

Πρόσθεσα στα προγράμματα μου το αντικείμενο Hospital που αντιπροσωπεύει ένα νοσοκομείο που βρίσκεται σε ένα τυχαίο κελί το οποίο έχει περιορισμένους πόρους για άτομα, και κρατά μια λίστα με τους ανθρώπους που φιλοξενεί(Array List). Επίσης, δημιούργησα τη κλάση HospitalCellNumber η οποία είναι βοηθητική κλάση που περιέχει το σημείο στο οποίο είναι το νοσοκομείο. Με ακριβώς την ίδια σκεπτική δημιούργησα τις κλάσεις QuarantineZone που αντιπροσωπεύει τη καραντίνα που βρίσκεται σε ένα συγκεκριμένο κελί και αποτελείται από το μέγεθος της και μια λίστα ανθρώπων ακριβώς όπως το hospital, και τη QuarantineCellNumber που πάλι λειτουργεί μόνο για να έχει το σημείο που είναι η καραντίνα. Επιπρόσθετα, δημιούργησα τη κλάση MMM(μέσα μαζικής μεταφοράς) τα οποία αποτελούνται από το μέγιστο αριθμό ατόμων που μπορούν να δεχτούν, δύο ακέραιους για του που ξεκινά και τελειώνει η διαδρομή του( εδώ χρησιμοποιώ το πίνακα με τις χώρες και οι χώρες αντιπροσωπεύονται από τον δείκτη τους στο πίνακα), και μια λίστα με ανθρώπου και πόσες φορές θα εκτελεί τη διαδρομή κάθε μέρα. Πρόσθεσα ακόμα την κλάση patient που δείχνει πιο αναλυτικά στοιχεία κάποιο άνθρωπο που ασθενεί, δηλαδή αποτελείται από την μεταβλητή του αν είναι άρρωστος, πόσες μέρες είναι άνθρωπος, αν πήγε στο νοσοκομείο ποια χρονική στιγμή πήγε (λεπτά) και την emfanish δηλαδή αν παρουσίασε στοιχεία από τα οποία φαίνεται πως είναι άρρωστος και έχω μια συνάρτηση setEmfanish στη οποία αν είναι άρρωστος για πάνω από 5 μέρες τότε αυτόματα φαίνεται η αρρώστια, αλλιώς λειτουργώ τυχαία για να δείξω αν φαίνεται.

```
public void setEmfanish() {
    if(days>5) {
        emfanish=true;
        return;}
    int a=(int) (Math.random()*4) +2; //mporei nane 2,3,4,5 meres
    if(a<=days) //an exun perasei o tixeos arithmos gia tis meres emfanishs
        tus iu tote emfanizete
        emfanish=true;
}
```

Το αντικείμενο Patient έγινε πεδίο του ανθρώπου, και κάθε κελί έχει ζώνη καραντίνας και νοσοκομείο αλλά ουσιαστικά ένα κελί είναι καραντίνα ζωνη κα ένα άλλο νοσοκομείο. Πρόσθεσα στο field τα πεδία

QuarantineCellNumber, HospitalCellNumber, με τη βοήθεια των οποίων μπόρω να βρω ποιο κέλι είναι ζώνη καραντίνας ή νοσοκομείο χωρίς να ψάχνω στις χώρες δίδοντας έτσι μεγαλύτερη αποδοτικότητα στο κώδικα.

Και για τη διαδικασία της καραντίνας και του νοσοκομείου δουλεύω με παρόμοιο τρόπο, δηλαδή έχω συνάρτηση η οποία δημιουργεί τη καραντίνα και νοσοκομείο ψευδοτυχαία(setQuarantine, makeHospital) και έχω μέθοδο για τη πρόσθεση ατόμων στη καραντίνα/ νοσοκομείο όπως

```
public boolean addToHospital(Human h) {
    int x=hospitalCell.getPoint().getX();
    int y=hospitalCell.getPoint().getY();

    if(board[y][x].getHospital().getNosuntes()<board[y][x].getHospital().getCapacity()) {
        board[y][x].getHospital().addPerson(h);
    }
```

```

board[y][x].getHuman().getPatient().initializeHospitalTime(ExtendedField.getRound());
    return true;
}
return false;

```

Για τη каранτίνα έχω τη συνάρτηση `checkForInfection` που αν τηρούνται οι προϋποθέσεις θα στείλουμε κάποιο каранτίνα και κάτι αντίστοιχο η `checkForHospital` για τα νοσοκομεία. Για τα νοσοκομεία έχω επίσης τη `hospitalExit` στη οποία κάποιος βγαίνει από το νοσοκομείο αν γίνει καλά (σε 7 μέρες με βάση τη βιβλιογραφία που διάβασα) και αποκτά ανοσία.

