



Πανεπιστήμιο Κρήτης –Τμήμα Επιστήμης Υπολογιστών

ΗΥ252– Αντικειμενοστρεφής Προγραμματισμός

Διδάσκων: Ι. Τζιτζικας

Χειμερινό Εξάμηνο 2020-2021

PAYDAY

Εισαγωγή

Κωνσταντίνος Δελής

Csd4623

8/12/2021

Περιεχόμενα

1. Εισαγωγή.....	2
2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model	2
3. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller.....	18
4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View.....	22
5. Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML.....	26
6. Λειτουργικότητα (Β Φάση).....	28
7. Συμπεράσματα	28

1. Εισαγωγή

Η υλοποίηση της εργασίας θα βασιστεί πάνω στο μοντέλο MVC (Model View Controller). Έτσι, σκοπός είναι ο Controller να είναι ο συνδετικός κρίκος των Model και View. Οπότε στη συνέχεια της αναφοράς γίνει η ανάλυση κυρίως των κομματιών Model και Controller και στο τέλος θα γίνει μία πιο γενικευμένη αναφορά στο κομμάτι View.

2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model

PACKAGE MODEL

Σε αυτό το πακέτο υπάρχουν οι κλάσεις BankTransactions, Board, Dice, Point, Jackpot και Player που υλοποιεί την διεπαφή PlayerADT.

PlayerADT

Το «συμβόλαιο» που περιέχει τις υπογραφές από τις μεθόδους του κάθε παίκτη

Το interface αυτό μας παρέχει τις εξής μεθόδους:

addBills(int)	Transformer: Προσθέτει ή αφαιρεί λογαριασμούς από τον παίκτη
addCard(DealCard)	Transformer: Προσθέτει μια κάρτα στο Inventory του παίκτη
addLoan(int)	Transformer: Προσθέτει ή αφαιρεί Χρέι από τον παίκτη
addMoney(int)	Transformer: Προσθέτει ή αφαιρεί λεφτά από τον παίκτη
calculateScore()	Accessor: Υπολογίζει τους βαθμούς ενός παίκτη
getBills()	Accessor: Επιστρέφει τους λογαριασμούς του παίκτη
getCardInventory()	Accessor: Επιστρέφει το Inventory με τις κάρτες του παίκτη
getCurrentTile()	Accessor: Επιστρέφει το Tile στο οποίο βρίσκεται ο παίκτης

getLoans()	Accessor: Επιστρέφει τα χρέι του παίκτη
getMoney()	Accessor: Επιστρέφει τα λεφτά του παίκτη
getName()	Accessor: Επιστρέφει το όνομα του παίκτη
hasMove()	Observer: Επιστρέφει true αν ο παίκτης έχει κουνηθεί σε αυτόν τον γύρο, αλλιώς false
hasWon()	Observer: Επιστρέφει true αν ο παίκτης έχει τελειώσει το παιχνίδι, αλλιώς false
isMyTurn	Observer: Επιστρέφει true αν ο παίκτης έχει τελειώσει η δικιά του σειρά, αλλιώς false
ongoingAction	Observer: Επιστρέφει true αν ο παίκτης εκτελεί κάποια δράση εκείνη την στιγμή, αλλιώς false
setCurrentTile(Tile)	Transformer: Θέτει το Tile του παίκτη σε αυτό που έχει δοθεί στην παράμετρο της μεθόδου
setMoveStatus()	Transformer: Θέτει το moveStatus του παίκτη σε αυτό που έχει δοθεί στην παράμετρο
setOngoingAction()	Transformer: Θέτει το ongoingAction του παίκτη σε αυτό που έχει δοθεί στην παράμετρο
setTurnStatus(boolean)	Transformer: Θέτει το turnStatus του παίκτη σε αυτό που έχει δοθεί στην παράμετρο
setWinStatus(boolean)	Transformer: Θέτει το winStatus του παίκτη σε αυτό που έχει δοθεί στην παράμετρο
toString()	Επιστρέφει την αναπαράσταση του παίκτη σε μορφή String

Player

Κλάση που υλοποιεί την διεπαφή PlayerADT. Δεν υπάρχει κάποια επιπλέον μέθοδος σε σχέση με αυτές που έχουν δηλωθεί στο PlayerADT ήδη, εκτός από τον constructor.

Attributes

private Tile currentTile	Αποθηκεύει το Tile στο οποίο βρίσκεται αυτή την στιγμή ο παίκτης
private String name	Το όνομα του παίκτη
private ArrayList<Card> inventory	Ένα ArrayList που έχει όλες τις κάρτες του παίκτη
private boolean won	True αν ο παίκτης έχει τελειώσει το παιχνίδι, αλλιώς False
private boolean myTurn	True αν ο παίκτης έχει τελειώσει την σειρά του, αλλιώς False
private boolean ongoingAction	True αν ο παίκτης δεν έχει ολοκληρώσει κάποια δράση, αλλιώς False
private boolean moved	True αν ο παίκτης έχει κουνηθεί σε αυτόν τον γύρο, αλλιώς False
private int money	Τα λεφτά του παίκτη
private int bills	Οι λογαριασμοί του παίκτη
private int loans	Τα χρέι του παίκτη

Υπόλοιπες μέθοδοι

public Player(String name, int initialMoney)	Constructor: Φτιάχνει έναν καινούριο παίκτη και του δίνει να αρχικά του λεφτά και το όνομά του
--	--

Point

Η κλάση κρατάει τις x,y συντεταγμένες ενός σημείου

Attributes

private int y	Η x συντεταγμένη
---------------	------------------

private int x	Η y συντεταγμένη
---------------	------------------

Μέθοδοι

Point (int, int)	Constructor: Δημιουργεί ένα νέο Point με x και y συντεταγμένες αυτές των παραμέτρων
getXcoordinate()	Accessor: Επιστρέφει την x συντεταγμένη
getYcoordinate()	Accessor: Επιστρέφει την y συντεταγμένη
setXcoordinate(int)	Transformer: Θέτει την x συντεταγμένη
setYcoordinate(int)	Transformer: Θέτει την y συντεταγμένη

