

**ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΟΣ**

**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ, ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΓΕΝΝΗΤΡΙΑ ΕΠΙΤΡΑΠΕΖΙΩΝ ΠΑΙΧΝΙΔΙΩΝ**

**Τρίτη Εβδομάδα**

**Ομαδική Εργασία:**

Αναστασιάδης Αλκίνοος (20003)

Ζήνα Ελένη (20046)

Λαγιόκαπας Δημήτριος (20079)

Μακρή Στυλιανή (20060)

Επιβλέποντες:

Κουρέας Αργύριος

Λάντζος Θεόδωρος

**ΣΕΡΡΕΣ****, 24 έως 28 ΑΠΡΙΛΙΟΥ 2023**

Περιεχόμενα

[Περίληψη 3](#_Toc133512155)

[Εισαγωγή 4](#_Toc133512156)

[1. Μεθοδολογία 5](#_Toc133512157)

[2. Υλοποίηση 8](#_Toc133512158)

[3. Χρονοδιάγραμμα 16](#_Toc133512159)

[4. Αποτελέσματα-Επίλογος 17](#_Toc133512160)

[5. Βιβλιογραφία 17](#_Toc133512161)

# Περίληψη

Αυτήν την εβδομάδα ασχοληθήκαμε με τον σχεδιασμό του δεύτερου τύπου επιτραπέζιου παιχνιδιού, και στην επέκταση του κώδικα έτσι ώστε να μπορέσουμε να το προσθέσουμε στην γεννήτρια μας. Έγιναν κάποιες διορθώσεις στον κώδικα για να υποδεχτεί το πρόγραμμά μας τα services πιο εύκολα. Επίσης, μελετήσαμε την λειτουργία των services μέσα από παραδείγματα και προσπαθήσαμε να εφαρμόσουμε αυτή τη λογική στο πρόγραμμα μας, για παράδειγμα βρίσκοντας τρόπο να χρησιμοποιήσουμε τα services συνδέοντας και άλλα παιχνίδια μέσα σε ένα πρόγραμμα. Έγινε επίσης η σχεδίαση του ταμπλό του δεύτερου τύπου παιχνιδιού και ο ορισμός των κανόνων και των ενεργειών που θα έχουν τα διάφορα tiles και οι κάρτες.

# Εισαγωγή

Ένα παιχνίδι σαν την «Μονόπολυ» είναι σαφώς πιο δύσκολο στον σχεδιασμό και στην εκτέλεση με κώδικα σε σχέση με το παιχνίδι που έχουμε δημιουργήσει μέχρι στιγμής. Το κύριο χαρακτηριστικό του δεύτερου τύπου παιχνιδιού που ξεκινήσαμε είναι πως δεν έχει συγκεκριμένη αρχή και τέλος, αλλά τα τετράγωνα του ταμπλό σχηματίζουν έναν βρόχο. Γι’ αυτόν τον λόγο ο νικητής επιλέγεται με βάση άλλα κριτήρια, όχι με το ποιος θα φτάσει πρώτος στο «τέρμα». Οπότε είναι απαραίτητο να προστεθούν επιπλέον μεταβλητές στην λογική του προγράμματος, όπως για παράδειγμα πόντοι, ζωές, δυνάμεις κλπ. Σαν ομάδα επιλέξαμε το σύστημα των πόντων, οι οποίοι θα προσθέτονται ή θα αφαιρούνται από κάθε παίκτη κατά τη διάρκεια του παιχνιδιού είτε από τις ιδιότητες των tiles, είτε μέσα από τις κάρτες.

Υπάρχει ένα είδος στρατηγικής στο παιχνίδι, αλλά η πορεία του παιχνιδιού είναι κυρίως τυχαία, εξαρτάται δηλαδή από τις ζαριές, το τι κάρτα θα τύχει σε κάθε παίκτη κλπ.

# Μεθοδολογία

Ξεκινήσαμε με ένα σχέδιο το οποίο βοηθάει στο να οπτικοποιήσουμε το ταμπλό και να βλέπουμε ξεκάθαρα το τι κάνει κάθε τετράγωνο.