Πρόσθεσα για τα MMM τη μέθοδο `mesoMove` στη `Field` η οποία συλλέγει ανθρώπους αν οι ίδιοι θέλουν (`Math.random() >= 0.50`) και τους βάζει στο MMM που θα τους μεταφέρει σε άλλη χώρα. Επίσης πρόσθεσα τη `addManyPeople` η οποία κάνει `spawn` βάζει τους ανθρώπους σε τυχαίες θέσεις σε μια χώρα, έτσι ώστε να τη χρησιμοποιήσω για τη μεταφορών μέσω των MMM αφού πηγαίνει το MMM στο προορισμό και εμφανίζει σε τυχαίες θέσεις τους ανθρώπους του. Στη `initialize` πρόσθεσα ποσοστά για το ηλικιακό γκρούπ και το φύλο του ανθρώπου

```

position_x = (int) (Math.random() * getLengthP());
position_y = (int) (Math.random() * getLengthM());
if(counter <= 16 * numOfHumans)
    a = 'A';
else if(counter <= 38 * numOfHumans)
    a = 'B';
else if(counter <= 60 * numOfHumans)
    a = 'C';
else if(counter <= 84 * numOfHumans)
    a = 'D';
else
    a = 'E';
if(counter <= 51 * numOfHumans)
    g = 'W';
else
    g = 'M';

```

Αυτό είναι το αντίστοιχο υπόμνημα για τις ηλικιακές ομάδες

```

age: 0- 14 :A
*    15-29 B
*    30-44 C
*    45-64 D
*    65 +  E

```

Τέλος στη `field` άλλαξα τη `printStatistics` για να βγάζει και για τα άτομα που είναι νοσοκομείο/καραντίνα. Επίσης όσο αφορά τα σύνολα δουλεύω όπως στη προηγούμενη εργασία που έχουμε τα ίδιο αριθμό συνοριακών κελιών μεταξύ δύο περιοχών.

Στη human πρόσθεσα τη ηλικιακή ομάδα και το φύλο και το preniogw που είναι το προηγούμενο κελί που ήταν κάποιος άνθρωπος έτσι ώστε εκεί που χρειάζεται να ελεγχξουμε πως κάποιος ήταν δύο συνεχόμενα round στο ίδιο κελί να το χρησιμοποιήσουμε. Η κλάση αυτή έχει περισσότερο setter getter μεθόδους.

Στη μέθοδο TakeHomeSimulation δημιουργώ τη προσομοίωση που μας ζητείται από την άσκηση και την καλώ στη main της TakeHomeSimulation και τρέχω εκεί τη προσομοίωση για συγκεκριμένο χρονικό διάστημα σε ΛΕΠΤΑ.

Το c είναι οι μέρες της προσομοίωσης

```
int k=0;
    System.out.println("Arxika statistika");
    f.generalStatistics();
    for(int i=0;i<c*48;i++) {
        System.out.println();
        System.gc();
        System.out.println("Round "+f.getRound());
        System.out.println();
        f.nextGeneration();
        System.out.println();

        k++;
    }
```