Dice

Αυτή η κλάση στην ουσία φτιάχνει ένα ζάρι και με καταλληλη μέθοδο επιστρέφει έναν αριθμό απο το 1 έως το 6

Attributes

private int previousRoll	Κρατάει τον αριθμό που ετυχε στην τελευταία ρίψη του ζαριού
private Player owner	Ο ιδιοκτήτης του ζαριού/null αν δεν ανήκει σε κανέναν
private String imagePath	Το path της εικόνας του ζαριού

Μέθοδοι

Dice()	Constructor: Δημιουργεί ένα νέο Dice με null για ιδιοκτητή, previousRoll=1 και imagePath την εικόνα που το ζάρι έχει φέρει 1
Dice(Player)	Constructor: Δημιουργεί ένα νέο Dice με ιδιοκτητή τον Player της παραμέτρου, previousRoll=1 και imagePath την εικόνα που το ζάρι έχει φέρει 1
getImagePath()	Accessor: Επιστρέφει το path της εικόνας

getOwner()	Accessor: Επιστρέφει τον ιδιοκτήτη/null αν δεν έχει ιδιοκτήτη
getPreviousRoll()	Accessor: Επιστρέφει την προηγούμενη ζαριά
rollDice()	Transformer: Επιλέγει τυχαία έναν αριθμό από το 1 έως το 6, καλεί την setImagePath με παράμετρο τον αριθμό αυτό και μετά επιστρέφει τον αριθμό
Private setImagePath(int)	Private μέθοδος που καλεί η μέθοδος rollDice() Transformer: Θέτει το image path του ζαριού ώστε η εικόνα να αντιστοιχεί στον αριθμό που έγινε roll
setOwner(Player)	Transformer: Θέτει τον ιδιοκτήτη του ζαριού

Board

Η κλάση αυτή διαχειρίζεται ένα ArrayList από Tiles, που στην ουσία είναι το ταμπλό του παιχνιδιού

Attributes

private ArrayList<Tile> board	Ένα ArrayList από κλάσεις Tile που στην ουσία είναι το ταμπλό του παιχνιδιού
-------------------------------	--

Μέθοδοι

Board()	Constructor: Δημιουργεί και αρχικοποιεί το ταμπλό του παιχνιδιού. Οι 33 θέσεις του θα αποτελούνται από τα εξής Tiles:
---------	---

	<ul style="list-style-type: none"> • 1 StartTile (παντα στην πρώτη θέση), • 1 PaydayTile (παντα στην τελευταία θέση), • 8 MessageTiles (τυχαία θέση), • 5 DealTiles (τυχαία θέση), • 2 SweepStakesTiles (τυχαία θέση), • 3 LotteryTiles (τυχαία θέση), • 2 RadioTiles (τυχαία θέση), • 6 BuyerTiles (τυχαία θέση), • 2 FamilyCasinoTiles (τυχαία θέση), • 2 YardSaleTiles (τυχαία θέση)
getBoard()	Accessor: Επιστρέφει το ArrayList με τα Tiles
getClosestDealBuyer(Tile)	Accessor: Απο ένα δεδομένο Tile υπολογίζει το κοντινότερο Tile που είναι είτε DealTile είτε BuyerTile και το επιστρέφει. Αυτή η μέθοδος θα χρησιμοποιείται απο την κάρτα που μετακινεί τον παίκτη στην κοντινότερη θέση συμφωνίας/αγοραστή
getTile(int)	Accessor: Επιστρέφει το i-οστό Tile του ταμπλό
Private assignOrderNumbers()	Μια private μεθοδος που χρησιμοποιείται απο τον constructor. Θετει τον αριθμό της σειράς κάθε Tile
Private assignThursdaysSundays()	Μια private μεθοδος που χρησιμοποιείται απο τον constructor. Θετει true το πεδιο isThursday όλων των Tile που βρίσκονται σε θέση που είναι Πέμπτη (Ομοιως και για την Κυριακή)
Private shuffleboard()	Μια private μεθοδος που χρησιμοποιείται απο τον constructor. Ανακατεύει τα Tiles εκτος απο το πρώτο και το τελευταίο

Private assignCoordinates()	Μια private μεθοδος που χρησιμοποιείται απο τον constructor. Θετει της συντεταγμένες του κάθε Tile
-----------------------------	--

Jackpot

Η κλάση αυτή μαζεύει λεφτά απο παίκτες και οταν ένας απο τους δυο παίκτες φέρει 6 σε ζαριά η σηκώσει μια κάρτα που του λέει πως κερδισε το jackpot θα πάρει ολα τα λεφτά που έχουν μαζευτεί και τα λεφτά θα μηδενιστούν (μεχρι να λαβει παλι λεφτα απο τους παίκτες)

Attributes

private int jackpotMoney	Τα λεφτά που έχει το Jackpot
--------------------------	------------------------------

Μέθοδοι

Jackpot()	Constructor: Φτιάχνει ενα καινούριο jackpot με αρχικά λεφτά 0
getJackpotMoney()	Accessor: Επιστρέφει τα λεφτά που έχει το jackpot
playerPaysJackpot(Player, int)	Transformer: ο παίκτης που δίνεται ως παράμετρος πληρώνει το jackpot ενα ποσό χρημάτων που δίνεται επίσης ως παράμετρος
wonJackpot(Player)	Transformer: ο παίκτης που δίνεται ως παράμετρος κερδίζει ολα τα λεφτα του jackpot και επομένως τα λεφτά του jackpot μηδενίζονται
Private giveNecessaryLoans(Player)	Μια private μέθοδος που απλά δίνει τα απαραίτητα δάνεια σε ένα παίκτη μετά απο μια συναλλαγή ωστε ο παίκτης να μην έχει αρνητικά χρήματα

