

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΤΑΝΕΜΗΜΕΝΑ ΣΥΣΤΗΜΑΤΑ
Εαρινό Εξάμηνο 2018-2019
Καθ. Βάνα Καλογεράκη

Υποχρεωτική εργασία

Τα τελευταία χρόνια, λόγω της πληθώρας των αστικών συστημάτων που έχουμε διαθέσιμα, η ιδέα του να καταλαβαίνουμε τι συμβαίνει μέσα στην πόλη που ζούμε αποκτά ιδιαίτερη σημασία. Το πλεονέκτημα αυτών των συστημάτων έγκειται στην δυνατότητα να μπορούμε με σχετικά χαμηλό κόστος να έχουμε εγκατεστημένο ένα σύνολο από αισθητήρες μέσα στην πόλη, οι οποίοι σε πραγματικό χρόνο να μας παρέχουν πληροφορία τόσο για την κατάσταση της πόλης (αισθητήρες κυκλοφοριακής συμφόρησης, αισθητήρες επιπέδων ρύπανσης, αισθητήρες επιπέδων ήχου, κλπ.), όσο και για τις υποδομές της πόλης (αισθητήρες θέσης λεωφορείων, αισθητήρες διαθέσιμων ποδηλάτων κοινής χρήσης κλπ).

Τα συστήματα αυτά, αξιοποιούν ένα σύνολο από αισθητήρες, οι οποίοι είναι ικανοί, να μας επιστρέψουν διάφορες πληροφορίες, όπως προαναφέραμε. Έτσι, κάθε φορά που αυτή η πληροφορία είναι διαθέσιμη από τον αισθητήρα (όπως π.χ. η άφιξη του ενός λεωφορείου σε μια συγκεκριμένη στάση) προωθείται κατάλληλα σε μια αντίστοιχη υποδομή που επιτρέπει σε όποιον επιθυμεί να “καταναλώσει” την πληροφορία αυτή. Έτσι, για παράδειγμα, κάποιος μπορεί να οπτικοποιήσει αυτήν την πληροφορία (π.χ. την θέση του λεωφορείου πάνω σε ένα χάρτη της πόλης).

Στην παρούσα εργασία καλείστε να φτιάξετε ένα publish/subscribe σύστημα με χρήση της java 8, στο οποίο η πληροφορία που θέλουμε να αντλήσουμε είναι οι θέσεις των λεωφορείων μέσα στην πόλη. Λόγω του μεγάλου αριθμού λεωφορειακών γραμμών που υπάρχουν στην πόλη, καλείστε να υλοποιήσετε ένα έξυπνο σύστημα το οποίο, δεδομένου ενός συνόλου από λεωφορειακές γραμμές που ενδιαφέρει τον χρήστη, θα του επιστρέφει κατάλληλα την πληροφορία γι’αυτές τις γραμμές στο κινητό του. Καλείστε λοιπόν να υλοποιήσετε ένα publish/subscribe σύστημα το οποίο είναι υπεύθυνο να μεταφέρει την πληροφορία αυτή σχετικά με τις λεωφορειακές γραμμές που ενδιαφέρουν το χρήστη, έτσι ώστε να είναι δυνατή η άμεση προσπέλαση αλλά και η επιστροφή στο κινητό του χρήστη που την ζητά.

Η ανάπτυξη του συστήματος θα σας επιτρέψει να εξοικειωθείτε τόσο με την έννοια του publish/subscribe, όσο και με το να γνωρίσετε το περιβάλλον Android. Για την διευκόλυνσή σας, η ανάπτυξη αυτού του συστήματος θα γίνει σε δύο φάσεις: Η πρώτη φάση αφορά **την υλοποίηση του publish/subscribe framework που θα είναι υπεύθυνο να υλοποιήσει τον αλγόριθμο της μεταφοράς των δεδομένων**. Η δεύτερη φάση αφορά **τη δημιουργία μιας εφαρμογής Android, η οποία μέσω των κατάλληλων διεπαφών που θα έχετε φροντίσει να υλοποιήσετε στην πρώτη φάση, θα μπορεί να καλεί το framework, έτσι ώστε να εκτελείται ο**

αλγόριθμος ανεύρεσης της πληροφορίας για την θέση ενός συνόλου λεωφορείων, και στη συνέχεια αυτό το αποτέλεσμα να οπτικοποιείται στο κινητό. Παρακάτω ακολουθούν λεπτομερείς οδηγίες για την κάθε φάση και τι καλείστε να υλοποιήσετε σε αυτές.

