

**ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΟΣ**

**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ, ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΓΕΝΝΗΤΡΙΑ ΕΠΙΤΡΑΠΕΖΙΩΝ ΠΑΙΧΝΙΔΙΩΝ**

**Τέταρτη Εβδομάδα**

**Ομαδική Εργασία:**

Αναστασιάδης Αλκίνοος (20003)

Ζήνα Ελένη (20046)

Λαγιόκαπας Δημήτριος (20079)

Μακρή Στυλιανή (20060)

Επιβλέποντες:

Κουρέας Αργύριος

Λάντζος Θεόδωρος

**ΣΕΡΡΕΣ****, 1 έως 5 ΜΑΙΟΥ 2023**

Περιεχόμενα

[Περίληψη 3](#_Toc134124219)

[Εισαγωγή 3](#_Toc134124220)

[1. Μεθοδολογία 4](#_Toc134124221)

[2. Υλοποίηση 5](#_Toc134124222)

[3. Χρονοδιάγραμμα 13](#_Toc134124223)

[4.Αποτελέσματα-Επίλογος 14](#_Toc134124224)

[5.Βιβλιογραφία 14](#_Toc134124225)

# Περίληψη

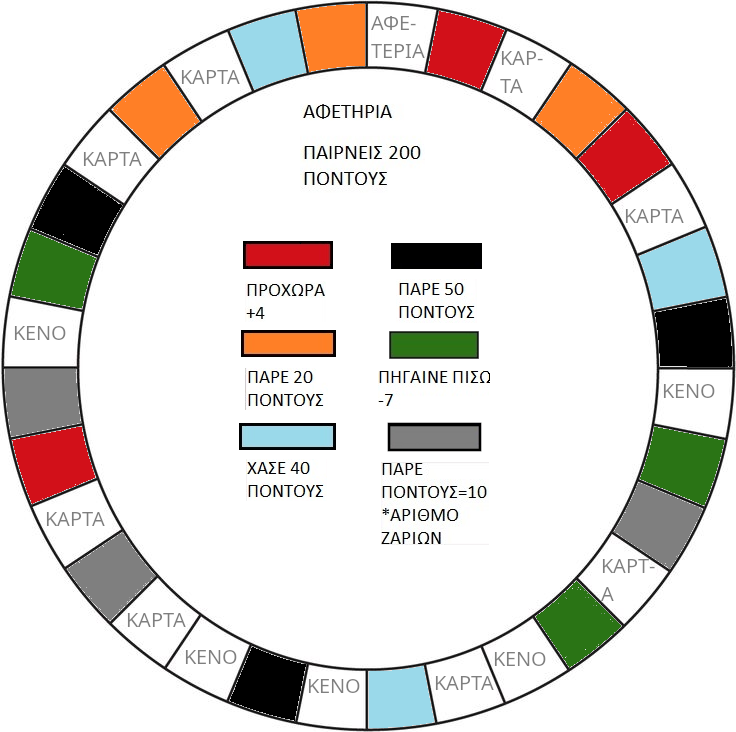
Αυτή την εβδομάδα κάναμε πολλές αλλαγές στο πρόγραμμα μας, τόσο στον κώδικα για το απλό παιχνίδι, όσο και για προχωρημένο. Μελετήσαμε καλύτερα και κάναμε χρήση των services σχεδόν σε κάθε πτυχή του προγράμματός μας. Έχουμε ξεχωριστές κλάσεις για ότι αφορά το απλό και το προχωρημένο παιχνίδι και αυτό μας βοηθάει στο να μπορούμε να κάνουμε συγκεκριμένες αλλαγές σε ότι χρειάζεται. Έγιναν κάποιες αλλαγές στη δομή του ταμπλό μας και το αρχείο json πήρε την τελική του μορφή.

# Εισαγωγή

Στα πλαίσια του προγραμματισμού, τα services είναι ένα εργαλείο που παρέχει μια συγκεκριμένη business λογική ή λειτουργικότητα που μπορεί να χρησιμοποιηθεί από άλλες ενότητες. Τα services εφαρμόζονται συχνά σαν κλάσεις ή interfaces τα οποία προσφέρουν μεθόδους, συναρτήσεις και ιδιότητες για να χρησιμοποιηθούν σε άλλα σημεία της εφαρμογής. Τα services είναι ανεξάρτητα και επαναχρησιμοποιούμενα, γεγονός που βοηθάει τους προγραμματιστές να τα συγχωνεύσουν σε διάφορα μέρη μιας εφαρμογής χωρίς να αντιγράφουν κώδικα. Στο δικό μας πρόγραμμα τα services μας βοήθησαν στο να γίνει το πρόγραμμα μας πιο ξεκάθαρο και πιο αποδοτικό.