BankTransactions

Η κλάση αυτή είναι βοηθητική και περιέχει μόνο στατικές μεθόδους που υλοποιούν μεταφορές χρημάτων μεταξυ παίκτη-παίκτη και παίκτη-τράπεζας

Στατικές Μέθοδοι

bankPaysPlayer(Player, int)	Transformer: Η τράπεζα πληρώνει τον παίκτη που δίνεται ως παράμετρος ενα ποσό χρημάτων που δίνεται επίσης ως παράμετρος
-----------------------------	---

playerPaysBank(Player, int)	Transformer: Ο παίκτης που δίνεται ως πληρώνει την τράπεζα ένα ποσό χρημάτων που δίνεται επίσης ως παράμετρος
playerPaysPlayer(Player, Player, int)	Transformer: Ο παίκτης που δίνεται ως παράμετρος πληρώνει τον αντίπαλο παίκτη ένα ποσό χρημάτων που δίνεται επίσης ως παράμετρος
Private giveNecessaryLoans(Player)	Μια private μέθοδος που απλά δίνει τα απαραίτητα δάνεια σε ένα παίκτη μετά απο μια συναλλαγή ώστε ο παίκτης να μην έχει αρνητικά χρήματα

PACKAGE MODEL.TILES

Tile

Μια abstract κλάση για τα Tiles του ταμπλό. Περιέχει τα βασικά πεδία και μεθόδους που χρειάζεται κάθε είδος Tile.

Attributes

private int orderNumber	Ο αριθμός του Tile στο ταμπλό (μεταξύ 0 και 32)
private String imagePath	Το path της εικόνας που αντιστοιχεί στο ταμπλό
private Point location	Οι συντεταγμένες στις οποίες βρίσκεται το Tile
private Boolean isThursday	True αν είναι Πέμπτη, αλλιώς False
private Boolean isSunday	True αν είναι Κυριακή, αλλιώς False

Μέθοδοι

Tile(String)	Constructor: Δημιουργεί ένα νέο Tile με imagePath που δίνεται ως παράμετρος
getImagePath()	Accessor: Επιστρέφει το path της εικόνας
getLocation()	Accessor: Επιστρέφει ένα Point που έχει τις συντεταγμένες του Tile

getOrderNumber()	Accessor: Επιστρέφει τον αριθμό της σειράς του Tile στο ταμπλό
isSunday()	Accessor: Επιστρέφει True αν είναι Κυριακή αλλιώς False
isThursday()	Accessor: Επιστρέφει True αν είναι Πέμπτη αλλιώς False
setImagePath(String)	Transformer: Θέτει το path της εικόνας του Tile
setIsSunday(Boolean)	Transformer: Θέτει το isSunday True αν είναι Κυριακή αλλιώς False
setIsThursday(Boolean)	Transformer: Θέτει το isThursday True αν είναι Κυριακή αλλιώς False
setLocation(Point)	Transformer: Θέτει τις συντεταγμένες του Tile
setOrderNumber(int)	Transformer: Θέτει τον αριθμό της σειράς του Tile στο ταμπλό
tileAction(Controller)	Abstract: Η λειτουργία θα διαφέρει για κάθε subclass του Tile, αλλά γενικά θα εκτελεί όλες τις ενέργειες που πρέπει να γίνουν όταν ο παίκτης φτάνει στο εκάστοτε είδος Tile
toString()	Επιστρέφει την αναπαράσταση του Tile σε String

BuyerTile extends Tile

Υποκλάση της Tile, χωρίς καινούρια attributes

Μέθοδοι

BuyerTile()	Constructor: Δημιουργεί ένα νέο BuyerTile και το αρχικοποιεί καλώντας τον Constructor της υπερκλάσης
tileAction(Controller)	Transformer: Αν ένας παίκτης φτάσει σε αυτό το Tile θα πρέπει να πουλήσει μία από τις κάρτες συμφωνίας του και στην συνέχεια να πραγματοποιηθούν όλες οι απαραίτητες μεταφορές χρημάτων. Σε

	περιπτωση που είναι Κυριακή ή Πέμπτη θα γίνουν και τα special events (Sunday match/ Thursday crypto)
--	--

DealTile extends Tile

Υποκλάση της Tile, χωρίς καινούρια attributes

Μέθοδοι

DealTile()	Constructor: Δημιουργεί ένα νέο DealTile και το αρχικοποιεί καλώντας τον Constructor της υπερκλάσης
tileAction(Controller)	Transformer: Αν ένας παίκτης φτάσει σε αυτό το Tile θα πρέπει να τραβήξει μια κάρτα συμφωνίας από την στοίβα και στην συνέχεια να πραγματοποιηθούν όλες οι απαραίτητες μεταφορές χρημάτων. Σε περίπτωση που είναι Κυριακή ή Πέμπτη θα γίνουν και τα special events (Sunday match/ Thursday crypto)

FamilyCasinoTile extends Tile

Υποκλάση της Tile, χωρίς καινούρια attributes

Μέθοδοι

FamilyTile()	Constructor: Δημιουργεί ένα νέο FamilyTile και το αρχικοποιεί καλώντας τον Constructor της υπερκλάσης
tileAction(Controller)	Transformer: Αν ένας παίκτης φτάσει σε αυτό το Tile θα πρέπει να λάβει 500\$ από την τράπεζα αν η πιο πρόσφατη ζαριά του ήταν ζυγός αριθμός, αλλιώς να πληρώσει να δώσει στο jackpot 500\$. Σε περίπτωση που είναι Κυριακή ή Πέμπτη θα γίνουν