Publish / Subscribe Μοντέλο

Το Publish / Subscribe μοντέλο είναι ένα προγραμματιστικό μοντέλο που επιτρέπει την αποστολή και λήψη δεδομένων που πληρούν συγκεκριμένα κριτήρια. Το πλεονέκτημα του publish / subscribe συστήματος είναι ότι επιτρέπει την άμεση αποστολή και προώθηση των δεδομένων σε πραγματικό χρόνο μέσω των συναρτήσεων “push” και “pull”. Υπό την προϋπόθεση ότι κάθε “push” συνάρτηση είναι ανεξάρτητη από την άλλη, θα πρέπει να υπάρξει κατάλληλη μέριμνα ώστε ο broker να μπορεί να δέχεται δεδομένα από διαφορετικούς publishers παράλληλα, με μοναδικό περιορισμό τον διαφορετικό ρυθμό publishing των δεδομένων από κάθε publisher. Επίσης, ο παραλληλισμός είναι απαραίτητος γιατί προσφέρει κάποια δυνατότητα ταυτόχρονης αποστολής των δεδομένων τόσο από τους publishers στους brokers, όσο και από τους brokers στους consumers. Το publish/subscribe μοντέλο στηρίζεται στη χρήση δύο συναρτήσεων:

**push(topic,value) -> [broker]
pull(topic,[broker]) -> [topic,value]**

- "Push" συνάρτηση: Ουσιαστικά προωθεί στον broker ένα ζεύγος key(ή αλλιώς topic)/value το οποίο και αποθηκεύεται σε κατάλληλη δομή ώστε να μπορεί να ανευρεθεί. Η είσοδος στη συνάρτηση push μπορεί να είναι ο αριθμός της γραμμής και η θέση του λεωφορείου κλπ., που έχει τη μορφή (κλειδί, τιμή). Η push συνάρτηση μπορεί να εκτελείται παράλληλα, πάνω σε διαφορετική είσοδο δεδομένων και από διαφορετικούς κόμβους (που ουσιαστικά αντιπροσωπεύουν τον κάθε ξεχωριστό αισθητήρα). Ο βαθμός παραλληλίας εξαρτάται από την εφαρμογή και μπορεί να την ορίσει ο χρήστης.
- "Pull" συνάρτηση: Η συνάρτηση “pull” είναι υπεύθυνη ώστε να πάρει - “καταναλώσει” τα values που σχετίζονται με το ίδιο κλειδί και να τα προωθήσει κατάλληλα στον subscriber (στην περίπτωση μας ο χρήστης που ενδιαφέρεται για μια ή και περισσότερες λεωφορειακές γραμμές). Για κάθε ξεχωριστό κλειδί δημιουργείται μια λίστα από τις τιμές που αντιστοιχούν σε αυτό το κλειδί. Αυτό το value στέλνεται σε όλους τους subscribers που είναι εγγεγραμμένοι στον συγκεκριμένο broker ταυτόχρονα και ενδιαφέρονται για το ίδιο κλειδί, ώστε να ακολουθήσει το επόμενο κατά σειρά δεδομένο.

Για τις ανάγκες του publish / subscribe μοντέλου χρειάζεται να κατασκευάσουμε τρία components, τον *publisher node*, τους *broker nodes* και τον *consumer node*.

1. **Publisher node:** Είναι υπεύθυνος να παίρνει τα δεδομένα από το κατάλληλο source. Ο κόμβος αυτός με κατάλληλο τρόπο και σε κατάλληλο χρονικό διάστημα στέλνει τα δεδομένα του στους brokers. Ουσιαστικά αυτό που κάνει είναι να κάνει push τα δεδομένα για τα κλειδιά για τα οποία είναι υπεύθυνος, κατάλληλα στους brokers που είναι υπεύθυνοι για τα συγκεκριμένα κλειδιά. Ο publisher node κατά την έναρξη της λειτουργίας του θα πρέπει να γνωρίζει για

ποια κλειδιά είναι υπεύθυνος καθώς επίσης και όλη την απαραίτητη πληροφορία για τους brokers. Θα πρέπει να γνωρίζει για κάθε διαθέσιμο broker για ποια κλειδιά είναι υπεύθυνος. Αυτό θα επιτευχθεί κατά τη στιγμή που εκκινείται η λειτουργία του κάθε broker. Γι'αυτό το λόγο θα πρέπει να γίνει **κάποιος μορφής συγχρονισμός** όταν κάθε broker ενημερώνει τον consumer για ποια κλειδιά είναι υπεύθυνος. Έτσι, όταν γίνει εκπομπή των δεδομένων, τότε αυτά θα μεταφερθούν στους αντίστοιχους brokers και όχι σε όλους. Αν για τον οποιονδήποτε λόγο ο publisher node δεν στέλνει αποτελέσματα (πχ γιατί ο sensor χάλασε και δεν στέλνει δεδομένα), τότε ο publisher node επιστρέφει κατάλληλο μήνυμα το οποίο θα πρέπει να μεταφερθεί στον broker και αφού γίνει κατάλληλα handle να προωθηθεί στους consumer nodes.

