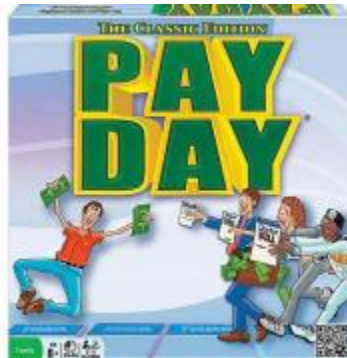


HY-252 PROJECT 2020-2021

Project



Stivaktakhs giorgos / csd4300

Περιεχόμενα

<u>1. Εισαγωγή</u>	1
<u>2. Η Σχεδίαση και οι Κλάσεις του Πακέτου Model</u>	1
<u>3. Η Σχεδίαση και οι Κλάσεις του Πακέτου View</u>	1
<u>4. Η Σχεδίαση και οι Κλάσεις του Πακέτου Controller</u>	2
<u>5. Η Αλληλεπίδραση μεταξύ των κλάσεων - Διαγράμματα UML</u>	2
<u>6. Λειτουργικότητα (Β Φάση)</u>	2
<u>7. Συμπεράσματα</u>	2

Εισαγωγή

Το project είναι βασισμένο στο μοντέλο MVC Model View ,Controller).

Δηλαδή θα προσπαθήσουμε ο Controller("εγκεφαλός") να συνδέει το Model με το View.Στις επομενες ενότητες θα δούμε αναλυτικότερα αυτά τα πακέτα και την αναπαράστασή τους σε διάγραμμα UML.

Η Σχεδίαση και οι Κλάσεις του Πακέτου Model

Στο πακέτο Model περιέχονται τα πακέτα **Board**(με την abstract class Tile και όλα τα πλακίδια),**Card** (με το interface Card και όλες οι Mail/Deal κάρτες),**Deck**(για την δημιουργία στοίβας από κάρτες),**Player** (αναπαράσταση παίχτη),**Position** (τη θέση των object),**Turn**(για την σειρά των παιχτών).

Package Board

Αρχικά το πακέτο **Board** περιέχει το ζάρι του παιχνιδιού.

Class Dice :

attribute:

1) private int dice_value (για να αποθηκεύτε η προηγούμενη τιμή του ζαριού)

methods:

1) public int roll_dice () (κάνει return μια τυχαία τιμή [1,6])

2) public int get_dice_value() (κάνει return την προηγούμενη τιμή)

Classes που κάνουν extend to dice :

- Tile
- Player
- Mail/Deal card

Abstract Class Tile :

attributes:

1) private int day; (η μερα απο 1-31)

2) private String str_day; (η μερα σε String [Δευτερα,Κυριακη])

3) private String name; (Το ονομα του Tile (πχ Jackpot))

methods:

1)Constructor : public Tile(int day , String str_day,String name)
(Creates a Tile)

2) Setters day,str_day,name(int,String) (set private variables)

- 3) Getters day, str_day, name() (get private variables)
- 4) public void day_action() (οταν το tile εχει σαν μερα Πεμπτη/Σαββατο)
- 5) String toString() (κανει return to info του tile)

Οι classes που κανου extend την TILE υλοποιουν την method void tile_action() η οποια κανει το action του καθε tile οταν "πεφτεις" πανω του .Οι classes αυτες ειναι :

- 1) Tile_Deal (το action τραβαι μια Deal καρτα)
- 2) Tile_Mail (το action τραβαι μια Mail καρτα)
- 3) Tile_Mail2 (το action τραβαι δυο Mail καρτες)
- 4) Yard_Sale (το action ριχνει το ζαρι πληρωνει $100 * \text{dice_value}$ και τραβαι μια Deal_Card)

Και οι τεσσερις περνουν ως παραμετρο (player p, deck p) για το ποιος παιχτης τραβαι και απο πιο deck.