	και τα special events (Sunday match/ Thursday crypto)
--	---

LotteryTile extends Tile

Υποκλάση της Tile, χωρίς καινούρια attributes

Μέθοδοι

LotteryTile()	Constructor: Δημιουργεί ένα νέο LotteryTile και το αρχικοποιεί καλώντας τον Constructor της υπερκλάσης
tileAction(Controller)	Transformer: Αν ένας παίκτης φτάσει σε αυτό το Tile θα πρέπει να διαλέξει έναν αριθμό από το 1 έως το 6. Ύστερα, διαλέγει και ο ανιπαλος έναν αριθμό και μετά γίνεται κλήρωση με ένα ζαρι μέχρι κάποιος να κερδίσει (1000\$ από την τράπεζα) και στην συνέχεια θα πραγματοποιηθούν όλες οι απαραίτητες μεταφορές χρημάτων. Σε περίπτωση που είναι Κυριακή ή Πέμπτη θα γίνουν και τα special events (Sunday match/ Thursday crypto)

MessageTile extends Tile

Υποκλάση της Tile

Επιπλέον Attributes

private final int numberOfMessageCards	Μπορεί να είναι είτε 1 είτε 2 ανάλογα με το πόσες κάρτες θα πρέπει να σηκώσει ο παίκτης
--	---

Μέθοδοι

MessageTile(int)	Constructor: Δημιουργεί ένα νέο MessageTile και το αρχικοποιεί καλώντας τον Constructor της υπερκλάσης. Επίσης θέτει το numberOfMessageCards
tileAction(Controller)	Transformer: Αν ένας παίκτης φτάσει σε αυτό το Tile θα πρέπει να τραβήξει μια (ή δύο) κάρτα/ες μυνημάτων απο την στοίβα και στην συνέχεια να πραγματοποιηθούν όλες οι απαραίτητες μεταφορές χρημάτων. Σε περίπτωση που είναι Κυριακή ή Πέμπτη θα γίνουν και τα special events (Sunday match/ Thursday crypto)

PaydayTile extends Tile

Υποκλάση της Tile, χωρίς καινούρια attributes

Μέθοδοι

PaydayTile()	Constructor: Δημιουργεί ένα νέο PaydayTile και το αρχικοποιεί καλώντας τον Constructor της υπερκλάσης
tileAction(Controller)	Transformer: Όταν ο παίκτης φτάσει σε αυτό το Tile έχει τελειώσει έναν μήνα παιχνιδιού και παίρνει 3500\$ και πληρώνει όλους τους λογαριασμούς του. Στην συνέχεια πληρώνει 10% απο τα χρέοι του στην τράπεζα και μετά μπορεί να ξεπληρώσει όλα ή κομμάτι απο τα χρέοι του. Τέλος, αν έχει τελειώσει όλους του μήνες του το παιχνίδι για αυτόν τελείωσε, αλλιώς πηγαίνει ξανά στην αρχή.

RadioTile extends Tile

Υποκλάση της Tile, χωρίς καινούρια attributes

Μέθοδοι

RadioTile()	Constructor: Δημιουργεί ένα νέο RadioTile και το αρχικοποιεί καλώντας τον Constructor της υπερκλάσης
tileAction(Controller)	Transformer: Αν ένας παίκτης φτάσει σε αυτό το Tile θα πρέπει να ρίξουν οι δύο παίκτες ζάρια (ξεκινάει αυτός που εφτάσε στο συγκεκριμένο Tile) και αυτός με την μεγαλύτερη ζαριά πληρώνει τον άλλο 1000\$. Σε περίπτωση που είναι Κυριακή ή Πέμπτη θα γίνουν και τα special events (Sunday match/ Thursday crypto)

StartTile extends Tile

Υποκλάση της Tile, χωρίς καινούρια attributes

Μέθοδοι

StartTile()	Constructor: Δημιουργεί ένα νέο StartTile και το αρχικοποιεί καλώντας τον Constructor της υπερκλάσης
tileAction(Controller)	Κενή μέθοδος/δεν υπάρχει κάποιο action σε αυτό το Tile

SweepTakesTile extends Tile

Υποκλάση της Tile, χωρίς καινούρια attributes

Μέθοδοι

SweepTakesTile()	Constructor: Δημιουργεί ένα νέο SweepTakesTile και το αρχικοποιεί καλώντας τον Constructor της υπερκλάσης
tileAction(Controller)	Transformer: Αν ένας παίκτης φτάσει σε αυτό το Tile θα πρέπει να ρίξει ένα ζάρι και η τράπεζα θα τον πληρώσει x1000 φορές την ζαριά του. Σε περίπτωση που είναι Κυριακή ή Πέμπτη θα γίνουν και τα special events (Sunday match/ Thursday crypto)

YardSaleTile extends Tile

Υποκλάση της Tile, χωρίς καινούρια attributes

Μέθοδοι

YardSaleTile()	Constructor: Δημιουργεί ένα νέο YardSaleTile και το αρχικοποιεί καλώντας τον Constructor της υπερκλάσης
tileAction(Controller)	Transformer: Αν ένας παίκτης φτάσει σε αυτό το Tile θα πρέπει να ρίξει ένα ζάρι και να πληρώσει x100 φορές την ζαριά στην τράπεζα. Σε περίπτωση που είναι Κυριακή ή Πέμπτη θα γίνουν και τα special events (Sunday match/ Thursday crypto)

PACKAGE MODEL.CARDS

Card

Είναι μία διεπαφή που θα υλοποιηθούν τα δυο είδη καρτών που υπάρχουν. Φτιάχνοντας τη διεπαφή Card μας δίνεται η δυνατότητα να προσπελάσουμε τα δεδομένα χωρίς να πρέπει να ορίσουμε αν μία κάρτα είναι κάρτα συμφωνίας ή μυνήματος.