## Αναλλοίωτοι περιορισμοί:

1. Δυο παίκτες δεν μπορούν να είναι στο ίδιο κελί ταυτόχρονα δηλαδή κάθε κελί μπορεί να φιλοξενεί ένα άνθρωπο τη φορά εκτός και αν είναι νοσοκομείο ή καραντίνα.
2. Κάποιος άνθρωπος δεν μπορεί να είναι εκτός τους χωρού δηλαδή έξω από τον δυσδιάστατο πίνακα αν δεν ήταν σε σύνορο αφού αν είναι σε σύνορο θα τον μεταφέρει στη χώρα που είναι το σύνορο.
3. Ο άνθρωπος που έχει ανοσία δεν μπορεί να αρρωστήσει.
4. Ο χώρος που διαδραματίζονται οι προσομειώσεις είναι ορθογώνιος μήκος και πλάτος.
5. Ο χρόνος προσομειώσης που δίνει ο χρήστης είναι σε ώρες και και κάθε βήμα-κίνηση δηλαδή round αντιστοιχεί σε 30 λεπτά.
6. Αριθμός των ατόμων πρέπει να είναι μικρότερο από το εμβαδό
7. Όταν μολυνθεί κάποιο κελί όσο περνά ο χρόνος μειώνεται ο ποσοστό μόλυνσης του κελιού
8. Όλες οι κινήσεις γίνονται ταυτόχρονα και άρα δεν μπορεί κάποιος να μπει στο κελί που ήμουν αφού κινούμαστε ταυτόχρονα και ο άλλος βλέπει πως είναι γεμάτο το κελί
9. Ο κάθε άνθρωπος μπορεί να κινηθεί ένα κελί τη φορά εκτός και αν φύγει από κάποιο σύνορο και πάει σε άλλη χώρα όπου κάνει άλλη μία επιπρόσθετη κίνηση για να φύγει από το σύνορο και δεν πηγαίνει σε άλλο σύνορο
10. Ο άνθρωπος μπορεί να αρρωστησει από άνθρωπο που είναι κοντά του κατά ένα κελί δηλαδή δεξιά,αριστερά,πάνω,κάτω,πάνω δεξιά,πάνω αριστερά,κάτω δεξιά,κάτω αριστερά
11. Στο αρχικό καθορισμό των θέσεων των ανθρώπων όλοι έχουν τη ίδια πιθανότητα για να βρεθούν σε ένα ελεύθερο τετράγωνο

