**­­ A blue and red logo

Description automatically generated**

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΚΩΝΣΤΑΝΤΙΝΟΣ ΚΟΥΚΟΣ Π21070

ΑΛΕΞΙΟΣ ΒΑΣΙΛΕΙΟΥ Π21009

ΚΡΟΙΤΟΡ ΚΑΤΑΡΤΖΙΟΥ ΙΩΑΝ Π21077

ΑΠΑΛΛΑΚΤΙΚΗ ΕΡΓΑΣΙΑ ΕΞΑΜΗΝΟΥ ΜΑΘΗΜΑΤΟΣ ΤΕΧΝΟΛΟΓΙΑΣ ΛΟΓΙΣΜΙΚΟΥ:

**ΠΕΙΡΑΙΑΣ**

**Ιούνιος 2024**

**ΠΡΟΛΟΓΟΣ – ΕΙΔΙΚΟ ΘΕΜΑ**

Η παρούσα εργασία αναπτύχθηκε ως μέρος του μαθήματος Τεχνολογίας Λογισμικού, με κύριο στόχο τη δημιουργία μιας ιστοσελίδας για ένα μεσιτικό γραφείο. Επιλέξαμε να βασιστούμε στην υπάρχουσα ιστοσελίδα [Spitogatos](https://www.spitogatos.gr/) για να προχωρήσουμε στον σχεδιασμό και την υλοποίηση του συστήματος. Η εργασία αποτελείται από τρία κύρια μέρη: τον κώδικα, την τεκμηρίωση και τα διαγράμματα.

**ΕΚΦΩΝΗΣΗ**

**ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ:** Με βάση το ειδικό θέμα της κάθε ομάδας καλείστε να αναπτύξετε λογισμικό, καθώς και να το συνοδεύσετε με την αντίστοιχη ανάπτυξή του σε μοντέλο κύκλου ζωής λογισμικού. Σύμφωνα με το ειδικό θέμα της εργασίας που έχει δοθεί στην ομάδα σας να γίνουν τα εξής:

1. Καταγραφή και παρουσίαση της ανάλυσης απαιτήσεων της εργασίας σας. Παραγωγή παρουσίασης απαιτήσεων και αρχιτεκτονικής δομής με τη χρήση των βασικών διαγραμμάτων της UML.

2. Ανάπτυξη των τεσσάρων φάσεων του αντικειμενοστραφούς μοντέλου ανάπτυξης λογισμικού (Rational Unified Process) χρησιμοποιώντας τα 9 διαγράμματα της UΜL με χρήση ενός εργαλείου CASE της επιλογής σας. Προτεινόμενα CA5E Τ00LS: Rational Rose, Visual Studio Ultimate, ArgoUML.

3. Ανάπτυξη της εργασίας. Η εργασία πρέπει να είναι λειτουργική και σωστά δομημένη. Χρήση της γλώσσας ή του εργαλείου που έχει δηλώσει η ομάδα σας. Εφόσον υπάρχει, χρήση της γλώσσας/τεχνολογίας που υποδηλώνει το θέμα της εργασίας σας. Σε κάθε περίπτωση, προτείνεται η χρήση κάποιας αντικειμενοστραφούς γλώσσας προγραμματισμού.

**Διευκρινήσεις:**

➢ Τα διαγράμματα που θα υλοποιήσετε κατά την ανάπτυξη των μοντέλων θα πρέπει υποχρεωτικά να σχετίζονται με την υλοποιημένη εργασία και το αντίστροφο.

