

# PROJECT ΣΤΗΝ JAVA - 1η ΦΑΣΗ

## Προσομοίωση ενός Οικοσυστήματος



Project Εαρινού εξαμήνου 2002 στο μάθημα HY252

### Φάση 1η – Σχεδιασμός

Ομάδα Εργασίας:

- Γεωργόπουλος Γιώργος A.M: 1340 [ggeorgop@csd.uoc.gr](mailto:ggeorgop@csd.uoc.gr)
- Κριτσωτάκης Μανώλης A.M: 1395 [kritsot@csd.uoc.gr](mailto:kritsot@csd.uoc.gr)
- Παπαδογιαννάκης Αντώνης A.M: 1415 [papadog@csd.uoc.gr](mailto:papadog@csd.uoc.gr)

# ΑΝΑΦΟΡΑ 1<sup>ης</sup> ΦΑΣΗΣ

## 1. Περιγραφή της Εργασίας

Το EcoSystem Simulator είναι ένα πρόγραμμα το οποίο προσομοιώνει την εξέλιξη ενός οικοσυστήματος μέσα στο οποίο ζουν και αναπτύσσονται ταυτόχρονα διάφορα ήδη οργανισμών, σαρκοφάγων, φυτοφάγων και φυτών. Συγκεκριμένα, τα ζώα που υπάρχουν μέσα στο σύστημα είναι λιοντάρια, ζέβρες, τίγρεις, ελέφαντες, αρκούδες, καμηλοπαρδάλεις και από φυτά, δέντρα και θάμνοι.

Οι λειτουργίες που παρέχονται στο χρήστη είναι περιορισμένες καθώς αυτός το μόνο που κάνει είναι να παρακολουθεί την εξέλιξη του οικοσυστήματος. Έτσι, πριν και κατά την διάρκεια της προσομοίωσης, ο χρήστης επιλέγει τους οργανισμούς που θα υπάρχουν στο οικοσύστημα. Κατά τη διάρκεια της προσομοίωσης θα υπάρχουν οι εξής λειτουργίες:

- Λειτουργίες Προσομοίωσης οι οποίες περιλαμβάνουν την έναρξη (start), το πάγωμα (pause), την συνέχιση (resume) και την επανεκκίνηση (restart) της προσομοίωσης.
- Λειτουργίες Πληροφοριών με τις οποίες δίνεται στον χρήστη η δυνατότητα να βλέπει πληροφορίες σχετικά με ένα πληθυσμό οργανισμών ή και σχετικά με ένα οργανισμό συγκεκριμένα.

Το περιβάλλον της εφαρμογής θα αποτελείται από 4 κυρίως μέρη. Το πρώτο και βασικό είναι ο χάρτης πάνω στον οποίο εξελίσσονται όλοι οι οργανισμοί. Η εξέλιξη ενός οργανισμού προέρχεται από την περιπλάνησή του στο χάρτη με τυχαίο τρόπο, την αλληλεπίδραση του με άλλους οργανισμούς και τον θάνατό του. Εκτός από το χάρτη, το περιβάλλον περιλαμβάνει μία μπάρα λειτουργιών στην οποία θα βρίσκονται τα κουμπιά που θα ελέγχουν τις λειτουργίες που προαναφέραμε καθώς και την μπάρα αντικειμένων η οποία θα περιέχει όλους τους οργανισμούς και από την οποία με την διαδικασία του drag and drop, θα τοποθετούνται στο χάρτη. Τέλος, στο κάτω μέρος θα βρίσκονται περιοχές όπου θα εμφανίζονται οι απαραίτητες πληροφορίες.

## **2. Περιγραφή του Περιβάλλοντος**

Η εφαρμογή θα αποτελείται από ένα παράθυρο το οποίο θα περιέχει τον χάρτη πάνω στον οποίο θα περιπλανιούνται τα ζώα του οικοσυστήματος, καθώς και τα κουμπιά και τα πεδία με πληροφορίες.

Στον χάρτη, τα ζώα περιπλανιούνται ελεύθερα με το πέρασμα των εβδομάδων χωρίς να πέφτει το ένα πάνω στο άλλο. Η ελάχιστη απόσταση μεταξύ τους είναι καθορισμένη ώστε ούτε να πέφτει το ένα πάνω στο άλλο αλλά και να υπάρχει η δυνατότητα να έρχονται τόσο κοντά ώστε να τρέφονται.