# Μεθοδολογία

Αρχικά, διαμορφώσαμε το ταμπλό:



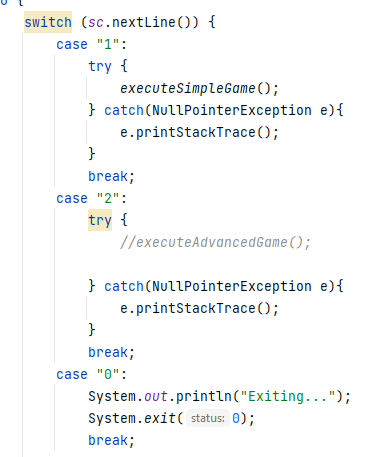
Κόκκινο tile: Προχώρα +4  
Πορτοκαλί tile: Πάρε 20 πόντους  
Γαλάζιο tile: Χάσε 40 πόντους   
Μαύρο tile: Πάρε 50 πόντους  
Πράσινο tile: Πήγαινε πίσω -7  
Γκρι tile: Πάρε πόντους = 10 φορές τη ζαριά σου

Βλέπουμε πως έχουν αφαιρεθεί τα tiles «Φυλακή», «Τζάκποτ» και «Προστασία» και έχουν γίνει κενά tiles για λόγους ευκολίας. Συνεπώς, αφαιρέθηκαν και οι κανόνες και οι κάρτες που αφορούν τα συγκεκριμένα τετράγωνα.

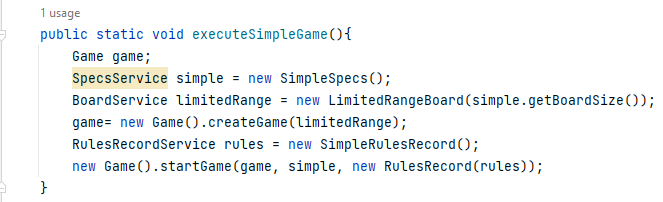
Στο πρόγραμμα μας πλέον υπάρχει μια ξεκάθαρη μορφή, με κάθε παράμετρο που υπάρχει στο παιχνίδι (κάρτες, παίκτες, κανόνες, ταμπλό κλπ) να έχει χωριστεί στην απλή και την προχωρημένη του έκδοση. Για παράδειγμα, υπάρχει η simpleTile και η AdvancedTile, η SimplePlayer και η AdvancedPlayer κλπ). Αυτό μας βοηθάει στο να έχουμε ξεκάθαρη εικόνα των διεργασιών, να χωρίζουμε εύκολα εργασίες και να μπορούμε να κάνουμε αλλαγές στη λογική και αρχιτεκτονική στο ένα παιχνίδι χωρίς να επηρεάζεται κάτι στο άλλο.

# Υλοποίηση

Αρχικά, στην κλάση main έχουμε την εξής αλλαγή στην chooseGame(), εφόσον πρέπει πλέον να λάβουμε υπόψη και την επιλογή του παίκτη για το Advanced παιχνίδι:

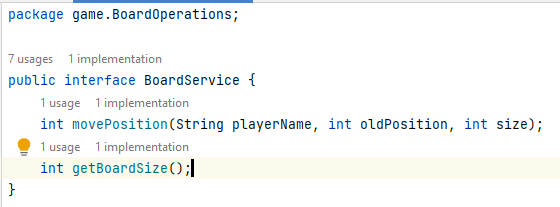


Εάν ο παίκτης επιλέξει 1, εκτελείται η executeSimpleGame, ενώ αντίστοιχα αν επιλέξει 2 εκτελείται η executeAdvancedGame (δεν απεικονίζεται, είναι παρόμοια με την executeSimpleGame)