# ΠΕΡΙΕΧΟΜΕΝΑ

[ΠΕΡΙΕΧΟΜΕΝΑ 2](#_Toc165024910)

[1. ΕΙΣΑΓΩΓΗ 4](#_Toc165024911)

[1.1 ΣΤΟΧΟΙ ΕΡΓΑΣΙΑΣ 4](#_Toc165024912)

[1.2 ΟΡΙΣΜΟΣ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΠΡΟΣ ΕΠΙΛΥΣΗ 4](#_Toc165024913)

[2. ΣΥΝΤΟΜΗ ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ RUP 4](#_Toc165024914)

[3. ΦΑΣΗ: ΕΝΑΡΞΗ (INCEPTION) 5](#_Toc165024915)

[3.1 ΣΥΛΛΗΨΗ ΑΠΑΙΤΗΣΕΩΝ 5](#_Toc165024916)

[3.2 ΑΝΑΛΥΣΗ ΣΧΕΔΙΑΣΜΟΥ 5](#_Toc165024917)

[3.2.1 Διαγράμματα Περιπτώσεων Χρήσης 5](#_Toc165024918)

[3.2.2 Διαγράμματα Τάξεων 5](#_Toc165024919)

[4. ΦΑΣΗ:ΕΚΠΝΟΗΣΗ ΜΕΛΕΤΗΣ (ELABORATION) 5](#_Toc165024920)

[ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ 5](#_Toc165024921)

# 1. ΕΙΣΑΓΩΓΗ

## ΣΤΟΧΟΙ ΕΡΓΑΣΙΑΣ

Βασικός στόχος της εργασίας είναι η υλοποίηση μιας ιστοσελίδας μεσιτικού γραφείου (spitoskilos), στην οποία οι χρήστες θα έχουν την δυνατότητα να εκδηλώσουν ενδιαφέρον για αγγελίες ακινήτων, αλλά και να καταχωρούν αγγελίες. Επιπλέον οι διαχειριστές μπορούν να επεξεργαστούν και να διαγράψουν χρήστες καθώς και τις αγγελίες τους.

## 1.2 ΟΡΙΣΜΟΣ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΠΡΟΣ ΕΠΙΛΥΣΗ

# 2. ΣΥΝΤΟΜΗ ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ RUP

Είναι μια από τις πρώτες μεθοδολογίες που συνέδεσε την ανάπτυξη λογισμικού με την αρχιτεκτονική βασισμένη σε μοντέλα και τον αντικειμενοστραφή προγραμματισμό. Το RUP επικεντρώνεται στην ανάπτυξη λογισμικού μέσω ενός σειράς επαναλαμβανόμενων διαδικασιών ανάπτυξης που ονομάζονται "φάσεις". Κάθε φάση αποτελείται από ορισμένες δραστηριότητες και παράγει συγκεκριμένα αποτελέσματα που χρησιμοποιούνται ως είσοδος για την επόμενη φάση. Το RUP επίσης παρέχει μια σειρά από οδηγίες, πρότυπα και πρακτικές για την υλοποίηση των φάσεων:

1. **Ανάλυση:** Συλλέγονται και αναλύονται οι απαιτήσεις του συστήματος, καθώς και οι λειτουργικές και μη λειτουργικές απαιτήσεις του πελάτη.
2. **Σχεδίαση:** Αναπτύσσεται η αρχιτεκτονική του συστήματος, καθώς και οι λεπτομερείς σχεδιαστικές λύσεις.
3. **Υλοποίηση:** Ο κώδικας του συστήματος αναπτύσσεται βάσει των σχεδιαστικών προδιαγραφών.
4. **Δοκιμές:** Το λογισμικό υποβάλλεται σε δοκιμές για την επιβεβαίωση της λειτουργικότητάς του και την ανίχνευση σφαλμάτων.
5. **Ανάπτυξη:** Επανεξετάζονται οι απαιτήσεις και τα σχέδια, και πραγματοποιούνται αναθεωρήσεις και ενημερώσεις κατά τη διάρκεια της ανάπτυξης.