2. **Broker Nodes:** Οι κόμβοι αυτοί είναι υπεύθυνοι για ένα εύρος από topics (στην περίπτωση μας: γραμμές λεωφορείων). Προκειμένου, λοιπόν να γίνει μια ισοκατανομή στο πόσα topics αναλαμβάνει κάθε broker node, χρειάζεται να γίνει κάποιος μορφής hashing. Στη συνέχεια, χρησιμοποιώντας μια hash function(π.χ. SHA1 ή MD5) παίρνουμε το Hash του String BusLineID και το Hash του IP+Port του. Έτσι ο κάθε broker θα είναι υπεύθυνος για όσα hashes εγγραφών είναι μικρότερα από το hash του IP+Port του. (Προσοχή για ποια hashes αντιστοιχούν στον broker με το μικρότερο hash, θα χρειαστεί η χρήση mod). Αυτοί οι κόμβοι ενημερώνουν κατάλληλα τους publisher nodes για το ποια κλειδιά είναι υπεύθυνοι και στην συνέχεια ανοίγουν επικοινωνία με αυτούς (με τους publisher nodes) έτσι ώστε να είναι σε θέση να λάβουν την πληροφορία όταν αυτή γίνει διαθέσιμη και να την προωθήσουν στους κατάλληλους consumers που έχουν γίνει register επάνω τους. Επίσης, σε κάθε νέα σύνδεση κάποιου νέου consumer τον ενημερώνουν κατάλληλα για το ποιοι είναι οι υπολοίποι brokers και για ποια topics είναι υπεύθυνοι. Ένα πολύ σημαντικό στοιχείο των brokers είναι ότι η πληροφορία την οποία αποστέλλουν στους consumer nodes πρέπει να την στείλουν ταυτόχρονα σε όλους. Για παράδειγμα, αν έχουμε δύο consumer nodes, που ενδιαφέρονται για το ίδιο topic (την ίδια λεωφορειακή γραμμή) τότε θα πρέπει η αποστολή των δεδομένων να γίνει ταυτόχρονα και στους δύο κάνοντας χρήση των αρχών πολυνηματικού προγραμματισμού.
3. **Consumer Node:** Ο Consumer Node είναι ουσιαστικά το κινητό το οποίο δέχεται την πληροφορία σχετικά με την γραμμή του λεωφορείου και την αντίστοιχη θέση του. Δέχεται από τους broker nodes tuples της μορφής: <busLine, (Bus LatLng)>. Ο consumer node βρίσκεται μόνιμα συνδεδεμένος με τον broker που είναι υπεύθυνος για την εκάστοτε γραμμή λεωφορείου. Επειδή, αυτό όμως δεν είναι πάντα γνωστό, ο consumer ρωτάει κατάλληλα όλους τους διαθέσιμους brokers για το ποιος είναι υπεύθυνος και για ποια κλειδιά. Συνεπώς, κατά την πρώτη επικοινωνία ενός consumer node μ'έναν από τους broker, λαμβάνει από αυτόν την πληροφορία για το ποιοι είναι οι υπόλοιποι brokers και για ποια κλειδιά. Ουσιαστικά επιστρέφεται ένα αντικείμενο της μορφής Info {ListOfBrokers {IP,Port} , < BrokerId, ResponsibilityLine>}. Με βάση αυτό το object, λοιπόν, ο consumer στην συνέχεια, είναι εύκολο να επιλέξει τον κατάλληλο broker ο οποίος θα του επιστρέψει την αντίστοιχη πληροφορία για μια δεδομένη γραμμή λεωφορείου.

Υλοποίηση συστήματος publish/subscribe πάνω από Java 8 [3]. Ο τρόπος λειτουργίας του publish/subscribe θα γίνεται ως εξής:

1. Σε κάθε broker κόμβο θα υπάρχει ένα instance του συστήματος που θα προωθεί κατάλληλα τα δεδομένα.
2. Κάθε instance (consumer node, broker nodes, consumer nodes) θα ακούει σε κάποιο προκαθορισμένο port για συνδέσεις.
3. Όταν λάβει ο broker node ένα query από τον χρήστη, αρχικά κοιτά αν είναι ήδη συνδεδεμένος μαζί του (αν έχει γίνει register ο consumer node στον συγκεκριμένο broker). Αν υπάρχει ήδη σύνδεση, τότε απλά φροντίζει να περιμένει κατάλληλα μέχρι να υπάρχουν διαθέσιμα δεδομένα τα οποία μπορεί να στείλει. Αν όχι, τότε δημιουργείται μια νέα σύνδεση και παράλληλα επιστρέφονται στον consumer η λίστα με τους υπόλοιπους brokers και για ποια κλειδιά είναι υπεύθυνοι. Όταν υπάρξουν τα δεδομένα, τότε ο broker τα στέλνει κατάλληλα στον consumer. Αν δεν έχει, επιστρέφει κατάλληλο μήνυμα με το οποίο ενημερώνει τον χρήστη.
4. Κάθε broker θα προωθεί τα δεδομένα για το συγκεκριμένο εύρος δεδομένων για το οποίο είναι υπεύθυνος.
5. Όταν ο consumer node παραλάβει την πληροφορία την οπτικοποιεί κατάλληλα.

Android Application

Θα αναπτύξετε μια εφαρμογή που θα εκτελείται σε συσκευές με λειτουργικό Android και θα εμφανίζει στο χάρτη τις θέσεις των λεωφορείων που ζητά ο χρήστης σε πραγματικό χρόνο. Η εφαρμογή θα υλοποιηθεί στην πλατφόρμα Android και θα επωφελείται από το publish/subscribe framework το οποίο θα τρέχει ανεξάρτητα. Η εφαρμογή android θα εκτελείται ως εξής:

1. Η βασική οθόνη της εφαρμογής αυτής θα περιλαμβάνει μια οθόνη Google Maps. Πάνω σε αυτή την οθόνη ο χρήστης θα μπορεί να επιλέξει τις γραμμές του λεωφορείου για τις οποίες επιθυμεί πληροφορία.
2. Με το που γίνει αυτό, η εφαρμογή θα απευθύνεται στον κατάλληλο broker node ζητώντας την απαραίτητη πληροφορία αν υπάρχει, ειδάλλως θα περιμένει για κατάλληλο χρονικό διάστημα μέχρι αυτή να είναι διαθέσιμη.
3. Ακολουθείται η διαδικασία που περιγράψαμε στο κομμάτι της υλοποίησης της αποστολής και διαχείρισης των δεδομένων του publisher από τους brokers. Αν τυχόν δεν υπάρχει διαθέσιμη πληροφορία εκείνη τη στιγμή και έχει περάσει ένα εύλογο χρονικό διάστημα (κάποιων δευτερολέπτων), τότε επιστρέφεται κατάλληλο μήνυμα («ότι δεν βρέθηκε») σχετικά με την πληροφορία της συγκεκριμένης γραμμής στο κινητό παράλληλα όμως με την αποστολή πληροφορίας για τις υπόλοιπες λεωφορειακές γραμμές.
4. Με το που λάβει την πληροφορία από τον broker πρέπει να «ζωγραφίσει» κατάλληλα τις θέσεις των λεωφορείων πάνω στον χάρτη.

Bonus (15%): Στην περίπτωση που πέσει ένας broker, να μεταφερθούν κατάλληλα τα κλειδιά στους επόμενους brokers και να συνεχίσει να δουλεύει το σύστημα κανονικά.

Παραδοτέα εργασίας:

Το project θα παραδοθεί σε δύο φάσεις:

Παραδοτέο Α: (Ημερομηνία παράδοσης: 7/4/2019)

Στο παραδοτέο αυτό, θα πρέπει να έχετε ολοκληρώσει εντελώς το publish/subscribe σύστημα, όπως ακριβώς σας έχει ζητηθεί, έτσι ώστε να μπορεί να χρησιμοποιηθεί στην επόμενη φάση της εργασίας του μαθήματος.

Παραδοτέο Β: (Ημερομηνία παράδοσης: 24/5/2019)

Το παραδοτέο αυτό αποτελεί το android application, που περιγράφηκε πριν. Στη φάση αυτή το σύστημα θα πρέπει να είναι πλήρως λειτουργικό, με όλα τα components του να λειτουργούν σωστά.

Ομάδες: Όλοι οι φοιτητές θα πρέπει να σχηματίσουν ομάδες των τριών (3) ατόμων προκειμένου να εκπονήσουν την προγραμματιστική τους εργασία. Γλώσσα προγραμματισμού θα είναι η Java, στην οποία και θα παρέχεται υποστήριξη από τους βοηθούς του μαθήματος.

Αναφορές – Χρήσιμοι Σύνδεσμοι:

- [1] <https://docs.oracle.com/javase/8/docs/api/java/util/stream/package-summary.html>
- [2] Android. URL : <http://code.google.com/android/http://code.google.com/android/>
- [3] Android SDK: <http://developer.android.com/sdk/index.html>
- [4] Android Studio <http://developer.android.com/sdk/index.html>