12. Ο κάθε άνθρωπος έχει 50% πιθανότητα να κινηθεί κάθε γύρο ,εχτός και αν είναι σε σύνορο που έχει 100% πιθανότητα κίνησης και εχτός και αν δεν μπορεί να πραγματοποιηθεί κίνηση- να είναι εγκλωβισμένο.
13. Όλες οι κατευθύνσεις κίνησης έχουν τη ίδια πιθανότητα επιλογής εκτός και αν ο άνθρωπος είναι σε σύνορο.
14. Αν κάποιος φύγει από κάποιο σύνορο μπορεί να πάει στο σύνορο κάποιας άλλης χώρας και θα μετακινηθεί σε κάποιο κελί που ΔΕΝ είναι σύνορο και αν παραμείνει στη χώρα του τότε δεν πρέπει να μετακινηθεί σε άλλο σύνορο.
15. Το κάθε κελί σύνορο είναι συνδεδεμένο με κάποιο συγκεκριμένο κελί και αν είμαι σε αυτό το κελί και φύγω έξω από τη χώρα θα πάω στο συνδεδεμένο κελί-σύνορο της άλλης χώρας και είναι αμφίδρομη η σχέση δηλαδή ισχύει  $\alpha \rightarrow \beta$  ΚΑΙ  $\beta \rightarrow \alpha$ .
16. Όταν έχω σύνορο με κάποια χώρα αυτό καταλαμβάνει η συνεχόμενα κελιά, που πρέπει να είναι στο περίγραμμα του πίνακα και να είναι κάθετα ή οριζόντια.
17. Όταν φεύγω από μια χώρα και πάω σε άλλη τότε κάνω ακόμα μια κίνηση σε ελεύθερο κελί που δεν είναι σύνορο.
18. Αν υπάρχει άνθρωπος στο σύνορο μιας χώρας και κινηθεί έξω από αυτή άρα πρέπει να μεταφερθεί στη άλλη χώρα, και στη άλλη χώρα στο κελί-σύνορο που θα μεταφερθεί υπάρχει άλλος άνθρωπος τότε τους αλλάζω και τους δύο ανθρώπους χώρα και δυστυχώς ο δεύτερος άνθρωπος χάνει τη επιλογή του να μείνει στη χώρα του.
19. Αν ο άνθρωπος είναι σε κάποιο σύνορο και βγει έξω από τη χώρα του αλλά στη άλλη χώρα το σύνορο που θα μεταφερθεί είναι περικυκλωμένο από ανθρώπους δηλαδή δεν θα μπορεί να φύγει από εκεί θα μείνει στο σύνορο του χωρίς να μεταφερθεί και το επόμενο round αν θέλει να φύγει από τη χώρα θα το ξανα δοκιμάσει.
20. Δεν μπορεί μια χώρα να δημιουργήσει σύνορο με χώρα που δεν υπάρχει.
21. Δεν μπορεί μια χώρα να δημιουργήσει σύνορο με τον εαυτό της.
22. Δεν μπορεί μια χώρα να δημιουργήσει σύνορο με χώρα με την οποία έχει ήδη σύνορο.
23. Δεν μπορεί μια χώρα να δημιουργήσει σύνορο με χώρα στην οποία δεν μπορεί να δημιουργηθεί σύνορο στο περίγραμμα της δηλαδή να μην έχει η συνεχόμενα ελεύθερα κελιά στο περίγραμμα της.
24. Η μεταφορά στο νοσοκομείο και στο νοσηλευτήριο γίνεται απευθείας δηλαδή δεν έχει διάρκεια.
25. Το νοσοκομείο και η καραντίνα είναι ένα κελί που παράγεται τυχαία.
26. Στη αρχή δεν υπάρχουν άνθρωποι με ανοσία
27. Αν κάποιος άνθρωπος πάει νοσοκομείο και περάσει μια εβδομάδα τότε γίνεται υγιής και έχει ανοσία στον ιό.
28. Η κίνηση των MMM γίνεται απευθείας δηλαδή όταν έχει σειρά κάποιο MMM να μεταφερθεί τότε μαζεύει τους ανθρώπους, τους μεταφέρει στο προσορισμό όπου τους βάζει σε μια θέση τον καθένα τυχαία.
29. Κανείς δεν φεύγει από την καραντίνα.
30. Αν γεμίσει η καραντίνα, το νοσοκομείο, το MMM τότε δεν δέχεται άλλα άτομα.
31. Κάποιος πρέπει να εμφανίσει συμπτώματα για να πάει στο νοσοκομείο.
32. Κάνουμε το έλεγχο για κορονοϊό κάθε 24 ώρες.
33. Αν γίνει έλεγχος σε κάποιον βγει θετικός και είναι άρρωστος βαριά ή είναι μεγάλης ηλικίας(>65) τότε πηγαίνει στο νοσοκομείο αντί καραντίνα.
34. Στέλνω κάποιον στο νοσοκομείο με βάση του αν εμφανίστηκε η αρρώστια και είναι βαριά άρρωστο, πρόσθετα όμως πς αν κάποιος έχει πάνω από 0.5 πιθανότητα βαριάς ασθένειας(Δηλαδή είναι ηλικιωμένος >65) τότε τον στέλνω απευθείας στο νοσοκομείο γίνεται και στη πραγματικότητα.

**Χρησιμοποίησα το τα ArrayList<>.**

Αυτές είναι οι έρευνες που διάβασα και βασίστηκα για να βάλω στις διάφορες ηλικιακές ομάδες και στα φύλα τους ανθρώπους και πόσο είναι το ποσοστό για να ασθενήσουν βαριά αυτά(Βασίστηκα στα δεδομένα του 2016)

