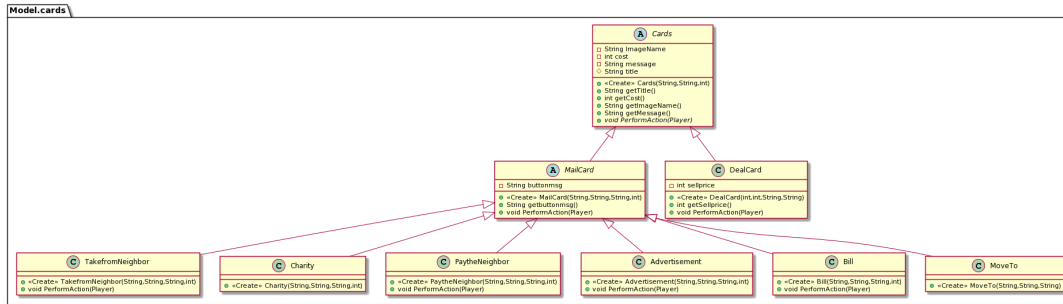
Events



Επόμενη κλάση είναι τα Events, η οποία είναι Abstract και έχει 2 subclasses την Crypto και Football. Η κλάση Event έχει έναν constructor, και μια abstract method Perform Action. Το μόνο πεδίο είναι τα λεφτα (για το οποίο έχουμε και getter, τα οποία αρχικοποιούνται στον constructor για κάθε κλάση). Το EventAction θα εκτελεί τις απαραίτητες εντολές για κάθε Event.

Cards

Έπειτα έχουμε τις κάρτες:



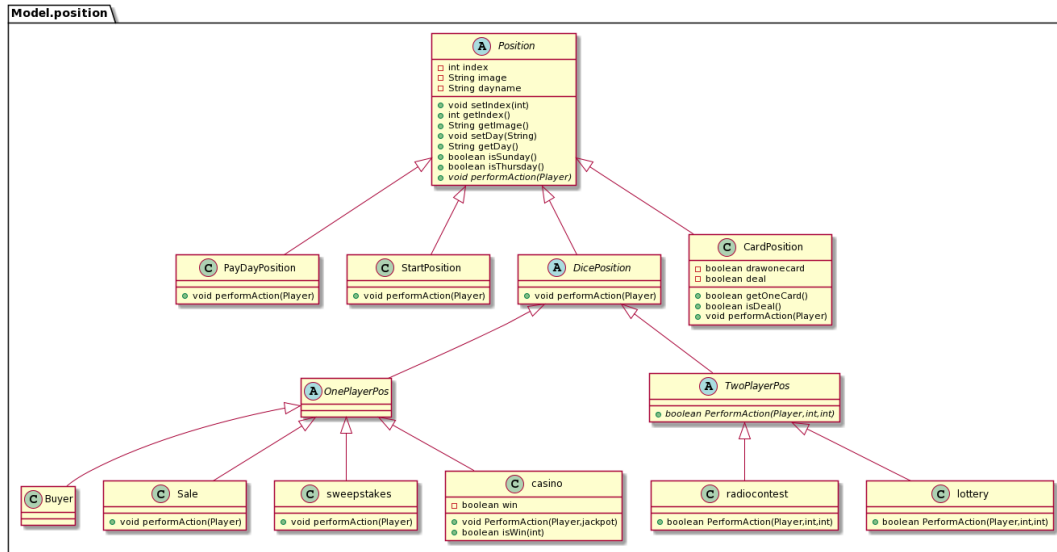
`public abstract class Cards` : Superclass όλων των καρτών,περιέχει πληροφορίες,όπως η εικόνα της κάρτας,πόσα λεφτά αξίζει και το μήνυμα που εμφανίζει(χαρακτηριστικά που κάθε κάρτα έχει)και τον τίτλο της κάρτας.Περιέχει μεθόδους `get` για κάθε πεδίο της.Ο constructor αρχικοποιεί τα πεδία με βάση του παραμέτρους που δέχεται.Τέλος έχει και μια `abstract` μέθοδο `PerformAction` που μπορεί να χρησιμοποιηθεί από τα subclasses για τα action που κάνουν.