Στα αριστερά του χάρτη βρίσκεται η μπάρα αντικειμένων η οποία παρουσιάζει μια λίστα με όλους τους διαθέσιμους οργανισμούς από όπου ο χρήστης επιλέγει όποιον επιθυμεί για προσθήκη στο χάρτη και στο οικοσύστημα. Οι οργανισμοί θα εμφανίζονται σε συμβολικές φωτογραφίες και καθώς δεν χωράνε όλοι σε μια οθόνη της μπάρας αντικειμένων, θα υπάρχει η δυνατότητα κύλισης.

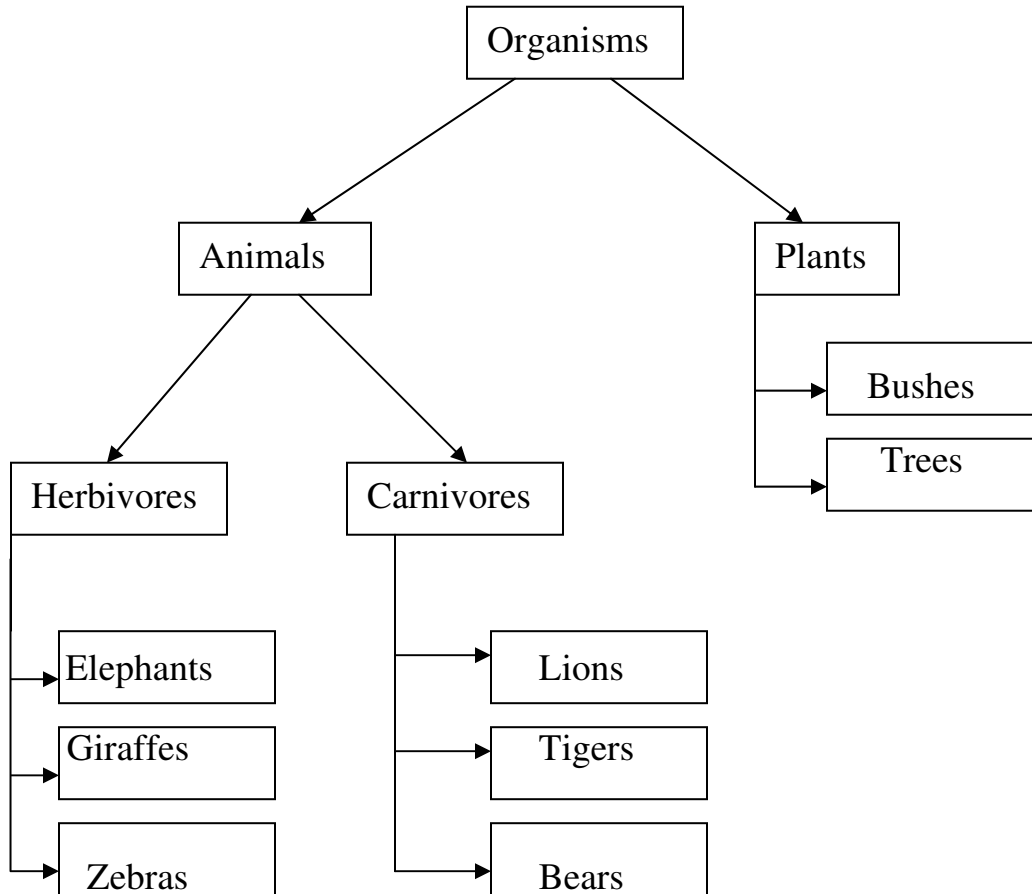
Στην κορυφή του παραθύρου θα βρίσκονται τα κουμπιά ελέγχου της προσομοίωσης. Συγκεκριμένα θα υπάρχουν τα κουμπιά start, pause, restart, resume.

Τέλος, στο κάτω μέρος του παραθύρου θα βρίσκονται δύο πεδία στα οποία θα παρουσιάζονται στο ένα οι πληροφορίες για κάθε άτομο ξεχωριστά και στο άλλο στατιστικές πληροφορίες για όλα τα όμοια ήδη, δηλαδή πληροφορίες που αφορούν το μέσο όρο μεγέθους των οργανισμών της κατηγορίας που διαλέγουμε, το πλήθος των οργανισμών που ζουν αυτή την στιγμή, την μέση κατανάλωσή τους κατά την τελευταία εβδομάδα, ποσοστά θανάτων των οργανισμών σε σχέση με τους συνολικούς θανάτους κατά την διάρκεια της προσομοίωσης, στατιστικά των ηλικιών ανά οργανισμό.