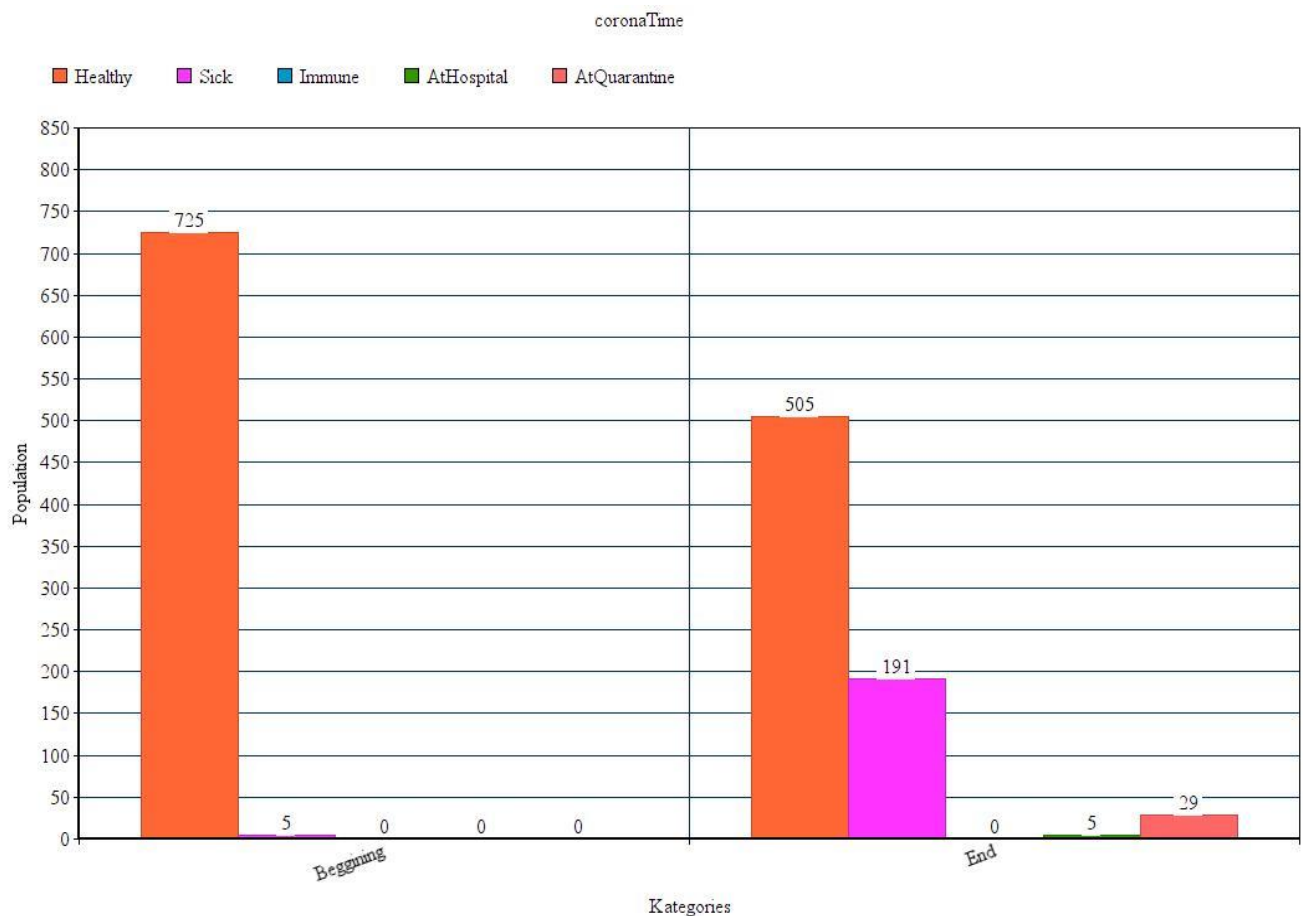
<https://www.weforum.org/agenda/2020/03/coronavirus-young-people-hospitalized-covid-19-chart/>

<https://eody.gov.gr/erotiseis-kai-apantiseis-gia-ton-neo-koronoio-sars-cov-2-gia-tis-ekpaideytikes-monades/>

## Z3 Επίμετρο 1

Από τις προσομοιώσεις μου σε σχέση με το και με το Project5 συμπέρανα πως τα μέσα μεταφορά και το να μη λαμβάνουν προστασία βοηθά περισσότερο στην εξάπλωση του ιού. Επίσης κατανόησα τη σημασία που έχει το να σταματήσουν να κινούνται τα MMM, η κοινωνική αποστασιοποίηση και πως με το που γεμίσουν τα νοσοκομεία τότε η κατάσταση βγαίνει εκτός ελέγχου. Άρα τελικό συμπέρασμα να μένουμε σπίτι για να καταπολεμήσουμε τον ιο και να ευχομαστε να μην γεμίσουν τα νοσοκομεία και οι ζώνες καραντίνας. Επίσης όσα περισσότερα test κάνουμε τόσο το καλύτερο- τόσο πιο πολλούς ασυμπτωματικούς φορείς θα βρίσκουμε.