Επιπλέον προβλέπει τα παρακάτω χαρακτηριστικά:

1. **Έναρξη (Inception**): Καθορίζει την προοπτική του έργου.
2. **Εκπόνηση µμελέτης (Elaboration):** Σχεδιασμός των απαιτούμενων δραστηριοτήτων και πόρων. Καθορισμός των χαρακτηριστικών και σχεδιασμός της αρχιτεκτονικής.
3. **Κατασκευή (Construction):** Ανάπτυξη του προϊόντος σε µία σειρά βηµατικών επαναλήψεων.
4. 4. **Μετάβαση (Transition):** Προμήθευση του προϊόντος στην κοινότητα χρηστών (παραγωγή, διανομή, εκπαίδευση).

# 3. ΦΑΣΗ: ΕΝΑΡΞΗ (INCEPTION)

## 3.1 ΣΥΛΛΗΨΗ ΑΠΑΙΤΗΣΕΩΝ

1. **Λειτουργικές απαιτήσεις:**

Όσον αφορά τις λειτουργικές απαιτήσεις, το σύστημα θα πρέπει να είναι ικανό να προβάλει τις αγγελίες, συμπεριλαμβανομένης και της δυνατότητας προβολής σε μορφή χάρτη, και να προσφέρει την δυνατότητα καταχώρησης νέας αγγελίας στους συνδεδεμένους (και μη) χρήστες, καθώς και να επιτρέπει σε έναν επισκέπτη να δημιουργήσει λογαριασμό. Επιπλέον θα διατηρείται το login session (credentials, profile και preferences) του κάθε χρήστη, αφότου συνδεθεί, καθ’ όλη την διάρκεια της περιήγησής του στην ιστοσελίδα. Τα δεδομένα για τα χαρακτηριστικά των ακινήτων θα εξάγονται από μία βάση δεδομένων, πράγμα που προσφέρει τη δυνατότητα φιλτραρίσματος των ακινήτων βάση διάφορων χαρακτηριστικών, επιτρέποντας στους χρήστες να επιλέγουν τα ακίνητα που πληρούν τις προτιμήσεις και τις ανάγκες τους. Από την άλλη οτιδήποτε αφορά την πληροφορίες για τον ίδιο το χρήστη (username, preferences, password κλπ.), θα αποθηκεύονται σε μία διαφορετική βάση. Για τον διαχειριστή οι λειτουργίες συμπεριλαμβάνουν όλα τα προαναφερόμενα και επιπλέον κάθε είδους επεξεργασία χρηστών και αγγελιών, καθώς και της διαγραφής.

(Αφορούν τις λειτουργίες ή τις ενέργειες που πρέπει να εκτελεί το σύστημα, όπως λειτουργίες δημιουργίας, ανάγνωσης, ενημέρωσης και διαγραφής δεδομένων, εκτέλεση συγκεκριμένων διεργασιών ή υπολογισμών, και δυνατότητες αναζήτησης και φιλτραρίσματος δεδομένων.)

1. **Μη λειτουργικές απαιτήσεις:**

Όσον αφορά την τις μη λειτουργικές απαιτήσεις, η ιστοσελίδα θα πρέπει να ανταποκρίνεται γρήγορα στη περιήγηση των χρηστών. Ιδιαίτερα θα χρησιμοποιηθεί non-SQL database για την αποθήκευση των χαρακτηριστικών των ακινήτων. Η Redis είναι ιδανική για την αποθήκευση δεδομένων σε μνήμη και τη γρήγορη ανάκτησή τους μέσω απλών δομών δεδομένων, όπως key-value pairs, λίστες και σύνολα. Αυτό επιτρέπει την εκτέλεση γρήγορων ερωτημάτων χωρίς την ανάγκη για πολύπλοκες διαδικασίες ζήτησης δεδομένων, βοηθώντας στην αύξηση της απόδοσης του συστήματος.

Επιπλέον τα passwords θα είναι hashed ώστε να παραμένουν ασφαλή από ανεπιθύμητη πρόσβαση.

Όπως είναι προφανές στα πλαίσια της εργασίας αυτής και δεδομένου του περιορισμένου χρόνου, δεν είναι δυνατό να επιτευχθεί η βέλτιστη απόδοση και ασφάλεια της ιστοσελίδας, καθώς αυτό συμπεριλαμβάνει και γνώσεις τις οποίες δεν έχουμε αποκτήσει ακόμα.