Μέθοδοι

getCardNumber()	Accessor: Επιστέφει τον αριθμό της κάρτας (αριθμός της κάρτας είναι η σειρά στην οποία βρίσκεται στον πίνακα απο δεδομένα καρτών στην κλάση PayDayCards που δίνεται)
getInfo()	Accessor: Επιστέφει το String[] με τα δεδομένα της κάρτας
getType()	Accessor: Επιστέφει τον τύπο της κάρτας
getValue()	Accessor: Επιστέφει την αξία της κάρτας
setCardNumber(int)	Transformer: Θέτει τον αριθμό της κάρτας

<code>setInfo(String[])</code>	Transformer: Θέτει το πεδίο <code>String[]</code> της κάρτας
<code>toString()</code>	Επιστρέφει την αναπαράσταση της κάρτας σε <code>String</code>

MessageCard extends Card

Υλοποιεί την διεπαφή `Card`, και δεν περιέχει κάποια επιπλέον μέθοδο

Attributes

<code>private String[] info</code>	Για το <code>MessageCard</code> το μέγεθος του <code>String</code> θα είναι 6 και περιέχει τον Τίτλο, το είδος της κάρτας μυνήματος, το μήνυμα, την αξία της κάρτας και το όνομα την φωτογραφίας της
<code>private int cardNumber</code>	Ο αριθμός της σειράς της κάρτας στον πίνακα της κλάσης <code>PayDayCards</code> που δίνεται

DealCard extends Card

Υλοποιεί την διεπαφή `Card`

Attributes

<code>private String[] info</code>	Για το <code>DealCard</code> το μέγεθος του <code>String</code> θα είναι 8 και περιέχει τον Τίτλο, το είδος της κάρτας, το μήνυμα, την αξία αγοράς και πώλησης της κάρτας, το όνομα την φωτογραφίας της και τις δύο επιλογές που έχουμε
<code>private int cardNumber</code>	Ο αριθμός της σειράς της κάρτας στον πίνακα της κλάσης <code>PayDayCards</code> που δίνεται

Επιπλέον μέθοδο που έχει, πέραν αυτών που υλοποιεί από το `Card` interface

getCost()	Επιστρέφει το κόστος της κάρτας, δηλαδή ποσο χρειάζεται για να την αγοράσει κάποιος
-----------	---

DeckType

Enum που περιέχει τα είδη των στοιβών απο κάρτες που μπορούμε να έχουμε (DEAL, MESSAGE)

CardDeck

Μια κλάση που διαχειρίζεται ένα ArrayList απο κάρτες

Attributes

private ArrayList<Card> deck	ArrayList απο κάρτες, στην ουσία η στοιβα απο κάρτες
private int maxCapacity	Ο μέγιστος αριθμός καρτών (εξαρτάται απο τον τύπο του deck)

Μέθοδοι

CardDeck(ArrayList<Card>, DeckType)	Constructor: Δημιουργεί ένα νέο Deck και το αρχικοποιεί με τύπο DEAL ή MESSAGE (και αναλογα για DEAL, 20 το maxCapacity και για MESSAGE 48 το maxCapacity) και deck το ArrayList που δίνεται
CardDeck(DeckType)	Constructor: Δημιουργεί ένα νέο Deck και το αρχικοποιεί με τύπο DEAL ή MESSAGE και deck το ArrayList που δίνεται
addCardToBottom(Card)	Transformer: Προσθέτει μια κάρτα στη πρώτη θέση του ArrayList

getDeckSize()	Accessor: Επιστρέφει το μέγεθος του ArrayList
getTopCard()	Transformer/Accessor: Αφαιρεί απο το ArrayList την κάρτα που βρίσκεται στην τελευταία θέση και την επιστρέφει
isEmpty()	Observer: επιστρέφει True όταν το ArrayList είναι άδειο, αλλιώς False
shuffleDeck()	Transformer: Ανακατεύει την στοίβα
toString()	Επιστρέφει την αναπαράσταση του CardDeck σε String

3. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller

Controller

Αυτή η κλάση είναι ουσιαστικά το μυαλό του παιχνιδιού. Είναι υπεύθυνη για τη δημιουργία ενός νέου παιχνιδιού, τη δημιουργία στιγμιωτύπων παικτών, καρτών, στοιβών και του ταμπλό και προφανώς τη σύνδεση μεταξύ των γραφικών και του Model. Αυτό που κάνει η κλάση αυτή είναι να παίρνει τις επιλογές του χρήστη μέσω των γραφικών και αλλάζει τα δεδομένα των Models και ύστερα να ενημερώνει το View

Attributes

private Player player1, player2;	Οι παίκτες
private Board board;	Το ταμπλό
Private CardDeck dealDeck, messageDeck, dealRemoved, messageRemoved;	Οι στοίβες απο κάρτες
private Boolean gameFinished;	Δείχνει αν το παιχνίδι τελείωσε

private int player1month, player2month, gameMonths;	Οι μήνες που έχει τελειώσει ο κάθε παίκτης και οι μήνες που διαρκεί το παιχνίδι
private Dice player1dice, player2dice;	Τα ζάρια
private Jackpot jackpot;	Το jackpot
private GraphicalUI gui	Η βασική κλάση για τα γραφικά

Μέθοδοι

Controller()	Constructor: Φτιάχνει ένα νέο παιχνίδι, δηλαδή αρχικοποιεί κατάλληλα όλα τα attributes και καλεί τον constructor της κλάσης γραφικών
buyer()	Transformer: Αν ο παίκτης δεν έχει κάρτες συμφωνίας δεν γίνεται τίποτα, αλλιώς καλείται το CardInventoryUI και ο παίκτης επιλέγει μια από τις κάρτες του και την πουλάει
buyerCardSelected(DealCard)	Transformer: καλείται από το CardInventoryUI αφού έχει επιλεγεί μια κάρτα για πώληση και δίνονται τα λεφτά στον παίκτη
calculateScoreOfPlayer(Player)	Accessor: Υπολογίζει την βαθμολογία του παίκτη και την επιστρέφει
crypto()	Transformer: Καλεί το crypto popup window του GUI και αναλογα με την επιλογή του παίκτη και το αποτέλεσμα της τιμής του κρυπτονομισματος ο παίκτης πληρώνεται, χάνει λεφτά, ή δεν γίνεται τίποτα
drawDealCard()	Transformer: τραβάει μια κάρτα από το dealDeck, καλεί το showDealCard της κλάσης γραφικών PayDayCards και αναλογα με την επιλογή του χρήστη ή επιστρέφει την κάρτα ή την βάζει το dealRemoved και επιστρέφει null αν το χρήστη αγνοήσει την συμφωνία