Δημιούργησα μια γραφική που δείχνει τα αποτελέσματα πριν και μετά τη προσομοίωση μου για τους υγιείς, άρρωστους, με ανοσία, στο νοσοκομείο, σε καραντίνα ανθρώπους με δεδομένα από το simulation του προγράμματος μου.



## Z4 Επίμετρο 2

```
/**
 * Avth einai h mehtodos initialize ,arxikopoiei to simulation dhladh dinei to
 * enavsma kai kapoia arxika stoixeia gia na xekinisei.
 *
 * @param numOfHumans arithmos twn anthrwpwn
 * @param infectionP pososto molinshs
 * @param c to capacity tu nosokomeiu
 * @param capto capacity ths zwnhs karantinas
 * @param s to onoma ths xwras
 *
 */
public void initialize(int numOfHumans, double infectionP, int c,int cap,String
s) {

    name = s;
    count++;
    infectionPercentage=infectionP;
    int counter=0, position_x, position_y;
    char g,a;//gender kai age group
    position_x = (int) (Math.random() * getLengthP());
    position_y = (int) (Math.random() * getLengthM());
    if(counter<=16* numOfHumans)
        a='A';
    else if(counter<=38 * numOfHumans)
        a='B';
    else if(counter<=60 * numOfHumans)
        a='C';
    else if(counter<=84 * numOfHumans)
        a='D';
    else
        a='E';
    if(counter<= 51 *numOfHumans)
        g='W';
    else
        g='M';

    Human h = new Human("h"+counter, false , true,a,g);

    while(!board[position_x][position_y].setHuman(h)) {
        position_x =(int) Math.random() * getLengthM();
        position_y =(int) Math.random() * getLengthP();
    }

    counter++;

    for(int i=1; i < numOfHumans; i++) {

        position_x =(int) (Math.random() * getLengthP());
        position_y =(int) (Math.random() * getLengthM());
        if(counter<=16)
```

```

        a='A';
    else if(counter<=38 * numOfHumans)
        a='B';
    else if(counter<=60 * numOfHumans)
        a='C';
    else if(counter<=84 * numOfHumans)
        a='D';
    else
        a='E';
    if(counter<= 51 *numOfHumans)
        g='W';
    else
        g='M';

    Human h2 = new Human("h"+counter, false , false,a,g);

    while(! board[position_x][position_y].setHuman(h2)) {
        position_x =(int) Math.random() * getLengthM();
        position_y =(int) Math.random() * getLengthP();
    }
    counter++;
}

```

Σε αυτή τη συνάρτηση πρώτα δημιουργώ τον άνθρωπο που έχει μολυνθεί με τον ιό και μετά όλους τους υγιείς ανθρώπους. Λαμβάνω υπόψη μου τις έρευνες που διάβασα για τα ποσοστά των διαφόρων ηλικιακών γκρουπ. Και διαφοροποιώ τις διάφορες ηλικιακές ομάδες με το χαρακτήρα που έχουν στο πεδίο gender ,age.

Όσο αφορά τις ηλικίες και το φύλο των ατόμων έλαβα υπόψη μου τα δεδομένα από τη σελίδα που μας δώσατε για το 2016 και κατέληξα στα εξής ποσοστά

0-14: A : 16%

15-29: B : 22%

30-44 : C : 22%

45-64: D : 24%

65+ : E : 16%

Άντρες : M : 49%

Γυναίκες: W : 51%



## Z5 Επίμετρο 3

```
/**
 * Η sinarthsh pou ektelei tus deigmatoliptikus elegxus gia na duume an einai
kapoios arrwstos tote ton stelume sth karantina.
 * Thewrw pws kapoios den mporei na figei apo thn karantina.
 *
 * @param a posus elegxus tha kanume
 */

public void checkforInfected(int a) {
    int counter=0;
    for(int i=0;i<getLengthP();i++) {
        for(int j=0;j<getLengthM()&& counter<a;j++) {
            if(board[i][j].getHuman()!=null) {
                if( board[i][j].getHuman().getSick() &&
Math.random()>=0.2) {
                    double chance=0;
                    double m=Math.random();
                    switch(board[i][j].getHuman().getAge()) {
                        case 'A':
                        case 'B':
                            chance=0.05;
                            break;
                        case 'C':
                            chance=0.208;
                            break;
                        case 'D':
                            chance=0.301;
                            break;
                        case 'E':
                            chance= 0.60;
                            break;
                    }
                    if(m<=chance || chance>0.5) {
//An einai megalhs hlikias o anthrwpos tote paei apevtheias sto nosokomeio
                        if(addToHospital(board[i][j].getHuman())) {
                            board[i][j].removeHuman();
                        }
                    }
                    else if(addToQuarantine(board[i][j].getHuman()))
//an o anthrwpos einai arrwstos tote paei sth karantina kai afaireitai apo to board
                        board[i][j].removeHuman();
                    }
                    counter++;
                }
            }
        }
    }
}
```

```
}
```