5) Payday (Η payday ειναι παντα η μερα 31 Wednesday, και το action κανει :

- 1: set τα λεφτα του παικτη σε +500
- 2: πληρωνει τους λογαριασμους (εαν δεν εχει περνει δανειο)
- 3: πληρωνει 10% του δανειου του (εαν δεν εχει περνει δανειο)
- 4: μπορει να πληρωσει το δανειο η ποσοστο του δανειου σε πολλαπλασιο του 1000
- 5: αν ειναι τελευταιος μηνας αδειαζουμε το Deck με τις deal_cards του
- 6: αν δεν ειναι ο τελευταιος μηνας αρχιζει απο την αρχη του board

Το περνει ως παραμετρο boolean pay για αν θελει να πληρωσει ποσοστο του δανειου και int pay_value για το ποσο που θελει να πληρωσει (πρεπει να ειναι

πολλαπλασιο του 1000 και να διαθετει το ποσο).

6)Jackpot(To jackpot ειναι παντα η μερα 32 και δεν γινεται να πεσεις πανω του ,αν το ζαρι για move ειναι =6 τοτε γινεται trigger το Jackpot)

attributes : private int jackpot_value;

methods :

1: setJackpot_value(για να κανει set το Jackpot)

2: getJackpot_value(κανει return το Jackpot)

3:το action(προσθετει στα λεφτα του παιχτη το jackpot και κανει το jackpot=0)

7)Sweepstakes(το action ριχνει το ζαρι και προσθετει στα λεφτα του παιχτη 1000 *dice_value)

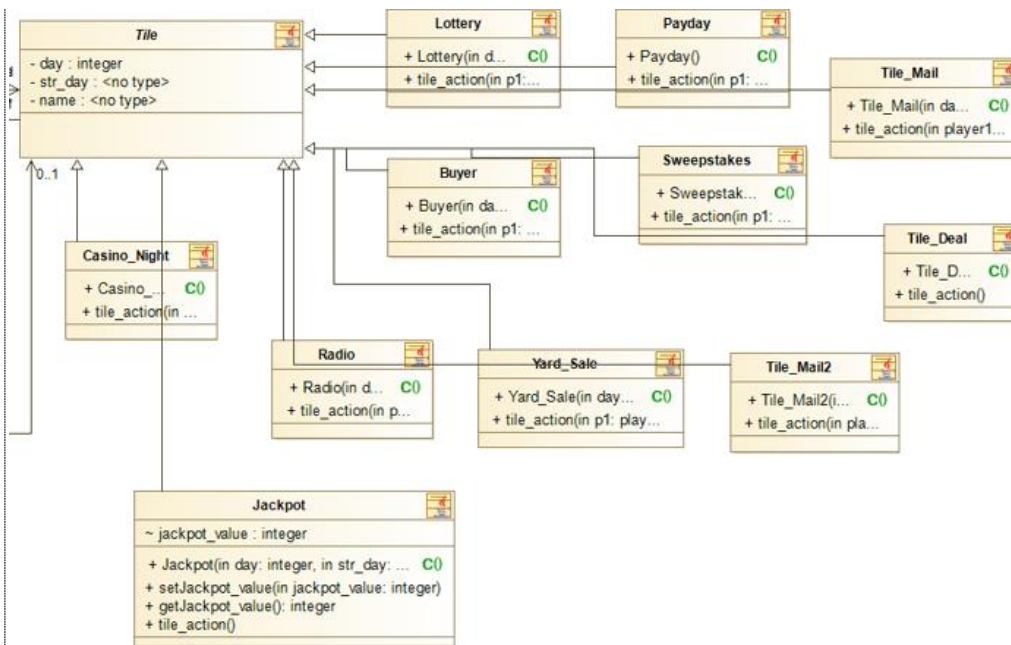
Το action του jackpot και τοθ Sweepstakes περνουν ως παραμετρο (player p) για το σε ποιον παιχτη γινονται τα changes.