drawMailCard()	Transformer: τραβάει μια κάρτα από το messageDeck, καλεί το showMailCard της κλάσης γραφικών PayDayCards και επιστρέφει την αξία και το είδος της κάρτας
family()	Transformer: Καλεί το φαμίλυ popup window του GUI και αναλογα με την προηγούμενη ζαριά του παίκτη είτε του αγαιρούνται χρήματα, είτε του προσθέτονται
getGameFinished()	Observer: επιστρέφει true αν το παιχνίδι τελείωσε, αλλιώς false
getJackpot()	Accessor: επιστρέφει το jackpot
lottery()	Transformer: Καλεί το lottery popup window του GUI, το οποίο επιστρέφει τον νικητή, στον οποίο προστιθονται χρήματα
movePlayerToDealerBuyer(Player)	Transformer: Καλεί το movePawnToTile του GUI και μεταφέρει το πιόνι στην κοντινότερη θέση Dealer/Buyer
payday()	Transformer: Καλεί το payday popup window του GUI και δίνονται 3500\$ στον παίκτη, αφαιρούνται από τα χρήματά του οι λογαριασμοί του και ένα επιτοκίο 10% από το δάνειό του. Του αφαιρούνται όλες οι κάρτες συμφωνίας και του δίνεται η δυνατότητα να ξεπληρώσει όλο ή μέρος από το δάνειό του
radio()	Transformer: Καλεί το radio popup window του GUI, το οποίο επιστρέφει τον νικητή, στον οποίο προστιθονται χρήματα
private refillDeal()	Transformer: γεμίζει το dealDeck χρησιμοποιώντας κάρτες που έχουν τοποθετηθεί στην στοίβα απορριψής dealRemoved
private refillMail()	Transformer: γεμίζει το messageDeck χρησιμοποιώντας κάρτες που έχουν τοποθετηθεί στην στοίβα απορριψής messageRemoved

sundayMatch()	Transformer: Καλεί το sunday popup window του GUI και αναλογα με την επιλογή του παίκτη και το αποτέλεσμα της τιμής του κρυπτονομισματος ο παίκτης πληρώνεται, χάνει λεφτα, ή δεν γίνεται τίποτα αν δεν ποντάρει
sweepstakes()	Transformer: Καλεί το sweepstakes popup window του GUI, το οποίο επιστρέφει έναν αριθμό απο το 1 εως το 6 και ο παίκτης πληρώνεται x1000 την ζαριά αυτή
whoseNotTurn()	Accessor: επιστρέφει τον παίκτη του οποίου δεν είναι η σειρά να παίξει
whoseTurn()	Accessor: επιστρέφει τον παίκτη του οποίου είναι η σειρά να παίξει
yardSale()	Transformer: Καλεί το sweepstakes popup window του GUI, το οποίο επιστρέφει έναν αριθμό απο το 1 εως το 6 και ο παίκτης πληρώνει x100 την ζαριά αυτή και προστίθεται στις κάρτες του μία κάρτα συμφωνίας

Εμφολευμένες Κλάσεις

DiceRollListener()	O ActionListener των JButton «Roll Dice» των δύο παικτών. Που περιέχει τις δράσεις που γίνονται όταν πατηθεί το καθε τέτοιο κουμπί
EndTurnListener()	O ActionListener των JButton «End Turn» των δύο παικτών. Που περιέχει τις δράσεις που γίνονται όταν πατηθεί το καθε τέτοιο κουμπί
GetCardsListener()	O ActionListener των JButton «Get Cards» των δύο παικτών. Που περιέχει τις δράσεις που γίνονται όταν πατηθεί το καθε τέτοιο κουμπί
GetLoanListener()	O ActionListener των JButton «Get Loan» των δύο παικτών. Που περιέχει τις δράσεις

	που γίνονται όταν πατηθεί το κάθε τέτοιο κουμπί
--	---

4. Η Σχεδίαση και οι Κλάσεις του Πακέτου View

Η πρώτη κλάση **GraphicUI** είναι η βασική κλάση των γραφικών που κάνει extend το JFrame και δημιουργεί το βασικό παράθυρο που έχει το ταμπλό, τις στοίβες με τις κάρτες και 3 panels ένα για τον κάθε παίκτη που μέσα βρίσκονται οι πληροφορίες (χρήματα, χρέοι, λογαριασμοί), τα κουμπιά και το ζάρι του κάθε παίκτη. Επίσης υπάρχει ένα panel που περιέχει βασικές πληροφορίες (πχ ποιανού η σειρά είναι και μια περιγραφή για τις πιο πρόσφατες ενέργειες που έγιναν).