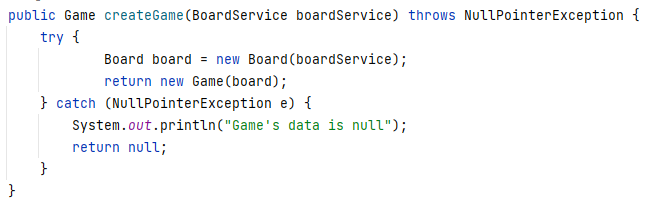


Η executeSimpleGame (και αντίστοιχα η executeAdvancedGame με τις δικές της παραμέτρους) δημιουργεί το απλό παιχνίδι, αρχικοποιεί το ταμπλό και ξεκινά το παιχνίδι. Δημιουργεί αντικείμενο τύπου “Game”, αρχικοποιεί τα specifications του παιχνιδιού με χρήση της ‘SimpleSpecs’ κλάσης και δημιουργεί ένα αρχείο κανόνων με χρήση της ‘SimpleRulesRecord’. Τέλος, ξεκινάει το παιχνίδι με την startGame().

Η κλάση Board πλέον χρησιμοποιεί την BoardService, η οποία ορίζει τις μεθόδους τις οποίες πρέπει μια κλάση να περιέχει για να παρέχει τη λειτουργικότητα του game board. Συγκεκριμένα περιέχει την movePosition, και την getBoardSize:



Οι αλλαγές στην startGame της Game, η οποία πλέον παίρνει specsService και RulesRecord αντικείμενα ως παραμέτρους, εκτυπώνει τους κανόνες και προσθέτει τους παίκτες στο παιχνίδι με χρήση της SimplePlayer [[13]](#_5.Βιβλιογραφία) κλάσης:



Υπάρχουν λοιπόν και οι SimplePlayer (για τον απλό τύπο παιχνιδιού) και AdvancedPlayer (για τον προχωρημένο τύπο παιχνιδιού) κλάσεις. Στην SimplePlayer έχουμε τα εξής πεδία:

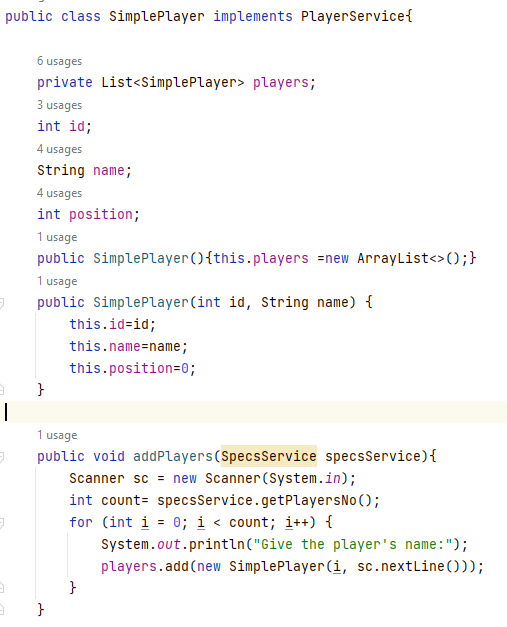
‘players’ : μία λίστα από SimplePlayer αντικείμενα που αντιπροσωπεύουν τους παίκτες του παιχνιδιού.

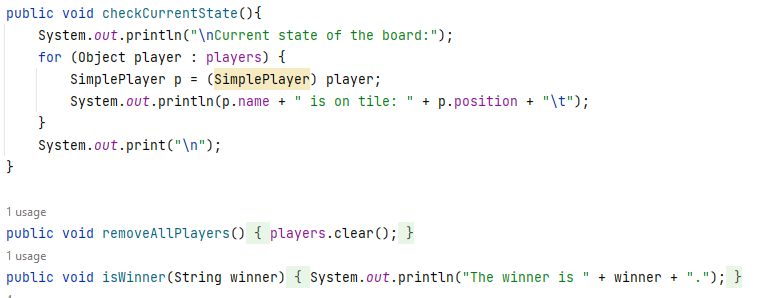
‘id’ : έναν ακέραιο που αντιπροσωπεύει την ταυτότητα του παίκτη.

‘name’ : ένα string που αντιπροσωπεύει το όνομα του παίκτη

‘position’ : έναν ακέραιο που αντιπροσωπεύει την θέση του παίκτη στο ταμπλό.