Άμεσα subclasses της `Cards` είναι :

- `public class DealCard extends Cards` : περιέχει ένα ακόμα πεδίο `sellprice`,που είναι η τιμή πώλησης της κάρτας και προφανώς μια `get method`.Στο `PerformAction` μειώνει τα λεφτα του παίκτη αν την αγοράσει και προσθετει την κάρτα στην συλλογή του.
- `public abstract class MailCard extends Cards` : είναι `abstract` καθώς υπάρχουν πολλών ειδών κάρτες μηνυμάτων.Περιέχει το `String buttonmsg` (προστέθηκε στη Β φάση) το οποίο περιέχει το μήνυμα για το button.Έχει τις παρακάτω υποκλάσεις:
- `public class Advertisement extends MailCard`
- `public class Bill extends MailCard`
- `public class Charity extends MailCard`
- `public class MoveTo extends MailCard`
- `public class PaytheNeighbor extends MailCard`
- `public class TakefromNeighbor extends MailCard`

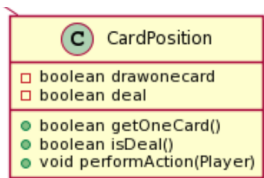
Οι οποίες περιέχουν έναν constructor (ίδιο με της κλασης `Cards`),και οι κλάσεις `Advertisement`,`Bill`, `PaytheNeighbor`, `TakefromNeighbor` χρησιμοποιουν το `PerformAction`,το οποίο εκτελεί ό,τι κάνει η συγκεκριμένη κάρτα.Η `PerformAction` είναι ευκολία στον controller όταν ελεγχουμε τι είδος καρτας είναι αυτή,να μην χρειάζεται να γράφουμε εκεί τις εντολές που πρέπει να εκτελεστούν.Σε κάποιες κλασεις ωστόσο αυτο δεν μπορεί να αποφευχθεί και για αυτο δεν κάνουν `Override` την `Perform Action`.

Position

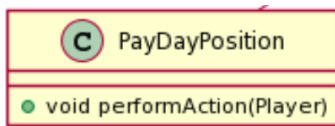


Η κλάση Position είναι abstract καθώς υπάρχουν πολλά είδη position. Αναλυτικότερα
 Μια κλάση Position έχει τα εξής πεδία: το αριθμό της θέσης, το όνομα της ημέρας και το URL της εικόνας. Ο Constructor αρχικοποιεί το index και το imageUrl. Υπάρχουν τόσο setters όσο και getters για όλα τα πεδία. Τέλος υπάρχουν 2 μέθοδοι που ελέγχουν αν είναι Κυριακή (για τον αγώνα) και Πέμπτη (για το Crypto), ενώ υπάρχει και μια abstract method PerformAction(Player p) που χρησιμοποιείται για να κάνει τις κατάλληλες εντολές (Player p=ο παίκτης που βρίσκεται στη θέση αυτή).

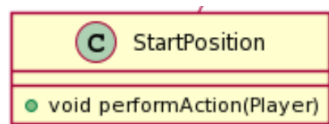
Subclasses του Position:



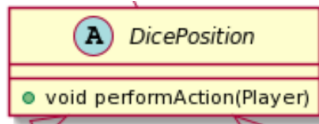
- public class CardPosition extends Position: Περιέχει δύο ακόμη πεδία, το boolean drawOneCard, το οποίο είναι final και έχει μόνο getter και το boolean deal το οποίο είναι αυτο final και έχει μόνο getter. Χρησιμοποιείται ο constructor της superclass και το PerformAction δεν κάνει κάτι εδώ (τα action του θα υλοποιηθεί στον Controller).



- public class PayDayPosition extends Position : Ο constructor αρχικοποιεί την θέση πάντα να είναι η τελευταία.

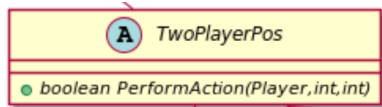


- `public class StartPosition extends Position` : Παρομοιο με `Position` με την μόνη διαφορά να αρχικοποιεί το `index=0`.

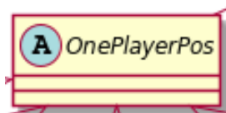


- `public abstract class DicePosition extends Position` : Παρομοιο με `Position`

Τα subclasses του `DicePosition` είναι:



- `public abstract class TwoPlayerPos extends DicePosition` : Αποτελεί superclass για τις θέσεις που αφορούν 2 άτομα. Έχει μια `public abstract boolean PerformAction(Player p, int n1, int n2)`, η οποία δέχεται τον παίκτη που βρίσκεται στη θέση αυτή και `n1` και `n2` είναι οι επιλογές των παικτών. Τα subclasses της είναι:
 1. `public class lottery extends TwoPlayerPos` : Κάνει override την `PerformAction`. Εδώ το `n1` και το `n2` είναι οι αριθμοί που επέλεξαν οι 2 παίκτες.
 2. `public class radiocontest extends TwoPlayerPos` : Κάνει override την `PerformAction`. Εδώ το `n1` και το `n2` είναι οι αριθμοί των ζαριών των 2 παικτών.