Attributes

private JLabel p1data, p2data	JLabel με τις πληροφορίες των παικτών (λεφτα, δάνεια, λογαριασμοί)
private JLabel out	JLabel με το κείμενο του output JPanel
private JLabel jackpotMoney	JLabel που γράφει τα λεφτά που έχει μαζέψει το jackpot
private JLabel dice1, dice2	JLabel με την εικόνα των ζαριών
private JLabel pawn1, pawn2	JLabel με την εικόνα των πιονιών
private JButton p1rollDice, p2rollDice, p1getloan, p2getloan, p1endTurn, p2endTurn, p1cards, p2cards	Τα κουμπιά των παικτών

Μεθοδοι

GraphicUI(Board, Player, Player, Dice, Dice)	Constructor: Δημιουργεί μια γραφική αναπαράσταση του ταμπλό, τοποθετεί τα πιονία στην αρχή και φτιάχνει τα JButtons, JLabels και JPanels
changeDiceImage(Pawn, Dice)	Transformer: Αλλάζει την φωτογραφία του ζαριού για να δείχνει το

changeOutputPanelMessage(String)	Transformer: Αλλάζει το περιεχόμενο του output panel ώστε το νέο μήνυμα να είναι αυτό που περνάται ως παράμετρος
cryptoPopUp(Dice)	Ανοίγει ένα dialog και ο παίκτης αποφασίζει αν θα αγοράσει κρυπτονομισμα και ενημερώνεται αν κέρδισε ή όχι
familyPopUp(int)	Ανοίγει ένα dialog και ο παίκτης ενημερώνεται αν κέρδισε ή έχασε λεφτα
getP1cards()	Accessor: Επιστρέφει το p1cards JButton
getP1endTurn()	Accessor: Επιστρέφει το p1endTurn JButton
getP1getloan()	Accessor: Επιστρέφει το p1getloan JButton
getP1rollDice()	Accessor: Επιστρέφει το p1rollDice JButton
getP2cards()	Accessor: Επιστρέφει το p2cards JButton
getP2endTurn()	Accessor: Επιστρέφει το p2endTurn JButton
getP2getloan()	Accessor: Επιστρέφει το p2getloan JButton
getP2rollDice()	Accessor: Επιστρέφει το p2rollDice JButton
loanPopUp(int)	Ανοίγει ένα dialog και ο παίκτης ενημερώνεται αν προστέθηκαν δάνεια αυτόματα για να μην έχει αρνητικά λεφτά
lotteryPopUp(Player, Player, Dice)	Ανοίγει ένα dialog και ο κάθε παίκτης επιλέγει έναν αριθμό, και μετα ριχνουν ένα ζάρι και επιστρέφεται ο παίκτης που επέλεξε τον αριθμό πιο κοντά στη ζαριά
movePawnToTile(Pawn, Tile)	Μεταφέρει το πόνι στο Tile που δίνεται ως παράμετρος
paydayPopUp(Player)	Ανοίγει ένα dialog και αναφέρει στον παίκτη όλες τις δράσεις που θα γίνουν (πχ θα λαβει 3500\$, θα πληρώσει λογαριασμούς κτλ) και μετα του δίνει την

	δυνατοτητα να ξεπληρώσει το δάνειό του ή μέρος αυτού)
radioPopUp(Player, Player, Dice)	Οι παίκτες ριχνουν ένα ζάρι και αυτός με την υψηλότερη ζαριά είναι ο νικητής και τον επιστρέφει η συνάρτηση
refreshPlayerData(Player, Player, Jackpot)	Κάνει refresh τα δεδομένα των παικτών (χρηματα, λογαριασμούς, δανεια) και τα λεφτά του jackpot
sundayPopUp(Dice)	Ανοίγει ένα dialog και ο παίκτης αποφασίζει αν θα ποντάρει για το αποτέλεσμα του αγώνα και ενημερώνεται αν κέρδισε ή όχι
sweepstakesPopUp(Dice)	Οι ο παίκτης ρίχνει ένα ζάρι και η ζαριά του επιστρέφεται
winPopUp(int, int)	Ανοίγει ένας διάλογος που ενημερώνει του παίκτες πόσο ήταν το σκόρ τους και ποιός νίκησε, και αφού κλείσουν το παράθυρο το πρόγραμμα τερματίζει
wonJackpot(int)	Ο παίκτης ενημερώνεται ότι έφερε 6 σε ζαριά και νίκησε το jackpot
yardPopUp(Dice)	Ο ο παίκτης ρίχνει ένα ζάρι και η ζαριά του επιστρέφεται, και ενημερώνεται ότι κέρδισε μια κάρτα συμφωνίας
private getScaledImage(Image, int, int)	Επιστρέφει την εικόνα σε μέγεθος width x height που είναι οι int παράμετροι
private startMenuPopup()	Ανοίγει ένας διάλογος και οι παίκτες επιλέγουν το μήκος του παιχνιδιού

Η δεύτερη κλάση θα κάνει CardInventoryUI η οποία επίσης κάνει extend την JFrame και φτιάχνει ένα παράθυρο όπου δείχνει όλες τις κάρτες που έχει στην κατοχή του ο παίκτης.

Attributes

private ArrayList<DealCard> inventory	Οι κάρτες
private JList list	Ενα JList που περιέχει τις κάρτες με τις εικόνες τους
private DealCard selected	Η κάρτα που έχει επιλεγεί

Μέθοδοι

<code>CardInventoryUI(ArrayList<DealCard>)</code>	Constructor: Δημιουργεί ένα <code>JList</code> με τις κάρτες με τις εικόνες τους χρησιμοποιώντας το <code>ArrayList</code> με το <code>inventory</code> του παίκτη που δίνεται ως παράμετρος
<code>buyerMode(Controller)</code>	Accessor: Σε αυτό το mode ο παίκτης κάνει κλικ σε μια από τις κάρτες του και η κάρτα πωλείται μέσω της συνάρτησης <code>buyerCardSelected(Card)</code> του <code>Controller</code>
<code>getJFrame()</code>	Accessor: Επιστρέφει reference σε αυτή την κλάση
<code>viewer()</code>	Accessor: Σε αυτό το mode ο παίκτης απλά κοιτάζει τις κάρτες του χωρίς να μπορεί να κάνει κάτι

Χρησιμοποιεί την βοηθητική κλάση `IconList` που παράγει ένα `JList` που περιέχει και εικονίδια και το επιστρέφει