Στο ταμπλό μας έχουμε 32 τετραγωνάκια, κάποια με συγκεκριμένες ιδιότητες και κάποια κενά. Έχουμε 6 χρωματικές ομάδες από 3 τετράγωνα, τα οποία είναι μοιρασμένα στο ταμπλό αλλά έχουν την ίδια ιδιότητα, για παράδειγμα όταν κάποιος παίκτης πέσει πάνω σε οποιοδήποτε πορτοκαλί τετράγωνο, παίρνει 20 πόντους.

Σημαντικό είναι να ορίσουμε τους κανόνες του παιχνιδιού μας:

1. Το παιχνίδι τελειώνει μετά από 30 γύρους ( δηλαδή όταν παίξουν όλοι οι παίκτες 30 φορές).
2. Νικητής είναι αυτός που έχει συγκεντρώσει τους περισσότερους πόντους μετά από τους 30 γύρους.
3. Όλοι οι παίκτες ξεκινάνε με 200 πόντους.
4. Κάθε φορά που κάποιος παίκτης περνάει από την αφετηρία, του προσθέτονται 200 πόντοι.
5. Αν κάποιος παίκτης φέρει διπλή ζαριά (2 ίδια ζάρια), ξαναπαίζει.
6. Όταν κάποιος παίκτης είναι φυλακή μπορεί να βγει είτε με διπλό ζάρι (2 ίδια ζάρια), είτε θυσιάζοντας 250 πόντους. Ο παίκτης στη φυλακή έχει δύο προσπάθειες για να βγει με τη χρήση του ζαριού, αν αποτύχει πρέπει υποχρεωτικά να θυσιάσει τους πόντους. Αν δεν έχει τους πόντους χάνει και βγαίνει από το παιχνίδι.
7. Αν κάποιος παίκτης φτάσει στους 0 πόντους μετά από κάποια ενέργεια, χάνει και βγαίνει από το παιχνίδι.
8. Αν κάποιος παίκτης πάει σε κάποιο τετράγωνο προωθημένος από κάποια κάρτα ή από άλλο τετράγωνο, η ιδιότητα του τετραγώνου στο οποίο πήγε ΔΕΝ εκτελείται. Για παράδειγμα, αν ο παίκτης ρίξει τα ζάρια, πέσει σε ένα κόκκινο τετράγωνο (που τον προωθεί 4 τετράγωνα) όταν προχωρήσει 4 θέσεις και φτάσει στο μαύρο τετράγωνο με την ιδιότητα «Πάρε 50 πόντους», οι πόντοι του ΔΕΝ θα μετρήσουν, απλά ο παίκτης θα μείνει εκεί μέχρι να έρθει ξανά η σειρά του.

Έπειτα, πρέπει να ορίσουμε τις ιδιότητες των καρτών:

* Πάρε 100 πόντους από τυχαίο παίκτη.
* Πήγαινε στην φυλακή.
* Πήγαινε στην αφετηρία.
* Χάσε 150 πόντους.
* Ανταλλαγή πόντων με τυχαίο παίκτη.
* Άλλαξε θέση με τον παίκτη με τους λιγότερους πόντους.
* Άλλαξε θέση με τον παίκτης με τους περισσότερους πόντους.
* Διπλασίασε τους πόντους σου.
* Υποδιπλασίασε τους πόντους σου.

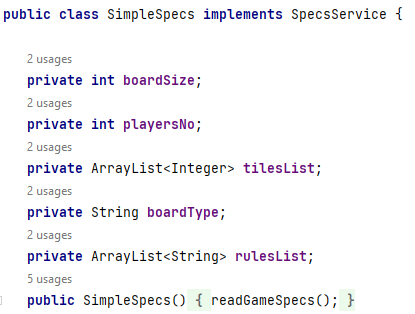
Μία από αυτές τις κάρτες θα επιλέγεται τυχαία κάθε φορά που κάποιος παίκτης πατήσει σε τετράγωνο «Τυχαία κάρτα».