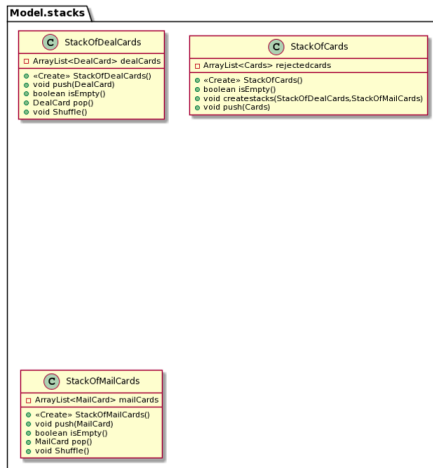


- `public abstract class OnePlayerPos extends DicePosition` : Δεν έχει κάποιο νέο χαρακτηριστικό. Αποτελεί superclass των εξής κλάσεων:
 1. `public class Buyer extends OnePlayerPos`: Δεν έχει κάποιο νέο χαρακτηριστικό
 2. `public class casino extends OnePlayerPos`: Έχει ένα ακόμα attribute, την `boolean` μεταβλητή `win`, η οποία μας δείχνει αν έχει κερδίσει ο παίκτης ή όχι. Ο constructor όπως και η superclass. Μια μέθοδος `PerformAction(Player p, jackpot j)` που εκτελεί τις κατάλληλες εντολές (συμφωνα με το τι κάνει κάποιος στο casino και τα λεφτά που μπορεί να δώσει στο jackpot). Και μια μέθοδος ακόμη `isWin` που με βάση τον αριθμό του ζαριού που δέχεται ως παράμετρο, ενημερώνει το `win` (αποφασίζει δηλαδή αν νίκησε ή όχι).

3. `public class Sale extends OnePlayerPos`: Χρησιμοποιεί τον default constructor (δηλαδή της superclass) και επίσης κάνει override την `PerformAction` της `OnePlayerPos`. Στο `Perform action` θα εκτελούνται οι εντολές όπως για παράδειγμα να μειωθούν τα λεφτά του παίκτη κτλ.

4. `public class sweepstakes extends OnePlayerPos`: Χρησιμοποιεί τον default constructor (δηλαδή της superclass) και επίσης κάνει override την `PerformAction` της `OnePlayerPos`. Στο `Perform action` θα εκτελούνται οι εντολές όπως για παράδειγμα να αυξηθούν τα λεφτά του παίκτη κτλ.

Stacks



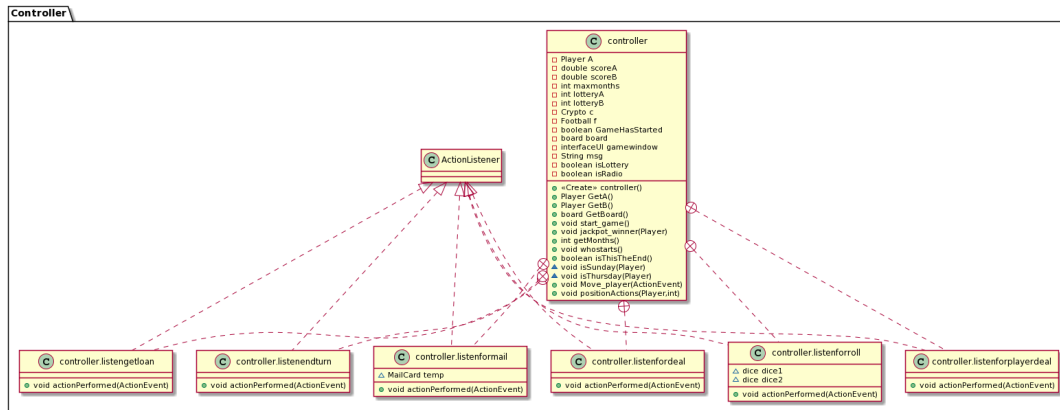
Τελος,εχουμε τις κλάσεις Stacks για τις κάρτες.Αυτές είναι:

- 1.public class StackOfCards : χρησιμοποιείται για τις κάρτες που απορρίψαμε
- 2.public class StackOfDealCards : χρησιμοποιείται για τις κάρτες συμφωνίας
- 3.public class StackOfMailCards : χρησιμοποιείται για τις κάρτες μηνυμάτων

Υλοποιουν στοιβες καρτών.Κάθε μια τους έχει ένα ArrayList<Cards>.Οι 2 τελευταίες εχουν παρομοιες συναρτησεις οπως pop(αφαιρει στοιχειο και το επιστρέφει),push(προσθετει στοιχείο στην στοιβα),isEmpty(ελέγχει αν είναι άδεια) και Shuffle(που ανακατεύει τις κάρτες της στοιβας).Η StackofCards δεν εχει Shuffle(αφού

Ποτέ δεν ανακατεύουμε τις κάρτες αυτές),δεν εχει pop(αφου ποτε δεν παίρνουμε κάρτα από αυτήν την στοιβα) και τελος έχει μια μέθοδο που δημιουργεί τις άλλες 2 στοιβες(στο τελος πρεπει να είναι άδεια η στοιβα μας).

3. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller Controller



Το πακετο του Controller αποτελείται από μία κλάση controller και τους

και τους ActionListeners για τα κουμπιά του interface. Αναλυτικότερα, η κλάση

Controller περιέχει τα πεδία:

Player A,B : οι παίκτες του παιχνιδιού

int lotteryA, lotteryB, maxmonths : το στοιχημα των 2 παικτών (added in B phase) στο lottery και οι μήνες που θα διαρκέσει το παιχνίδι.

double scoreA, scoreB : το σκορ των 2 παικτών.

Crypto c : ένα event crypto .

Football f : ένα event football.

boolean GameHasStarted, isLottery, isRadio : “σημαίες” για το αν ξεκίνησε το παιχνίδι ή όχι, αν είναι lottery position ή radio (the last two were added in phase B).

board board : το “ταμπλο” μας.

interfaceUI gamewindow : το γραφικό μας περιβάλλον.

String msg : το μήνυμα που θα δίνεται στο gamewindow για να το προβάλλει στο InfoBox.

Ο constructor του controller , αρχικοποιεί τα πεδία και χρησιμοποιεί UI κλάση για να εμφανιστεί JOptionPane για να διαλέξουν όνομα οι παίκτες και να μπορεί να τους δημιουργήσει. Αφού καλέσει την whostarts για να αποφασίσει ποιος παίκτης θα ξεκινήσει, μετά καλεί μια συνάρτηση αρχικοποίησης, την Start_game. Η μεθοδος Start_game ξεκινά το παιχνίδι.

Η μεθοδος jackpot_winner ενημερώνει τα λεφτα του παίκτη που νίκησε.

Η μεθοδος winner επιστρέφει το ονομα του νικητή jackpot.

Η μεθοδος whostarts ρίχνει τα ζάρια για κάθε παίκτη και αυτος που έχει μεγαλύτερο αριθμό τα έχει hasTurn=true.

Η μεθοδος `isthisTheEnd` ελεγχει αν `maxmonths==0` η όχι.

Τελος, στον controller εχουμε τις εξης inner κλάσεις,που είναι subclasses του `ActionListener` και χρησιμοποιούνται για τα κουμπια του interface:

`listenforroll` implements `ActionListener` --> για το dice button

`listenforplayerdeal` implements `ActionListener` --> για το deal card button του παίκτη

`listengetloan` implements `ActionListener` --> για το "Get a loan" button

`listenendturn` implements `ActionListener` --> για το "End Turn" button

`listenformail` implements `ActionListener` --> για το "Mail card" button

`listenfordeal` implements `ActionListener` --> για το "Deal card" button

Στην Β φαση αφαιρέθηκαν τα εξής :

- `Menulistener` implements `ActionListener`--> για το menu button
- `listenfornewgame` implements `ActionListener` --> για το "New Game" button
- Η μεθοδος `UpdateMonths` ενημερώνει τους μήνες και των 2 παικτών.
- Η μεθοδος `newgame` δημιουργεί ένα νεο controller και εξαφανίζει το παλιό `interfaceUI`.
- Η μέθοδος `RejectLastCards` με παράμετρο `Player p`, αδειάζει τα card stack του `p`. Αφαιρέθηκε στην β φαση
- Η μέθοδος `gameStatus` επιστρέφει ένα πίνακα από `String` για να χρησιμοποιηθει απο το `interfaceUI` στο `InfoBox`.