Σε αυτή τη ,μέθοδο διατρέχω τη χώρα και πραγματοποιώ ελέγχους για τον ιο αυτή συνάρτηση τελειώνει είτε όταν δεν έχω άλλα άτομα να ελέγξω είτε όταν χρησιμοποιήσω όλα τα τεστ για τον ιο(που τα δέχομαι σαν παράμετρο).Επίσης αν βρώ κάποιον βαριά άρρωστο ή που να έχει τον ιο και να είναι >65 τότε τον στέλνω στο νοσοκομείο.

```
/**  
 * Sinarthsh pu ektelei to test tu COVID-19 se oles tis xwres kata to οποιο an  
 kapoios vgei thetikos metaferete se karantina.  
 *  
 * @param a  
 */  
public void coronaCheck(int a) {  
    for(int k=0;k<list.size();k++)  
        list.get(k).checkforInfected(a);  
}
```

Σε αυτή τη μέθοδο διατρέχω την λίστα με τις χώρες μου και κάνω τον έλεγχο για τον ιό, σε όλες τις χώρες πραγματοποιώ ίδιο αριθμό εργαστηριακών ελέγχων.

## Z6 Επίμετρο 4

Ο πιο κάτω κώδικας είναι το αντικείμενο που αντιπροσωπεύει τα MMM δηλαδή είναι τα μέσα μαζικής μεταφοράς που αποτελούνται από το μέγεθος, που ξεκινούν, που πάνε, έχουν λίστα με τους ανθρώπους και πόσες φορές τη μέρα γίνεται η μεταφορά.

```
package ChristodoulouPanagiotis.tk;
```

```
/**
 * Object pu ekprosswpei ta mesa amzikhs anaforas.
 *
 * Apoteleitai apo to megethos tu mmm, apo poia xwra einia, se poia xwra paei, mia lista
 me tus anthrwpus pu exei, poses fores taxidevei mesa sth mera.
 */
import java.util.ArrayList;

public class MMM {
    private int capacity;
    private int from;
    private int to;
    private ArrayList<Human> peopleList;
    private int times;

    public MMM(int a, int b, int c, int t) {
        capacity=a;
        from=b;
        to=c;
        times=t;
        peopleList=new ArrayList(capacity); // arxika den exei anthrwpus to meso
        mazikhs metaforas
    }
    /**
     * Epistrefei to arithmo atomwn pou forei to mmm.
     *
     * @return
     */
    public int getCapacity() {
        return capacity;
    }
    /**
     * Epistrefei tis fores pou taxidevei sth mera.
     *
     * @return
     */
    public int getTimes() {
        return times;
    }
}
```

```

    * epistrefei th xwra apo th opoia xekina.
    *
    * @return
    */
    public int getFrom() {
        return from;
    }
    /**
    * epistrefei th xwra pou paei.
    *
    * @return
    */
    public int getTo() {
        return to;
    }
    /**
    * prostheti kapoio atomo sto mmm.
    *
    * @param h
    */
    public void addPeopleList(Human h) {
        peopleList.add(h);
    }
    /**
    * Epistrefei to arithmo atomwn pou metaferei.
    *
    * @return
    */
    public int getMetaferwmenus() {

        return peopleList.size();
    }
    /**
    * epistrefei ena arrayList me ta atoma pou metaferei.
    *
    * @return
    */
    public ArrayList<Human> getPeople() {
        return peopleList;
    }
    /**
    * Sinarthsh pou katharizei/ exafanizei olus tus anthrwpus apo to MMM.
    *
    */
    public void removeAll() {
        peopleList=new ArrayList<Human>(capacity);
    }