Τα τέσσερα ιδιαίτερα τετράγωνα στις γωνίες του ταμπλό είναι

* Αφετηρία: εκκίνηση των παικτών και προσθήκη 200 πόντους σε όποιον παίκτη περάσει από εκεί.
* Προστασία: ο παίκτης απαλλάσσεται από κάθε αρνητική ενέργεια των άλλων παικτών (για τις κάρτες 5 και 7, αν πληροί τις προϋποθέσεις αυτών).
* Φυλακή: εκεί παραμένουν οι παίκτες μέχρι να φέρουν διπλή ζαριά ή να θυσιάσουν 250 πόντους (κανόνας 6).
* Τζάκποτ: ο παίκτης που θα πέσει πάνω σε αυτό το τετράγωνο επιλέγει μια από τις κάρτες του παιχνιδιού και η ιδιότητα της εκτελείται.

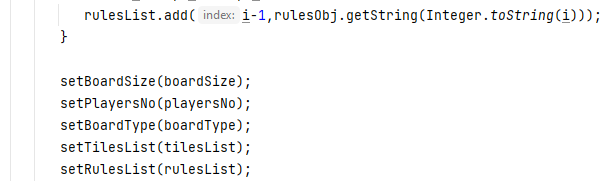
# 2. Υλοποίηση

Σαν πρώτος στόχος, έγινε η διαγραφή των import συναρτήσεων που είχαμε αναφέρει για να μας βοηθάνε με την εισαγωγή των δεδομένων από το json αρχείο. Όμως, αυτό δεν ήταν αισθητικά και προγραμματιστικά σωστό κι έτσι αλλάξαμε απλά τους constructors:

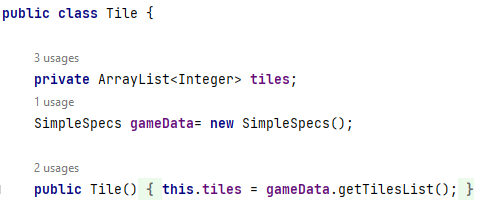


Η παλιά Specifications. Αλλάξαμε το όνομα της για να βάλουμε και άλλη παρόμοια κλάση για άλλο παιχνίδι. Όταν δημιουργούμε ένα αντικείμενο τύπου SimpleSpecs χωρίς ορίσματα σε άλλη κλάση, θα καλείται αυτός ο constructor για να χρησιμοποιήσουμε όποιο από τα δεδομένα θέλουμε από το αρχείο json.

Σε αυτό το σημείο αξίζει να σημειώσουμε ότι πλέον η readGameSpecs() δεν είναι static, για να κάνουμε τον κώδικα επεκτάσιμο με τα services, ώστε να διαβάζεται το κατάλληλο παιχνίδι με βάση τα στοιχεία που δίνουμε. Επίσης, είναι void και οι παράμετροι παίρνουν τιμή με setters εντός της readGameSpecs(), δηλαδή στο τέλος του διαβάσματος του αρχείου.



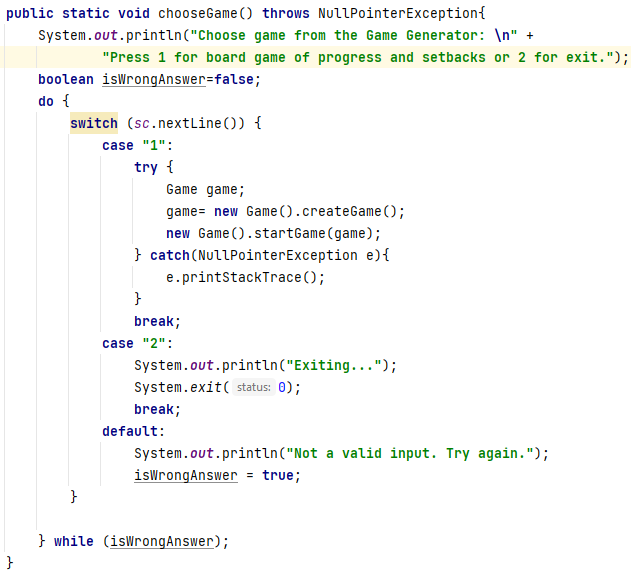
Εδώ, ο constructor της Tile() (παλιά Tiles) λαμβάνει την λίστα των tiles από έναν getter της SimpleSpecs(). Όπως ορίσαμε setters για τον παραπάνω λόγο, έτσι και για να έχουμε πρόσβαση σε αυτά τα δεδομένα από άλλες κλάσεις, ορίσαμε getters. Με τον ίδιο τρόπο έχουμε ορίσει τους constructors για την κλάση RulesRecord (παλιά Rules).