8)Buyer (player p,boolean sell) (Το action κανει sell την καρτα αν το sell ειναι true)

8)Casino_night (player p,Dice dice,Jackpot jackpot) (Το action περνει το last_dice_value και αναλογα την τιμη του ο παιχτης δινει 500 money στο jackpot ή περνει 500 απο την τραπεζα)

9)Lottery (player p1,player p2,int p1_value,int p2_value)(Το action δημιουργει ενα ζαρι και ριχνει μεχρι p1_value = dice_value ή p2_value = dice_value ,και δινει 1000 ευρω στον νικητη)

10)Radio (player p1,player p2)(Το action ριχνει τα ζαρια των παιχτων μεχρι καποιο να ειναι μεγαλυτερο επειτα δινει 1000 money στον νικητη)



(Uml Tile)

Το επομενο πακετο ειναι το **Card** το οποιο περιεχει :

Interface Card :

methods : καθε καρτα υλοποιει τις εξης methods

- 1)String getName(κανει return το name της καρτας)
- 2)String getAction_Str(κανει return το action text της καρτας)
- 3)String toDo_str (κανει return το text στο button της καρτας,πχ πληρωσε 50 ...)
- 4)String ToString (κανει return το info της καρτας)

Οι classes που κανουν implement την Card ειναι η **Deal_Card** και η **Abstract Mail_Card**

Deal_Card :

attributes :

int buy_value , sell_value(To value της καρτας)

player Card_Owner (Ο owner της καρτας)

String name,action_Str,ToDo_str,(τα παραπανω απο Card)

methods :

1) Constructor : Deal_Card(int buy_value,int sell_value,String name,String action_str,String toDo_str) (Δημιουργει μια Deal card)

2) set για buy_value,sell_value,card_owner

3) get για ολα τα attributes

4) int get ownerId(κανει return το ID του owner αν εχει αλλιως null)

5)Τα return strings του interface

Abstract Mail_Card extends Dice:

attributes :

String name,action_Str,ToDo_str,(τα παραπανω απο Card)

methods :

1) Constructor : Mail_Card(String name,String action_str,String toDo_str)

2)Τα get strings του interface

Ολες οι classes που κανουν extend την Mail_Card (οπως και με την Tile) εχουν τον constructor τους και υλοποιουν την method void card action() οι classes αυτες ειναι :

1)Get paid from the neighbour

card_action(player p1,player p2)(ο p2 δίνει 50 money στον p1 ,εαν δεν εχει περνει δανειο)

2)Pay the neighbour

card_action(player p1,player p2)(ο p1 δίνει 60 money στον p2 ,εαν δεν εχει περνει δανειο)

3)Move to

card_action(player p1)(player moves to the closest Mail_Tile/Deal_Tile)

4)Advertisement

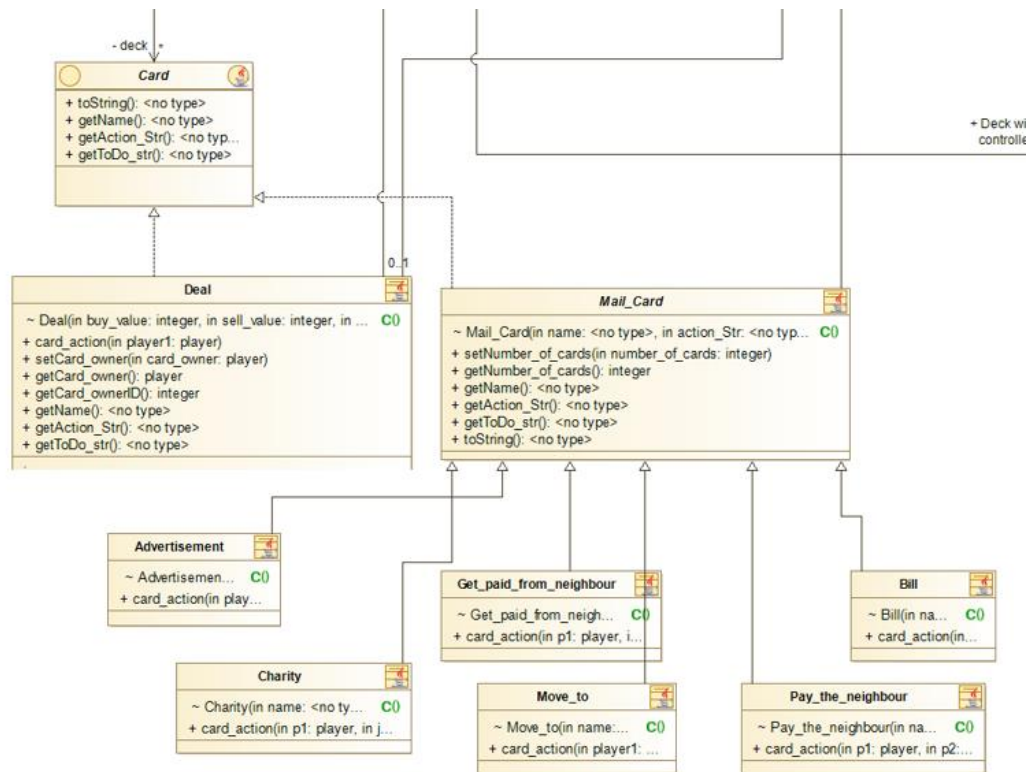
card_action(player p1)(ο p1 περνει 20 money)