Ενώ προστέθηκαν τα εξής :

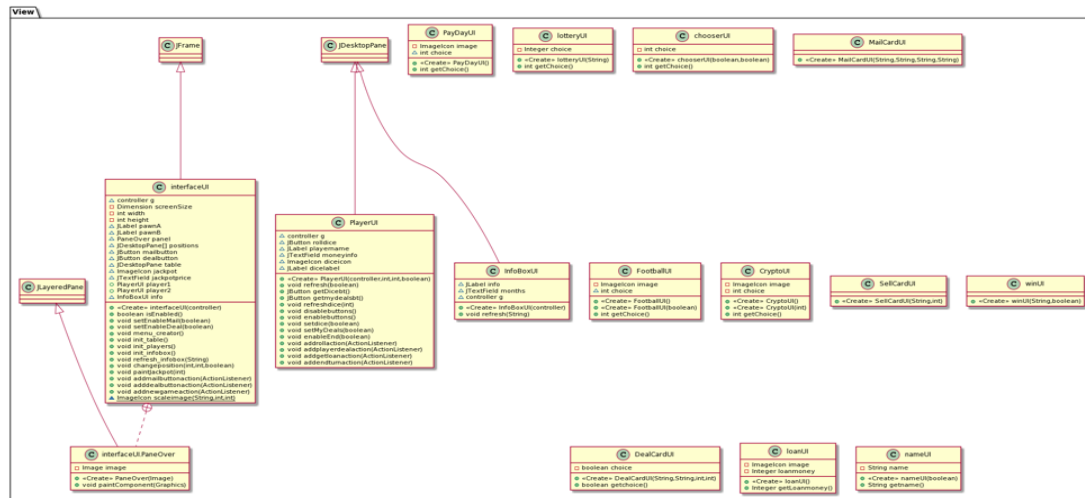
- Getters για τους `Players` και το `board`,επειδή χρειάζονται στο `View` και πρέπει να τηρούνται οι κανονισμοί του MVC.
- `void isSunday(Player)`,κάνει τις απαραίτητες ενέργειες αν είναι Κυριακή
- `void isThursday(Player)`,κάνει τις απαραίτητες ενέργειες αν είναι Τρίτη
- `void Move_player(ActionEvent e)`,μετακινεί τον παίκτη
- `void positionActions(Player,pos)`,κάνει τις απαραίτητες ενέργειες ανάλογα το `Position`

Παρατηρούμε ότι υπάρχει μια άμεση σύνδεση και αλληλεπίδραση του controller,του model και του view. Το controller περιέχει πεδία που είναι κλάσεις από το model. Αυτό σημεινει(όπως έχουμε πει και στην εισαγωγή)οτι τα αντικείμενα τύπου `Model` χρησιμοποιούνται απο το controller για την αποθήκευση σημαντικών πληροφοριών. Τα αντικείμενα που είναι απο το

model, έχουν αποθηκεύσει οργανωμένα τις πληροφορίες ενώ παράλληλα έχουν διάφορα methods που διευκολύνουν το controller σε κάποιες περιπτώσεις. Παρατηρείται επίσης ότι ο controller χρησιμοποιεί το interface UI αλλά και αλλα class του view, γεγονός που σημαίνει ότι αλληλεπιδρά και με το view package (αφού και το view package του επιστρέφει τιμές από το input του χρήστη).

4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View

Το package view περιέχει όλες τις κλάσεις που αφορούν το γραφικό περιβάλλον του παιχνιδιού.



InterfaceUI

Αρχικά, το main class για το γραφικό περιβάλλον είναι το InterfaceUI το οποιοστο οποίο είναι υπεύθυνο να έχει τα πιόνια,τον πίνακα θέσεων,τα κουμπια για mail και deal card,το menu,το info box,το jackpot και τέλος τις πληροφορίες των 2 player.Ο constructor θα καλεί τις μεθόδους αρχικοποίησης ενώ παράλληλα θα αρχικοποιούνται κάποια πεδία του και θα εμφανίζονται στο frame.Επίσης αυτή η κλάση περιέχει μεθόδους για να ανανεώνει το γραφικό της περιβαλλον.Παρατηρούμε επίσης ότι έχει και μεθοδους που προσθέτουν έναν ActionListener για κάθε κουμπί(οπου η υλοποιηση του είναι στο controller).Τελος έχει μια static method που θα χρησιμοποιείται για να κάνει scale μια εικόνα και να επιστρέφει ImageIcon.