Αφορούν την ποιότητα του λογισμικού και το περιβάλλον στο οποίο θα λειτουργεί, όπως η απόδοση, η ασφάλεια, η αξιοπιστία, η διαθεσιμότητα, η συμμόρφωση με τους κανονισμούς και οι περιορισμοί σχετικά με την απόδοση, την ασφάλεια και την αξιοπιστία.

1. **Απαιτήσεις περιβάλλοντος:**

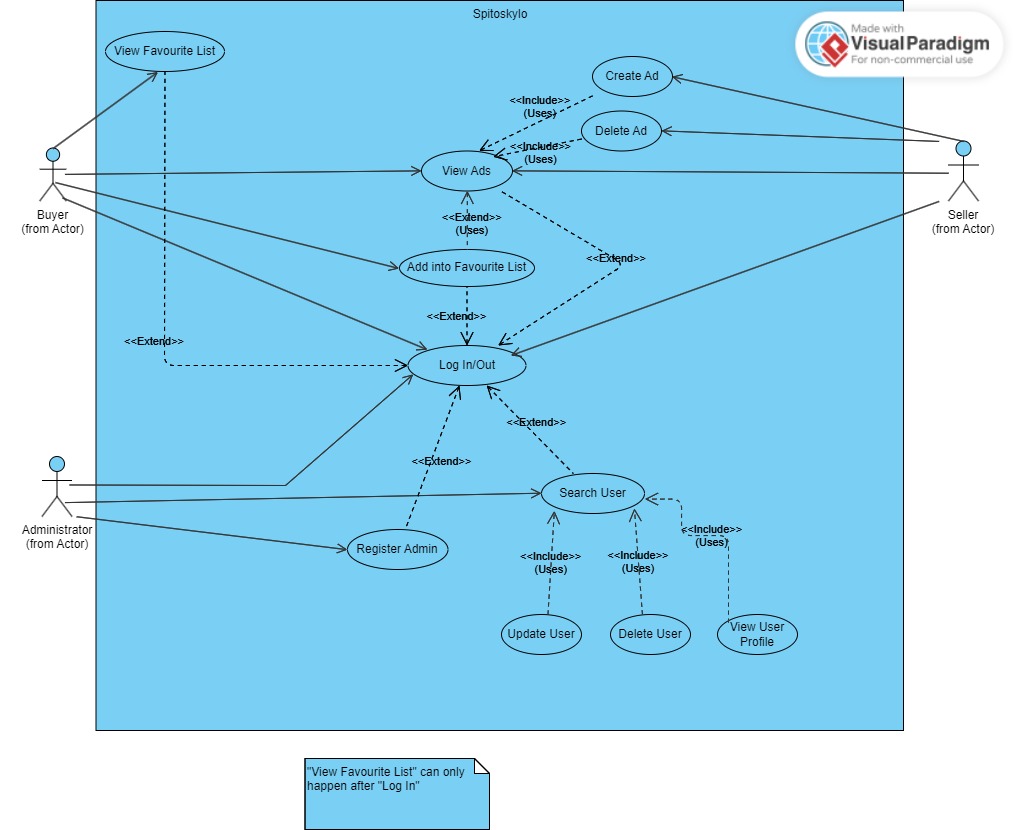
Όσον αφορά τις απαιτήσεις περιβάλλοντος, η ιστοσελίδα θα φιλοξενείται σε ένα καθορισμένο domain και θα λειτουργεί σε έναν διακομιστή. Το frontend θα αναπτυχθεί με τη χρήση του React, ενώ το backend θα υλοποιηθεί με τη Java.

Αφορούν το περιβάλλον στο οποίο θα λειτουργεί το σύστημα, όπως οι υπάρχουσες τεχνολογίες, οι υπάρχουσες εφαρμογές ή συστήματα, και οι περιορισμοί σχετικά με τις πλατφόρμες λειτουργικού συστήματος ή το υλικό.