## **3. Περιγραφή των οργανισμών**

Το οικοσύστημα που προσομοιώνουμε αποτελείται από διάφορα είδη οργανισμών. Όλοι οι οργανισμοί έχουν τέσσερα κύρια χαρακτηριστικά: την ηλικία τους, τον ρυθμό ανάπτυξής τους, την ενέργειά τους -μέγεθος- και την θέση στην οποία βρίσκονται. Η ηλικία αρχικά είναι μηδέν και μετριέται σε εβδομάδες. Ο ρυθμό ανάπτυξης είναι σταθερός και διαφορετικός για κάθε είδος οργανισμού. Η ενέργεια κάθε οργανισμού αρχικά είναι ένας αριθμός μεγαλύτερος του μηδέν που διαφέρει ανάλογα με το είδος οργανισμού και κατά την διάρκεια της προσομοίωσης αυτή μεταβάλλεται ανάλογα με τον συγκεκριμένο οργανισμό. Αυξάνεται και μειώνεται όσο είναι ο ρυθμός

ανάπτυξης του οργανισμού. Αν κάποια στιγμή η ενέργεια ενός οργανισμού γίνει μικρότερη ή ίση με το μηδέν τότε ο οργανισμός αυτός πεθαίνει. Τέλος η θέση κάθε οργανισμού μας δείχνει σε ποιο μέρος του χάρτη βρίσκεται. Οι οργανισμοί αποτελούνται από δύο κύριες κατηγορίες: Τα φυτά και τα ζώα. Τα φυτά τρέφονται κάθε εβδομάδα αυτόματα με ήλιο και νερό που υπάρχουν άφθονα και έτσι αυξάνεται κάθε εβδομάδα η ενέργειά τους κατά τον ρυθμό ανάπτυξης του κάθε φυτού. Αν ένα χορτοφάγο ζώο όμως φάει από ένα φυτό, η ενέργεια αυτού του φυτού μειώνεται κατά τον ρυθμό ανάπτυξής του. Τα ζώα κινούνται μέσα στο οικοσύστημα και ψάχνουν για τροφή. Πρέπει να φάνε ένα ορισμένο ποσό τροφής για να αυξήσουν την ενέργειά τους κατά τον ρυθμό ανάπτυξης τους κάθε εβδομάδα. Αλλιώς η ενέργειά τους μειώνεται κατά το ίδιο ποσό. Η κίνηση των ζώων γίνεται με τυχαίο τρόπο. Τα ζώα χωρίζονται σε χορτοφάγα και φυτοφάγα. Τα χορτοφάγα μπορούν να τραφούν μόνο από τα φυτά του οικοσυστήματος, τρώγοντας από αυτά τροφή ίση με τον ρυθμό ανάπτυξης του κάθε φυτού, όταν τα πλησιάσουν σε μικρή απόσταση. Τα σαρκοφάγα τρέφονται μόνο με άλλα ζώα, φυτοφάγα ή σαρκοφάγα, τα οποία έχουν το μισό ή λιγότερο μέγεθος από το δικό τους και τα έχουν πλησιάσει σε μικρή απόσταση. Η τροφή που παίρνουν από αυτά ισούται και πάλι με τον ρυθμό ανάπτυξης του θύματος, του οποίου η ενέργεια μειώνεται κατά τον ίδιο αριθμό. Τα φυτά που εμείς θα έχουμε διαθέσιμα για εισαγωγή στο οικοσύστημά μας είναι θάμνοι και δέντρα. Τα φυτοφάγα ζώα θα είναι ζέβρες, ελέφαντες και καμηλοπαρδάλεις. Τα σαρκοφάγα θα είναι λιοντάρια, τίγρεις και αρκούδες. Οι ρυθμοί ανάπτυξης, το φαγητό που πρέπει να τρώνε ανά εβδομάδα και η αρχική ενέργεια αυτών των ζώων θα καθοριστούν στην επόμενη φάση. Η ιεραρχία των οργανισμών του οικοσυστήματός μας όπως την περιγράψαμε φαίνεται παρακάτω.



#### 4. Ανάλυση των λειτουργιών

##### 4.1 Δημιουργία Νέου Οργανισμού

Ο χρήστης μέσω αυτής της λειτουργίας θα μπορεί να τοποθετεί πάνω στον χάρτη έναν νέο οργανισμό που αυτός θα επιλέγει. Τα εικονίδια των οργανισμών που θα μπορεί να δημιουργεί ο χρήστης θα φαίνονται στο αριστερό κομμάτι της οθόνης της προσομοίωσης και συγκεκριμένα στη Μπάρα Αντικειμένων. Εκεί θα υπάρχουν όλοι οι οργανισμοί που το πρόγραμμα μας προσομοιώνει.