PlayerUI & InfoBoxUI

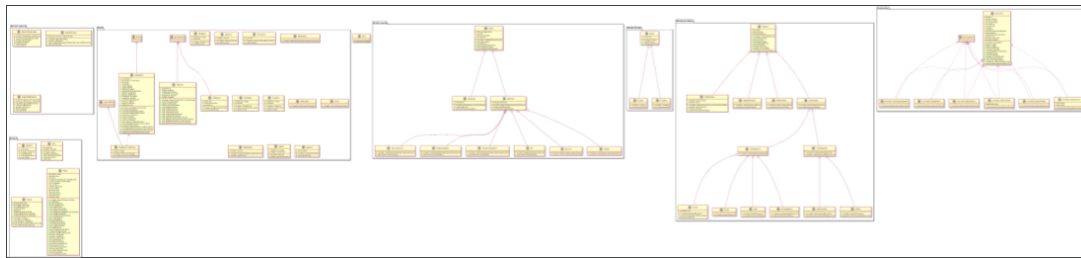
Τα επομένα ,2 class είναι εξίσου σημαντική, αφού αποτελούν κύρια components του Interface UI.Αυτά τα classes είναι το PlayerUI και το InfoBox.Είναι subclasses του JDesktopPane και JPanel αντιστοιχα ,που σημαίνει ότι κληρονομούν τις μεθόδους αυτών των κλάσεων.Εκτος όμως από αυτές που κληρονομούν,εχουν και οι δύο έναν constructor που τις αρχικοποιει και μια refresh μεθοδο που τις ενημερώνουν.Το Player UI επίσης έχει κουμπια και μια εικόνα ζαριου,επομενως εχει μεθοδους που προσθέτουν έναν ActionListener για κάθε κουμπί,και refresh για την εικόνα του ζαριού.



Pop-ups

Τα υπόλοιπα classes δημιουργούν ένα JOptionPane και μέσω της method get το controller μπορεί να πάρει το input του χρηστή.Επίσης μιας και δημιουργούνΤα JOptionPane στους constructors,κάποιες κλάσεις έχουν 2ο constructor με παραμετρο και δημιουργούν το κατάλληλο JOptionPane(πχ FootballUI και CryptoUI).

5. Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML



Στην παραπάνω εικόνα(αν και δεν φαίνεται καθαρά) είναι το διάγραμμα όλων των πακέτων που υπάρχουν στο παιχνίδι. Οι κλάσεις σίγουρα αλληλεπιδρούν μεταξύ τους ,αφου υπάρχουν ως πεδια μεσα σε άλλες.Ανταλλάσσουν μεταξύ τους πληροφορίες ,και αλλάζουν τα δεδομένα τους.Στο παραπάνω σχήμα φαίνεται η κληρονομικότητα μεταξύ των κλάσεων.Το βελάκι σημαίνει δείχνει πάντα το superclass ενώ το βελάκι με τον σταυρό (+) δείχνει ότι αυτή η κλάση είναι inner class

6. Λειτουργικότητα (B Φάση)

Υλοποιήθηκαν όλα τα ερωτήματα της εκφώνησης.Παρόλα αυτά δεν υλοποιήθηκαν κάποια πράγματα που θα ήθελα και θα έκαναν το παιχνίδι πιο ωραίο ,οπως η ύπαρξη Menu για new game,load game(ήταν και στο bonus),pop up στο τέλος που να ρωτά αν υπάρξει new game ή οχι και τέλος η προσθήκη μουσικής.

7. Συμπεράσματα

Η εργασία ήταν αρκετά ικανοποιητική και σου “μάθαινε” να χρησιμοποιεις σε ένα java swing.Η υλοποίηση του MVC σε java swing είναι κάπως περίεργη αφου το view πρέπει να “δρα” και σαν controller.Χρειάστηκαν έτσι κάποιες αλλαγές στην Α φάση.Στην Β φάση έχω στο view ένα controller object ώστε να μην υπάρχει άμεση επικοινωνία model και view και έτσι να τηρούνται οι κανόνες του MVC.Τέλος να σημειώσω πως προσπάθησα να δημιουργήσω το jar file για το παιχνίδι αλλά φαίνεται πως υπάρχει κάποιο πρόβλημα με τον βοηθητικό κώδικα ReadFile(που διαβάζει τα csv αρχεία) και εμφανίζει Null Pointer exception(ωστόσο το πρόγραμμα τρέχει κανονικά αν το τρέξει κανείς μέσω του IDE).