```

Αυτή(η πιο κάτω) η συνάρτηση εκτελεί την κίνηση σε μια χώρα για να μαζέψει όλους τους ανθρώπους που θέλουν να χρησιμοποιήσουν το MMM και τους προσθέτει στη λίστα(arrayList) του MMM και τους αφαιρεί(remove) από τη θέση που ήταν.

```
/**
 * Sinarthsh gi na ektelw to mazema/sillogh tw n anthrwpwn apo ta diafora kelia
gia na tus parw sth alh xwra me to meso metaforas.
 *
 * @meso pairnei san parametro to meso me to opoio tha tus parei
 */
public void mesoMove(MMM meso) {

    for(int j=0;j<this.getLengthM();j++) {
        for(int i=0;i<this.getLengthP();i++) {

            if(board[i][j].getHuman()!=null && meso.getMetaferwmenus(<meso.getCapacity()
&& Math.random())>=0.5) {

                meso.addPeopleList(board[i][j].getHuman());//tus prosthetw sth lista
                board[i][j].removeHuman();

            }

        }

    }

}
```

Σε αυτή τη συνάρτηση μαζεύω τους ανθρώπους από μια χώρα( με τη mesoMove) και τους προσθέτω/εμφανίζω σε τυχαίες χώρες στο προορισμό του MMM(με τη addManyPeople)

```
/**
 * Einai h methodos pu kalw gia na etelesw th kinsh enos sigkekrimenu Mesu
mazikh s anforas.
 * Ektelw kinsh apo A-->B kai xana erxete to metaforiko meso piso sto A adeio
epeidh an etelesw xana kinsh apo B-->A
 * den tha einai ebdiakrito pws kinithikan ta atoma.
 *
 * @param m1 to metaforiko meso
 */
public void motionbyMMM(MMM m1) {
    list.get(m1.getFrom()).mesoMove(m1);
    list.get(m1.getTo()).addManyPeople(m1.getPeople());
}
```

```
}
```

Αυτή η μέθοδος παίρνει σαν παράμετρο ένα `arrayList` με ανθρώπους και εμφανίζει τους ανθρώπους σε τυχαία σημεία στη χώρα οποία σημεία μπορεί να πάει ο άνθρωπος δηλαδή δεν είναι συνορα,καραντίνα,δεν έχει άλλο εκεί.

```
/**
 * Sinarthsh h opoia dexetai ena arrayList apo anthrwpus kai tus emfanizei tixea
 mesa sth xwra.
 *
 * @param listH to arrayList me tus anthrwpus
 */
public void addManyPeople(ArrayList<Human> listH) {
    int k=0;
    int megethos=listH.size();
    while(k<megethos)
    {
        boolean f=true;
        int x,y;
        do {
            x=(int) (Math.random()*this.getLengthM());
            y=(int) (Math.random()*this.getLengthP());

            f=checkEmpty(x,y) || board[y][x].getSinoro()!=null;

        }while(!f);
        board[y][x].setHuman(listH.get(k));
        k++;
    }
}
```

Ο πιο κάτω κώδικας αποφασίζει αν τώρα είναι η στιγμή για να εκτελέσει τη κίνηση του το MMM (δηλαδή να συλλέξει τους ανθρώπους και να τους εμφανίσει στη χώρα)

```
/**
 * Η sinarthsh pu apofasizei an sth parusa xronikh stigmh prepei na ginei h
metafora tu MMM
 * prwta kanw to xrono pume twra na ginei o antistoixos sto prwto 24wro(dhladh
apo 1-1440 perilamvanwmenu
 * meta diatrexw sth lista me ta MMM vlepw an eimai sto xrono pu prepei na
ektelestei h kinsh kai h metafora tu MMM tote ta ektelw.
 *
 */
public void kinshMesaMetaforas() {
    int t=this.time;
    while(t>1440)
        t=t-1440;

    for(int k=0;k<this.mesa.size();k++) {
        int s=1440/mesa.get(k).getTimes();
        if(t%s==0) {//gia nadw an einai h seira tu mmm na kanei to
dromologio tu.

                motionbyMMM(mesa.get(k));
                mesa.get(k).removeAll();
            }}
}
```