5)Bill

card_action(player p1)(το bill του p1 +500 ,πρεπει να το πληρωσει στο τελος του μηνα)

6)Charity

card_action(player p1,Jackpot jackpot)(ο p1 δίνει 50 στο jackpot,εαν δεν εχει περνει δανειο)



(Card Uml)

Το πακέτο **Deck** περιέχει την class Deck η οποία βοηθάει να δημιουργούμε ένα ArrayList από Cards

Deck :

attributes :

- 1) private player owner; (Εάν έχει owner)
- 2) private ArrayList<Card> deck ; (Ενα list με καρτες)

methods :

- 1) Constructor Deck() (φτιάχνει ένα deck)
- 2) getDeck/owner() (κάνει return το deck/owner(εάν έχει))

- 3) set Deck(Arraylist<Card> deck) (κάνει set το deck)
- 4) set owner(player p1) (κάνει set τον owner)
- 5) void init_Mail_Deck() (αυτο deck είναι πλέον ένα starter Mail_Deck)
- 6) void init_Deal_Deck() (αυτο deck είναι πλέον ένα starter Deal_Deck)
- 7) boolean isEmpty() (return true αν το deck είναι empty)
- 8) void add_card()(κάνει add μια καρτα)
- 9) void remove_card(Card c)(κάνει remove μια καρτα)
- 10) void shuffle()(ανακατεύει το deck)
- 11) int size() (κάνει return το size του deck)
- 12) Card getCard(int position)(κάνει return μια καρτα που βρίσκεται στο position της λίστας)

Το πακέτο **Player** περιέχει την class player μια από τις βασικές class του προγράμματος η οποία δημιουργεί την οντότητα player.

Player extends Dice :

attributes :

- 1)private String name; (Το name του player)
- 2)private int Money,Bills,Loan(Τα λεφτά του player)
- 3)private final int id(το ID του player)
- 4)private Deck deal_cards(όλες οι Deal cards του player)
- 5)private position curr_position(το position του player)
- 6)private int position (βοηθάει στο move του player)

7)private boolean has_finished(finished turn)

methods :

1)Constructor : player(String name,int id) (Μεσα στον constructor αρχικοποιουμε ολες τις τιμες του player και το money με 3500)

2) setMoney/Bills/Loan/deal_cards/position/curr_position/name
(Κανει set τα variables)

3) get Money/Bills/Loan/deal_cards/position/curr_position/name
(Κανει return τα variables)

4)το set/get του has_finsihed (ειναι για οταν εχει τελιωσει ο γυρος του)

5)int get_positionNumber(κανει return την ημερα του μηνα που βρισκετε)

6)String toString (κανει return το info του player)

Το πακετο **Position** περιεχει την class position.Ο σκοπος της ειναι να περιεχει μεσα της το board και να κραταει track το position του player

Position :

attributes :

1)private int curr_position; (το position σε μερες του μηνα)

2)private Tile curr_Tile; (Το tile που βρισκετε αυτη την στιγμη)

3)private Tile[] board=new board[32] (το board του παιχιδιου)

methods :

1)Constructor : position()(αρχικοποιει το curr_position=0)

2) get curr_position/curr_Tile/board(κανει return τα variables)

3) set curr_position/curr_Tile/board(κανει set τα variables)

4) void move_tile(int next_position) (κανει move τοσα Tiles όσο και το next_position)

5) toString() (κανει return το string του curr_tile)

Τελος το πακετο **Turn** περιεχει την class turn η οποια χρησιμοποιειται για να κραται track της σειρας των παιχτων του παιχνιδιου.