Στην κλάση Player, έχει επεξεργαστεί λίγο η addPlayers() για να δίνει ο παίκτης το όνομα του, οπότε δε χρειάζεται πλέον το String playersName. Επίσης, έχει γίνει και μια μικρή αλλαγή στον constructor:

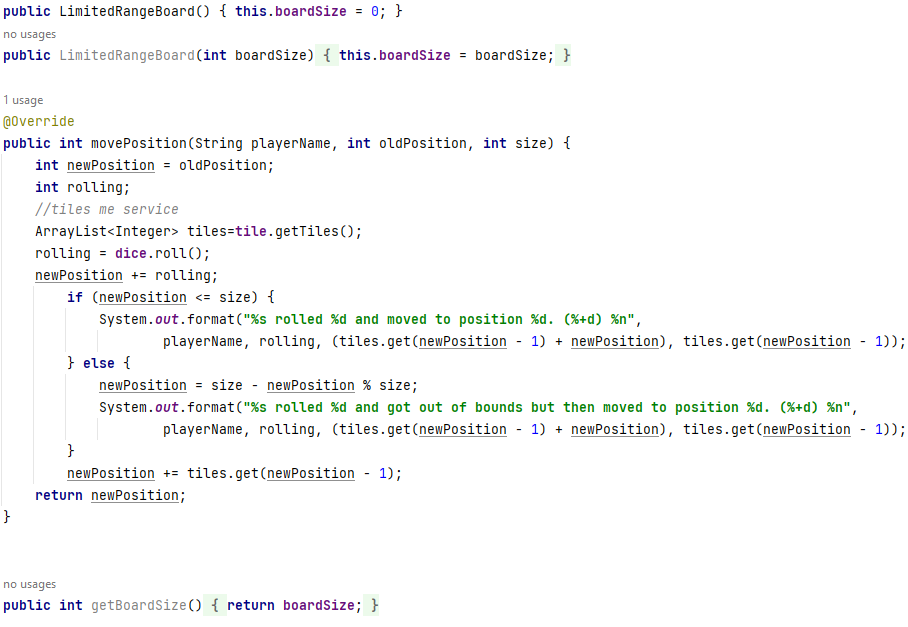


Στην κλάση Main (παλιά GenerateGame), άλλαξαν πολλά εφόσον δεν υπάρχει τίποτα static πλέον. Ουσιαστικά, άλλαξε η συνάρτηση chooseGame(), η οποία απλώς δημιουργεί το παιχνίδι με το κατάλληλο board στην createGame() και με βάση αυτό το παιχνίδι που δημιουργήθηκε, το στέλνουμε στην startGame() σαν όρισμα για να ξεκινήσει το παιχνίδι.



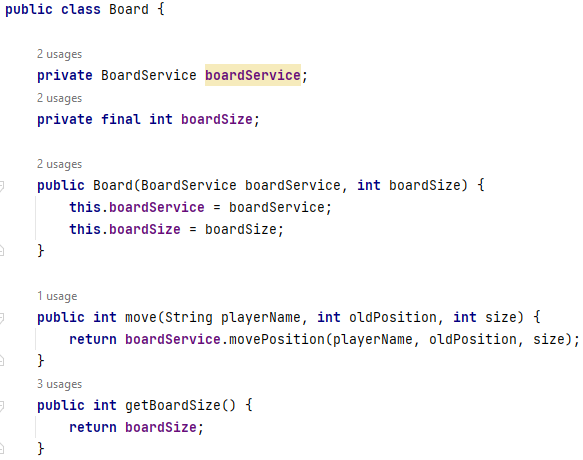
Η κλάση Board πλέον δεν είναι αυτή που είχε όλα τα δεδομένα του ταμπλό, αλλά τα «δανείζεται» για να κατευθύνει το πρόγραμμα στο σωστό ταμπλό για να γίνουν οι ενέργειες που χρειάζεται για τη συνέχεια του παιχνιδιού (νέες κλάσεις LimitedRangeBoard και EndlessBoard) μέσω του BoardService [13].