## 3.2 ΑΝΑΛΥΣΗ ΣΧΕΔΙΑΣΜΟΥ

## 3.2.1 Διαγράμματα Περιπτώσεων Χρήσης

**(1η Έκδοση)**



**ΠΕΡΙΠΤΩΣΗ ΣΥΝΔΕΣΗΣ BUYER/SELLER/ADMINISTRATOR:**

Όταν ο buyer/seller/administrator κάνει σύνδεση στην πλατφόρμα γίνεται η επιβεβαίωση των στοιχείων του και αντίστοιχα αν είναι σωστά εισέρχεται στην πλατφόρμα ενώ αν είναι λάθος του εμφανίζει το ανάλογο μήνυμα.

**ΠΕΡΙΠΤΩΣΗ ΠΡΟΒΟΛΗΣ ΛΙΣΤΑΣ ΑΓΑΠΗΜΕΝΩΝ:**

Ο buyer για να δει τις αγγελίες που έχει βάλει στην λίστα με της αγαπημένες πρέπει να είναι ήδη συνδεδεμένος στον λογαριασμό αλλιώς δεν θα είναι δυνατή η προβολή τους.

**ΠΕΡΙΠΤΩΣΗ ΠΡΟΣΘΗΚΗΣ ΣΤΗΝ ΛΙΣΤΑ ΑΓΑΠΗΜΕΝΩΝ:**

Ο buyer για να προσθέσει μια αγγελία στην λίστα με τις αγαπημένες, πρέπει να είναι ήδη συνδεδεμένος στον λογαριασμό αλλιώς δεν θα είναι δυνατή η προσθήκη, καθώς επίσης πρέπει να του προβάλλεται κάποια αγγελία μπροστά του.

**ΠΕΡΙΠΤΩΣΗ ΠΡΟΒΟΛΗΣ ΑΓΓΕΛΙΑΣ:**

Ο buyer για να προσθέσει μια αγγελία στην λίστα με τις αγαπημένες, πρέπει να είναι ήδη συνδεδεμένος στον λογαριασμό αλλιώς δεν θα είναι δυνατή η προβολή, καθώς επίσης πρέπει να του προβάλλεται κάποια αγγελία μπροστά του. Επιπλέον για να γίνει η προβολή μιας αγγελίας, πρέπει να έχει δημιουργηθεί μία από τον seller. Τέλος, ο seller για να διαγράψει μια αγγελία πρέπει να του προβληθεί πρώτα.

**ΠΕΡΙΠΤΩΣΗ ΔΗΜΙΟΥΡΓΙΑΣ ΑΓΓΕΛΙΑΣ:**

Ο seller για να δημιουργήσει μια αγγελία πρέπει να του προβάλλεται κάποια αγγελία μπροστά του. (για να του προβληθεί κάποια, έχει αναφερθεί παραπάνω τι χρειάζεται).

**ΠΕΡΙΠΤΩΣΗ ΔΙΑΓΡΑΦΗΣ ΑΓΓΕΛΙΑΣ:**

Ο seller για να διαγράψει μια αγγελία πρέπει να του προβάλλεται κάποια αγγελία μπροστά του. (για να του προβληθεί κάποια, έχει αναφερθεί παραπάνω τι χρειάζεται).

**ΠΕΡΙΠΤΩΣΗ ΔΗΜΙΟΥΡΓΙΑΣ ΧΡΗΣΤΗ:**

Ο administrator για να δημιουργήσει έναν χρήστη (seller ή buyer ή admin), πρέπει να είναι ήδη συνδεδεμένος στον λογαριασμό αλλιώς δεν θα είναι δυνατή η δημιουργία.