Enum Pawn

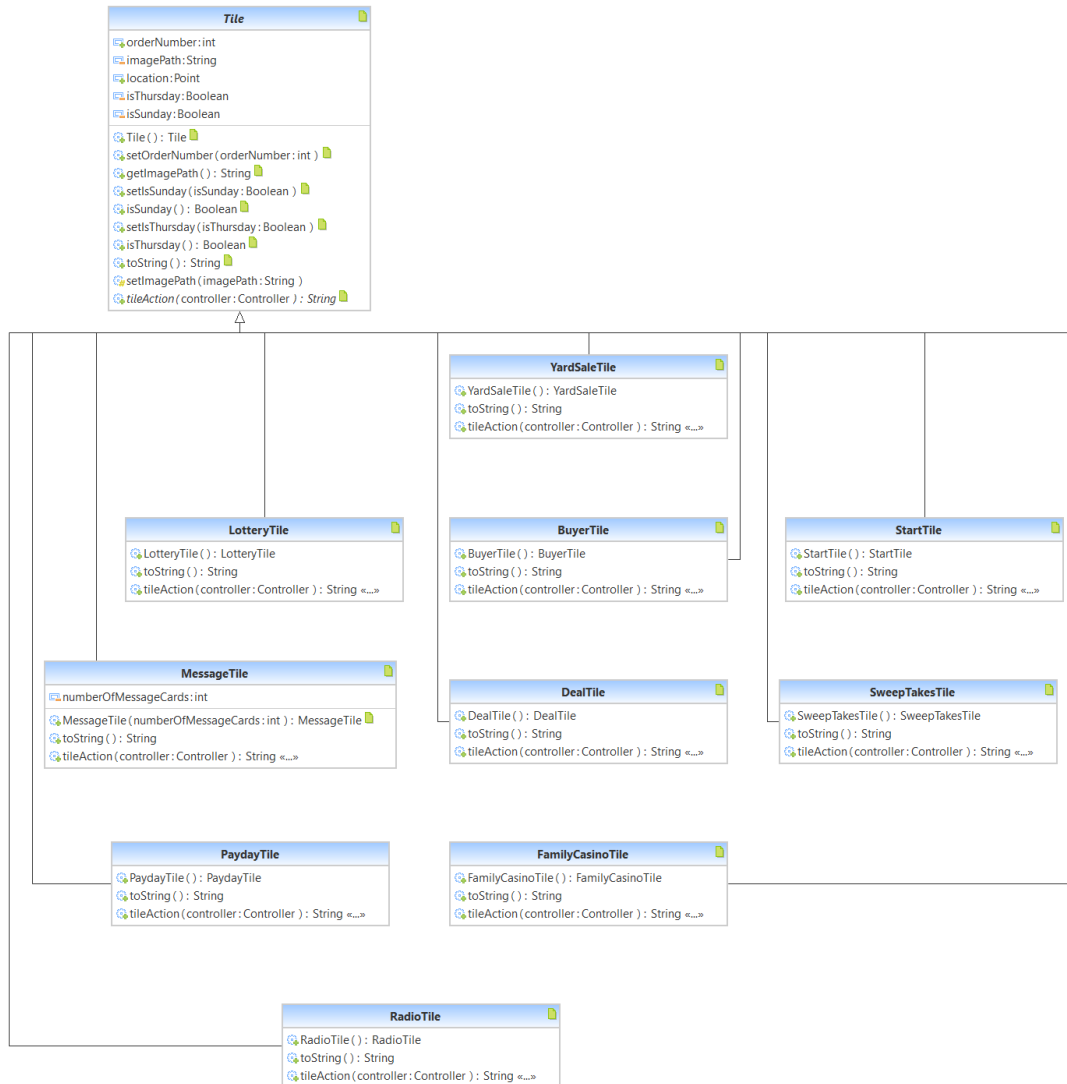
Έχει τιμές `p1` και `p2`

PayDayCards

Η κλάση που μας έχει δοθεί έτοιμη και χρησιμοποιείται για την δημιουργία των καρτών όσων αφορά το γραφικό περιβάλλον.

5. Η Αλληλεπίδραση μεταξύ των κλάσεων – Διαγράμματα UML

Πακέτο Model.Tiles: Παρακάτω φαίνονται όλα τα διάφορα είδη Tiles που υλοποιούν το Interface Tile



Πακέτο Model.Cards: παρακάτω φαίνονται τα δύο είδη καρτών που υλοποιούν το Interface Card. Επίσης, βλέπουμε πως η κλάση CardDeck χρησιμοποιεί και το enum DeckType, καθώς και το Interface Card (αφού αποθηκεύει σε ένα ArrayList οποιαδήποτε από τα δύο είδη καρτών)

6. Λειτουργικότητα (B Φάση)

Όλα τα ερωτήματα έχουν ολοκληρωθεί και το παιχνίδι λειτουργεί. Κατι που δεν ξερω αν είναι σωστό είναι ότι οι κάρτες ανοίγουν αυτόματα, χωρίς ο παίκτης να πατάει κατι. Γνωρίζω πως κάτι τέτοιο μπορεί να επιλυθεί με την υλοποίηση ενός `MouseListener` που να ενεργοποιείται όταν ο παίκτης πατάει πάνω στα `JLabel` που βρίσκονται τα εικονίδια των καρτών, ωστόσο λόγω υποχρεώσεων από άλλα μαθήματα και της εξεταστικής που πλησιάζει δεν βρήκα το χρόνο να αλλάξω το πρόγραμμα.

7. Συμπεράσματα

Νομίζω πως η εργασία ήταν πολύ χρήσιμη για την κατανόηση του τρόπου λειτουργίας ενός `object-oriented` προγράμματος, που στην ούσια δεν ακολουθεί μιά συγκεκριμένη και προκαθορισμένη πορεία (όπως οι ασκήσεις A1 και A2), αλλά η πορεία που εξελίσσεται το πρόγραμμα αλλάζει κάθε φορά με διαφορετικό τρόπο που εξαρτάται από τον χρήστη.