Η κλάση περιέχει τις εξής συναρτήσεις : addPlayers,checkCurrentState, removeAllPlayers, isWinner, καθώς και getters και setters για τα strings. Οι συναρτήσεις υπήρχαν ήδη στο πρόγραμμα μας και είναι γνώστες.





Στη συνέχεια, δημιουργήσαμε το service RulesRecordService που προς το παρόν περιέχει μια μέθοδο, την printRules. Όποια κλάση που εφαρμόζει αυτό το service πρέπει να έχει μια printRules μέθοδο.



Συγκεκριμένα, οι δύο νέες κλάσεις SimpleRulesRecord και AdvancedRulesRecord χρησιμοποιούν το service. Η SimpleRulesRecord έχει έναν constructor που δημιουργεί μια αναφορά του SimpleSpecs, που εφαρμόζει την SpecsService και παίρνει όλους τους κανόνες από εκεί. Έπειτα η printRules εκτυπώνει τους κανόνες. Αντίστοιχα δουλεύει και η AdvancedRulesRecord.

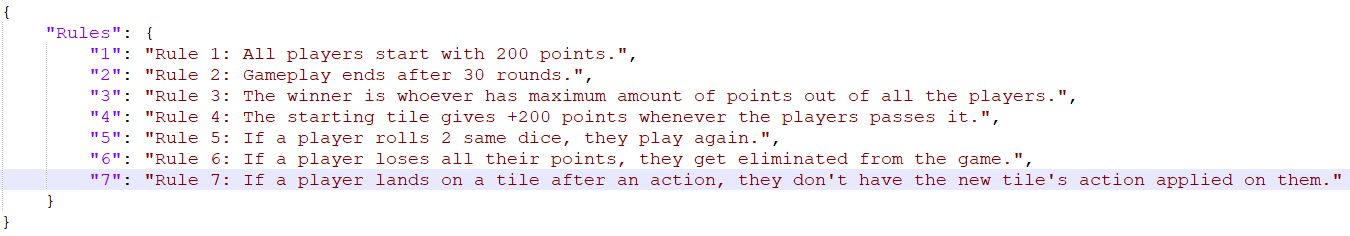


Η RulesRecord κλάση μπορεί εύκολα να τροποποιηθεί για να γίνει χρήση διαφόρων εφαρμογών της printRules μεθόδου, χωρίς να χρειάζεται να αλλάξουμε την υπόλοιπη κλάση. Μας επιτρέπει να τεστάρουμε τη κλάση εύκολα, για να βλέπουμε πως χρησιμοποιείται το interface του RulesRecordService.

To πλήρως διαμορφωμένο αρχείο json έχει την εξής μορφή (δεν περιέχεται ολόκληρο λόγω του μεγέθους του):







Δομή του json:

Game : περιέχει πληροφορίες για τις ρυθμίσεις του παιχνιδιού, τον αριθμό των παικτών και το μέγεθος και τον τύπο του ταμπλό.  
Tiles : περιέχει πληροφορίες για τα tiles του παιχνιδιού. Το index είναι η θέση του tile στο ταμπλό, το stat είναι το είδος του tile, parameter είναι η παράμετρος που επηρεάζει το κάθε tile, και value είναι το πόσο μεταβάλλεται η παράμετρος.  
Cards : Περιέχει πληροφορίες για τις κάρτες του παιχνιδιού, με κάθε κάρτα να εκπροσωπείται με το id, ένα μοναδικό identifier για κάθε κάρτα, το message που περιέχει το μήνυμα που εμφανίζει η κάρτα, και μετά το parameter και το value που δρούνε με παρόμοιο τρόπο όπως τα Tiles.  
Rules: Είναι διαμορφωμένο όπως και το json για το απλό παιχνίδι.

Τέλος, βλέπουμε την TilesService , η οποία περιέχει την getTiles που επιστρέφει μια λίστα από αντικείμενα τύπου «Τ». Ο γενικός τύπος Τ[14]επιτρέπει στο interface να χρησιμοποιείται με διάφορα είδη tiles, όπως strings, integers ή αντικείμενα.



# 3. Χρονοδιάγραμμα

29-30/4/2023:

Ελένη: Ολοκλήρωση σύνταξης του περιεχομένου του νέου json αρχείου.