Ουσιαστικά, ότι είχε η παλιά Board έχει τώρα η κλάση LimitedRangeBoard και έχουμε «συμμαζέψει» και την συνάρτηση movePosition() στα μηνύματα που έδειχνε παλιά, γιατί ήταν πολύ μεγάλα και με πολλές δομές επανάληψης που δεν φαινόταν καλά στο μάτι [[11]](https://stackoverflow.com/questions/22927920/java-printing-symbol-for-positive-number). Ίδια ακριβώς είναι και η EndlessBoard μέχρι στιγμής, γιατί δε γνωρίζουμε ακόμα πως θα εξελιχθεί.



Με την ευκολία που μας δίνει το service, έχουμε αφαιρέσει κομμάτια κώδικα που είχαμε στο παρελθόν για να γίνεται σωστά η δουλειά μας, π.χ. έχοντας το boardType ως όρισμα στην movePosition() που πλέον δε χρειάζεται.

Για να έχει νόημα το service, η κλάση Board έχει πλέον την εξής μορφή:

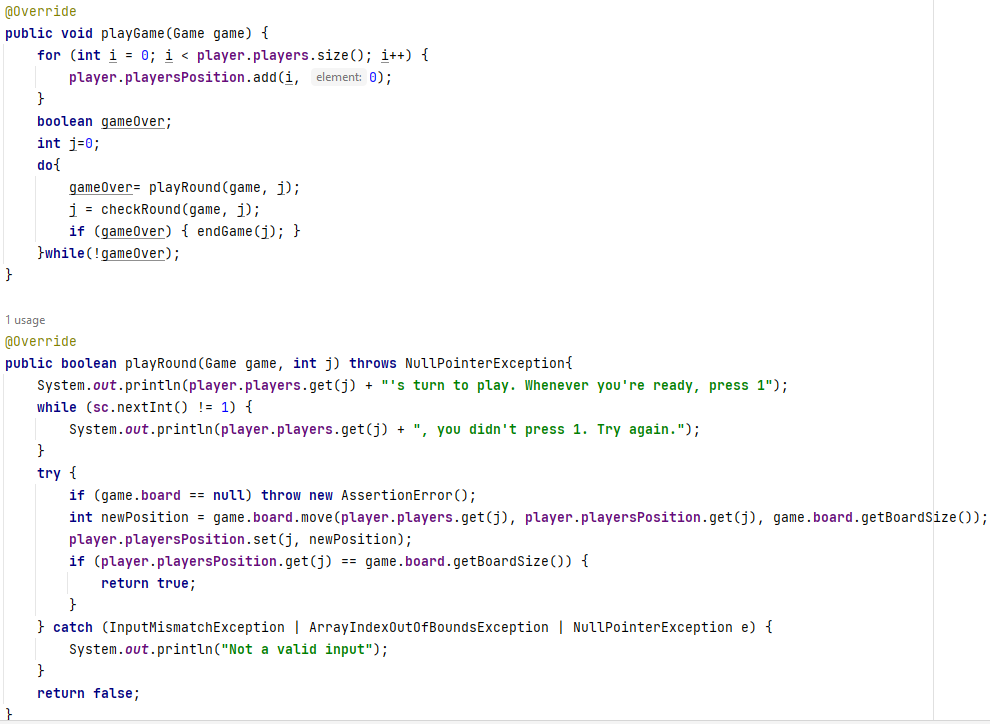


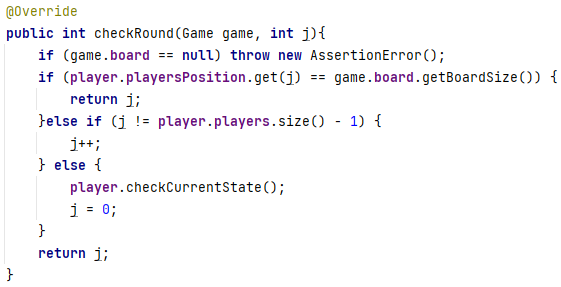
Ο ρόλος της νέας κλάσης Board είναι: με βάση το αντικείμενο που στέλνουμε μέσω του service για το ταμπλό που θέλουμε, να καλεί την κατάλληλη movePosition() για να μετακινείται ο παίκτης. Θα μπορούσαμε να πούμε ότι λειτουργούν αυτά τα δυο μαζί ως έναν διακόπτη που ορίζει ποια συνάρτηση ποιας κλάσης τύπου Board θα καλέσει. Παρατηρείται ότι η Board είναι ανεξάρτητη και από τις άλλες κλάσεις και από το service. Ας δούμε πως γίνεται όλη αυτή η λογική στην πράξη στην κλάση Game.