**ΠΕΡΙΠΤΩΣΗ ΕΓΓΡΑΦΗΣ ADMINISTRATOR:**

Ο administrator για να εγγράψει έναν administrator πρέπει να είναι ήδη συνδεδεμένος στον λογαριασμό αλλιώς δεν θα είναι δυνατή η αναζήτηση.

**ΠΕΡΙΠΤΩΣΗ ΑΝΑΖΗΤΗΣΗΣ ΧΡΗΣΤΗ:**

Ο administrator για να αναζητήσει έναν χρήστη πρέπει να είναι ήδη συνδεδεμένος στον λογαριασμό αλλιώς δεν θα είναι δυνατή η αναζήτηση.

**ΠΕΡΙΠΤΩΣΗ ΕΝΗΜΕΡΩΣΗΣ/ΔΙΑΓΡΑΦΗΣ/ΠΡΟΒΟΛΗΣ ΧΡΗΣΤΗ:**

Ο administrator ενημερώνει τα στοιχεία ενός χρήστη, να τον διαγράψει και να δει το προφίλ του. Οι διαδικασίες αυτές για να εκτελεστούν χρησιμοποιούν την διαδικασία της αναζήτησης του χρήστη.

## 3.2.2 Διαγράμματα Τάξεων

**(1η Έκδοση)**

A diagram of a visual parametric code

Description automatically generated

Σημειώνεται ότι σε όλες τις κλάσεις υλοποιήθηκαν οι κατάλληλοί getters/setters και για λόγους συντομίας αυτοί δεν αναγράφονται παρακάτω. Επιπλέον κάποια χαρακτηριστικά δεν αναλύονται καθώς θεωρούνται ως προφανή (π.χ. dateOfBirth 🡪 ημερομηνία γέννησης)

**User**

Αποτελεί την αφηρημένη κλάση του χρήστη η οποία κληρονομείται στην συνέχεια από τις κλάσεις Buyer, Seller και Administrator. Η κλάση αυτή αποφασίστηκε να μην έχει αντικείμενα καθώς οι τελικοί χρήστες θα ανήκουν σε μία από τις κλάσεις που αναφέρονται παρακάτω και οι οποίες έχουν όλες κοινά τα χαρακτηριστικά της κλάσης User. Περιλαμβάνει τα βασικά χαρακτηριστικά ενός χρήστη, ενώ το type εκφράζει τι τύπος χρήστη είναι αυτός που βρίσκεται εκείνη την στιγμή συνδεδεμένος (Buyer, Seller ή Administrator), καθώς το creation\_date αντιπροσωπεύει την ημερομηνία δημιουργίας του χρήστη. Επιπλέον το username είναι μοναδικό για κάθε χρήστη. Όσον αφορά τις μεθόδους, οι login() και logout() πραγματεύονται αντίστοιχα την σύνδεση και την αποσύνδεση του χρήστη από την συνεδρία.

**Buyer**

Αποτελεί την κλάση παιδί της User και εκφράζει τους χρήστες που επιθυμούν να αγοράσουν/ενοικιάσουν κάποιο ακίνητο. Το preferences αναφέρεται στις προτιμήσεις του χρήστη (π.χ. dark mode), ενώ το adID αναφέρεται στις αγγελίες για τις οποίες ο χρήστης ενδιαφέρεται. . Όσον αφορά τις μεθόδους, η showAvailableAds() εμφανίζει όλες τις διαθέσιμες αγγελίες, η addToFavourite() προσθέτει την αγγελία στην λίστα αγαπημένων του χρήστη και η viewFavouriteList() εμφανίζει την λίστα αγαπημένων του συγκεκριμένου χρήστη.

**Seller**

Αποτελεί την κλάση παιδί της User και εκφράζει τους χρήστες που επιθυμούν να καταχωρήσουν προς αγορά/ενοικίαση κάποιο ακίνητο. Το adID σε αυτήν την περίπτωση αναφέρεται στις αγγελίες που έχει καταχωρήσει ο Seller.