Διορθώσεις με τη χρήση του BoardService στην createGame() και προσθήκη SpecsService για να διαβάζονται τα κατάλληλα δεδομένα. Δημιουργία νέων συναρτήσεων στην Main για να δημιουργείται το κατάλληλο παιχνίδι με βάση την επιλογή του παίκτη. Υλοποίηση της κλάσης AdvancedSpecs για να διαβάζονται τα δεδομένα από το νέο json αρχείο (90% ολοκληρωμένη). Μικρές διορθώσεις που συζητήθηκαν στην θεωρία. Προετοιμασία κώδικα, διαμόρφωση κλάσεων για τα services. Έδωσε εργασίες στην υπόλοιπη ομάδα.

30/4/2023:

Αλκίνοος: Τροποποίηση της Tile ώστε να δουλεύει με την χρήση της SpecsService. Επανασχεδιασμός του ταμπλό κρατώντας τις βασικές λειτουργίες.

Δημιουργία της εικόνας του καινούργιου ταμπλό.

2/5:

Στέλλα: Δημιουργία αντικειμένου τύπου "RulesRecordService" στις συναρτήσεις execute της main για να τυπώνονται οι κατάλληλοι κανόνες ανάλογα τον τύπο του παιχνιδιού που έχει επιλέξει ο χρήστης.

Δημήτρης: αλλαγή της κλάσης Player για να χρησιμοποιεί την SpecsService.

Αλκίνοος : Δημιουργία νέας κλάσης, της AdvancedTile η οποία είναι η κλάση των Tile για το Advanced παιχνίδι τύπου Μονόπολυ. Χρήση Generic στις κλάσεις SpecsService, SimpleSpecs, AdvancedSpecs, TileService, SimpleTile και AdvancedTile ώστε να μπορεί να διαχειρίζεται το πρόγραμμα μέσω των Services. Αλλαγές στον τύπο δεδομένων (Integer στην SimpleSpecs και SimpleTile για το Simple Game και Object στις AdvancedSpecs και AdvancedTile για το Advanced Game). Επίσης έγινε τροποποίηση του άμεσα επηρεαζόμενου κώδικα ώστε να λειτουργεί το πρόγραμμα.

3/5/2023:

Ελένη: Συγκέντρωση εργασιών από την ομάδα και υλοποίηση τους στο πρόγραμμα.

Αλλαγές με την εμφάνιση των περιεχομένων του Project (ένα package για κάθε οντότητα που τη χαρακτηρίζει). Διορθώσεις για να γίνει πιο απλή η μορφή των συναρτήσεων και για την σωστή χρήση των νέων services (TileService, CardService).

Δημιουργία κλάσεων για κάθε Tile για το νέο παιχνίδι και δημιουργία άλλο ένα service για αυτά. Ολοκλήρωση του διαβάσματος του AdvancedSpecs για το νέο παιχνίδι.

Δημιουργία κλάσεων Player, AdvancedPlayer και μετονομασία της παλιάς Player σε SimplePlayer και service. Άλλαξε η λίστα των παικτών σε αντικείμενα για να κρατάνε τις ιδιότητες τους και έτσι άλλαξαν και οι συναρτήσεις σχετικά με το gameplay, την κίνηση, την νέα θέση κτλ

Στέλλα: Σύνταξη αρχικής μορφής του report.

4/5:

Στέλλα: Ολοκλήρωση του report.

Ομάδα: Ολοκλήρωση του project και μικρές αλλαγές στο json και στον κώδικα.

# 4.Αποτελέσματα-Επίλογος

Επιλογικά, αυτήν την εβδομάδα έγιναν μεγάλες αλλαγές στο πρόγραμμα μας, ακολουθώντας τη λογική των services. Είμαστε στην τελική ευθεία για να ολοκληρώσουμε το πρόγραμμα μας. Υπάρχουν πράγματα που πρέπει να προσθέσουμε για να είναι πλήρως εκτελέσιμος ο κώδικας, και σίγουρα υπάρχουν δυνατότητες για βελτιστοποίηση και για νέα features, όπως αποθήκευση παιχνιδιού, UI κλπ.

# 5.Βιβλιογραφία

|  |  |
| --- | --- |
| [14] | Objects and Classes in Java  <https://www.javatpoint.com/object-and-class-in-java> |
| [15] | The Java Tutorial - Generic Types  <https://docs.oracle.com/javase/tutorial/java/generics/types.html> |