Turn :

attributes :

1)private int curr_ID (Ο player που παιζει τωρα)

2)private int num,last playerID(num ειναι ποσοι ειναι οι παιχτες και last player ο τελευταιος που επαιζε)

methods :

1)Constructor : turn() (αρχικοποιει το num=2)

2)set curr_ID/last playerID/num (κανει set τα variables)

3)get curr_ID/last playerID/num (κανει return τα variables)

4) boolean isFinished(player p1) (ειναι true εαν ο p1 εχει τελιωσει τον γυρο του)

Τελος μας μενει το **View** και ο **Controller**

View :

Οπως ειπαμε και στην αρχη το View εχει ευθυνη για το γραφικο κομματι.Μεσα στο View θα δημιουργουμε ενα frame το οποιο θα περιεχει το κεντρικο Panel και αλλα δυο (ενα για καθε player),στο κεντρικο Panel θα τοποθετουμε ολα τα πλακιδια και στο Panel του καθε player θα εχει buttons οπως GetLoan για να παρει δανειο , My deal Cards για να δει τις καρτες που εχει στην κατοχη του , Roll Dice για να ριξει το ζαρι και End Turn για να τελιωσει ο γυρος του. Επισης θα εχει πληροφοριες οπως το ονομα του,τα λεφτα που εχει , και τους λογαριασμους που χρωσταει.Επισης θα υπαρχει και ενα τεταρτο Panel (Info Box) που θα ενημερωνει τους παιχτες για τις ενεργειες που πρεπει να κανουν ,ποιος παιζει αυτη την στιγμη και ποσοι μηνες μενουν πριν τελιωσει το παιχνιδι.Τελος θα υπαρχει ενα label στο κεντρικο Panel κατω απο το Jackpot που θα ενημερωνει για την τιμη του jackpot αυτη την στιγμη.

Controller :

Ο controller θα ειναι υπευθυνος για την ενωση του View-Model.Θα δημιουργει ενα παιχνιδι και θα αλλαζει τις μεταβλητες του Model οταν ο παιχτης παταει buttons στο View.Τελος ειναι υπευθυνος για την ενημερωση των ενεργειων και ποτε τελωνει το παιχνιδι.

attributes :

1) private int allMonths,curr_month (ποσους μηνες θα παιζουν,και σε ποιον ειναι τωρα)

2) private player P1,P2 (Οι δυο παιχτες)

3) private ArrayList<player>players (ενα arraylist με τους δυο παιχτες για την σειρα)

4)private Deck Mail_deck,Deal_deck (τα δυο κεντρικα decks που τραβανε οι παιχτες)

5)private Deck P1_deck,P2_deck(τα deck που περιεχουν τις deal cards των παιχτων)

6)private Dice dice1,dice2 (τα ζαρια των παιχτων)

7)Jackpot jackpot (για να κραταει την τιμη του Jackpot)

8)private Turn turn (για την σειρα)

9)private boolean is finished (true αν το παιχνιδι εχει τελιωσει)

10)private boolean p1_turn,p2_turn(true αν εχουν τελιωσει την σειρα τους)

11)private boolean p1_decision,p2_decision(για τις αποφασεις που περνουν μεσα στο παιχνιδι)

methods:

1)Constructor : controller() (αρχικοποιει τιμες και κανει choose random ποιος παιζει πρωτος)

2)private boolean first ()(return true αν p1 παιζει πρωτος)

3)void roll DiceAndMove () (ριχνει το ζαρι και κανει move αναλογα)

4)void DrawCard () (τραβαει μια καρτα)

5)void MyDealCard() (κανει return τις καρτες του player)

6)void set allMonths() (set ποσους μηνες θα παιξουν)

7)void GetLoan () (παιρνει δανειο (button))

8)void perform_day_action()(κανει κατι αν πεσει σε μερα Πεμπτη/Σαββατο)

9)void perform_tile_action()(κανει κατι αναλογα το tile που βρισκετε)

10)void Jackpot() (δινει το Jackpot αν ο παιχτης φερει 6)

Τελος.