**Admin**

Αποτελεί κλάση παιδί της User και εκφράζει τους διαχειριστές. Αυτοί έχουν την δυνατότητα να επεξεργαστούν ή να διαγράψουν υπάρχοντες χρήστες ή αγγελίες.

**Ad**

Αποτελεί την αφηρημένη κλάση της αγγελίας η οποία κληρονομείται στη συνέχεια από τις κλάσεις Residence και Land. Ο διαχωρισμός αυτός συνέβη καθώς πάνω στην υπόθεση ότι ένα ακίνητο θα κατηγοριοποιείται είτε ως κατοικία είτε ως έκταση. To adID είναι μοναδικό αναγνωριστικό για κάθε αγγελία, ενώ το adType αναφέρεται στην κατάσταση που προσφέρεται το ακίνητο, δηλαδή αν προσφέρεται ως ενοικίαση ή πώληση. Το dateOfUpload αναφέρεται στην ημερομηνία στην οποία αναρτήθηκε η αγγελία, ενώ το availability αναφέρεται στην ημερομηνία μετά την οποία το ακίνητο είναι διαθέσιμο να πωληθεί/ενοικιαστεί. Το numOfInterested αναφέρεται στον αριθμό των ατόμων που έχουν προσθέσει το ακίνητο στα αγαπημένα.

**Residence**

Αποτελεί την κλάση παιδί της Ad και εκφράζει τα ακίνητα τα οποία κατηγοριοποιούνται ως κατοικίες. Το dateOfConstruction αναφέρεται στην ημερομηνία κατασκευής της κατοικίας, ενώ το propertyCategory στη κατηγορία που καταλογίζεται το σπίτι (π.χ. μονοκατοικία, διαμέρισμα κλπ.). Τα numOfBathrooms και numOfBedrooms αναφέρονται στον αριθμό των μπάνιων και δωματίων αντίστοιχα.

**Land**

Αποτελεί την κλάση παιδί της Ad και εκφράζει τα ακίνητα τα οποία κατηγοριοποιούνται ως εκτάσεις γης. Το STI (συντελεστής δόμησης) είναι ορίεται ακολούθως: ‘’Σε οικόπεδο (παράδειγμα) 1.000τμ με σ. δ. 0,6 μπορούμε να κτίσουμε συνολικά 1.000 Χ 0,6 = 600τμ. Τα υπόγεια "επίσημα" γκαράζ προσμετρούνε στο σύνολο. Εδώ πρέπει να σημειώσουμε, ότι διατηρητέα κτήρια (που χαρακτηρίζονται έτσι από την Πολεοδομία) και δεν μπορούν να κατεδαφιστούν, οπότε και να ανοικοδομηθούν, μπορούν ωστόσο να πουλήσουν σε κάποιον τον σ. δ. που θα είχαν και αυτός να τον μεταφέρει σε άλλη περιοχή και σε άλλο κτήριο προς ανέγερση (Μεταφορά Συντελεστή)’’. Το accessibleByRoad λαμβάνει τιμές True αν η έκταση είναι προσβάσιμη μέσω δρόμου και False σε αντίθετη περίπτωση.

4. ΦΑΣΗ:ΕΚΠΝΟΗΣΗ ΜΕΛΕΤΗΣ (ELABORATION)

# ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ

**A. ΕΛΛΗΝΙΚΗ**

1. Σέργιος Θεοδωρίδης Άγγελος Πικράκης Κωνσταντίνος Κουτρούμπας Διονύσης Κάβουρας “*Εισαγωγή στην Αναγνώριση Προτύπων με MATLAB*”, ΙΑΤΡΙΚΕΣ ΕΚΔΟΣΕΙΣ Π.Χ. ΠΑΣΧΑΛΙΔΗΣ, Αθήνα, 2011.

**Β. ΞΕΝΟΓΛΩΣΗ**

1. Scikit-learn, Cross-validation: evaluation estimator performance , scikit-learn developers, <https://scikit-learn.org/stable/modules/cross_validation.html>