Η κλάση Game είναι πολύ διαφορετική πλέον με τις παραπάνω αλλαγές κι επίσης υπάρχει το πρώτο service που χρησιμοποιείται, της Board. Εδώ, όμως παρατηρείται πόσο απαραίτητο είναι:



Επίσης, βλέπουμε ότι τώρα έχουν λογική οι παραπάνω αλλαγές. Η συνάρτηση playGame() έχει χωριστεί σε playRound() και checkRound() για απλότητα, οι οποίες δέχονται σαν όρισμα το παιχνίδι που επέστρεψε η createGame().





Μη ολοκληρωμένη μορφή νέου json αρχείου για το νέο παιχνίδι [[12]](https://www.pluralsight.com/guides/get-javascript-objects-from-a-json-file):



# 3. Χρονοδιάγραμμα

18/4/2023:

Ελένη: Διορθώσεις, όπως διαγραφή των import συναρτήσεων και τύπου static και διαμόρφωση των συναρτήσεων και των constructor με βάση τα νέα δεδομένα. Δημιουργία νέων συναρτήσεων που σπάνε μια «μεγάλη» πχ playGame() με βοηθητικές συναρτήσεις τις playRound() και checkRound(). Δηλαδή έγιναν διορθώσεις οι οποίες θα βοηθήσουν το πρόγραμμα να υποδεχτεί τα Services πιο εύκολα.

24/4/2023:

Ομάδα: Σχεδιάσαμε το board και τις λειτουργίες του για να καταλάβουμε τι θα κάνουμε στο πρόγραμμα. Αποτυπώσαμε τις ενέργειες των τετραγώνων, καρτών και τους κανόνες. Οπτικοποίηση του ταμπλό και συμμόρφωση των ιδιοτήτων για να βγάζουν νόημα στην γεννήτρια μας.

25/4/2023:

Στέλλα: σύνταξη του report (Περίληψη, Εισαγωγή, Χρονοδιάγραμμα, Επίλογος)

Ομάδα: Σκέψεις και λίγες αλλαγές για να χρησιμοποιήσουμε το BoardService.

26/4/2023:

Αλκίνοος: Υλοποίηση της επιλογής board μεταξύ limited range και endless μέσω της χρήσης του BoardService και δημιουργία των κλάσεων Board, LimitedRangeBoard, EndlessBoard και του αντίστοιχου κώδικα, όπως και αλλαγές στην κλάση Game (συγκεκριμένα στις συναρτήσεις createGame() και playRound()), έτσι ώστε να υποστηρίζουν την χρήση του BoardService δεν έχει γραφτεί ακόμα η συνάρτηση movePosition() που αντιστοιχεί στην EndlessBoard.

Ελένη: Ολοκλήρωση του report.

# 4.Αποτελέσματα-Επίλογος

Επιλογικά, την τρίτη εβδομάδα έγινε ο αρχικός σχεδιασμός του δεύτερου τύπου παιχνιδιού, καθώς και ο ορισμός της λογικής και των κανόνων του. Αυτό θα μας βοηθήσει στις επόμενες εβδομάδες να έχουμε μια ξεκάθαρη εικόνα του στόχου μας και να διαμορφώσουμε ανάλογα και την αρχιτεκτονική του προγράμματος μας. Υπάρχουν πολλές παράμετροι στο συγκεκριμένο τύπο παιχνιδιού, όποτε είναι απαραίτητο να έχουμε καλογραμμένο και σωστά οργανωμένο κώδικα. Σε αυτό μας βοηθάνε τα services.

# 5.Βιβλιογραφία

|  |  |  |
| --- | --- | --- |
| [11] | Java printing symbol for positive number  <https://stackoverflow.com/questions/22927920/java-printing-symbol-for-positive-number> | |
| [12] | Get JavaScript Objects from a JSON File  <https://www.pluralsight.com/guides/get-javascript-objects-from-a-json-file> | |
| [13] | Service Example.rar (with different types of payments) | |
|  | | (δόθηκε στην αίθουσα) | |
|  |  | |