Η επιλογή του Οργανισμού που θέλουμε να δημιουργήσουμε, θα γίνεται πατώντας πάνω στο εικονίδιο του επιθυμητού οργανισμού. Στη συνέχεια ο χρήστης θα πρέπει να επιλέγει τη θέση μέσα στο χάρτη όπου θέλει να τοποθετηθεί ο οργανισμός. Αυτό γίνεται πατώντας πάνω στο επιθυμητό μέρος του χάρτη όπου και στη συνέχεια εμφανίζεται ένα

νεο εικονίδιο που αφορά τον οργανισμό. Προυποθεση είναι ότι στη θέση που εισαγουμε το νεο οργανισμο δεν υπαρχει ηδη αλλος οργανισμός.

## **4.2 Λειτουργίες Προσομείωσης**

Μέσω αυτών των λειτουργιών ο χρήστης μπορεί να παρακολουθήσει την κίνηση των οργανισμών και την αλληλεπίδραση μεταξύ τους. Κάθε οργανισμός (εκτός των φυτών) σε καθημερινή βάση (χρόνο που έχουμε καθορίσει εμείς να αντιστοιχεί σε μια μέρα) θα τρωει τους οργανισμούς που βρίσκονται σε κοντινή απόσταση με αυτόν (απόσταση που έχει καθοριστεί πάλι από εμάς) και ανάλογα αν η ενέργεια του το επιτρέπει. Ο οργανισμός ο οποίος επιτίθεται θα αυξάνει την ενέργεια του. Αντίστοιχα ο οργανισμός ο οποίος θα αποτελεί την τροφή του πρώτου οργανισμού θα χάνει ενέργεια.

Επιπλέον όταν κάποιος οργανισμός θα έχει πλέον μηδενική ενέργεια, τότε θα πεθαίνει και το εικονίδιο που αντιστοιχεί στο συγκεκριμένο οργανισμό θα απομακρύνεται από το χάρτη. Σε εβδομαδιαία βάση πλέον ο κάθε οργανισμός θα ανακτά η θα χάνει ποσά ενεργείας που θα αναλογούν στον ρυθμό ανάπτυξης του κι επιπλέον θα αυξάνεται η ηλικία του κατά μια ποσοστιαία μονάδα.

### **4.2.1 Έναρξη (Start)**

Μέσω αυτής της λειτουργίας ο χρήστης θα μπορεί να αρχίζει την προσομοίωση. Ουσιαστικά όμως, για να αρχίσει η προσομοίωση, θα πρέπει πρώτα ο χρήστης να έχει δημιουργήσει οργανισμούς και να τους έχει εισαγει στο χάρτη. Διαφορετικά δε θα υπαρχει ορατο αποτελεσμα με την έναρξη της προσομείωσης. Η λειτουργία αυτή θα πραγματοποιείται μέσω της επιλογής 'Start' που βρίσκεται στη πάνω μεριά της οθόνης και συγκεκριμένα στη Μπάρα Λειτουργιών. Εκεί ο χρήστης θα πρέπει να παταει το αντιστοιχο κουμπι και η προσομείωση θα ξεκιναι. Ο χρήστης θα μπορεί να προσθετει νεους οργανισμους και μετα την έναρξη της προσομείωσης με τον ιδιο τροπο που περιγραφηκε.

### **4.2.2 Πάγωμα (Pause)**

Επιπλέον ο χρήστης θα μπορεί να παγώνει την προσομοίωση. Αντίστοιχα αυτό θα γίνεται πατώντας το κουμπί 'Pause' από την Μπάρα Λειτουργιών. Η προσομοίωση θα σταματάει προσωρινά και ο χρήστης θα μπορεί να εισαγει νεους οργανισμους με τον ιδιο τροπο που

περιγραφήκε πριν. Φυσικά η λειτουργία αυτή έχει νόημα μόνο εφόσον προηγουμένως έχει επιλέγει η λειτουργία που αντιστοιχεί στην έναρξη η στην επανεκκίνηση της προσομοίωσης

#### **4.2.3 Συνέχιση (Resume)**

Με αυτή τη λειτουργία το πρόγραμμα μας θα συνεχίζει την προσομοίωση από εκεί που σταμάτησε πριν με το πάτημα του κουμπιού 'Pause'. Η συνέχιση της προσομοίωσης θα γίνεται πατώντας πάνω στο κουμπί 'Resume' από τη Μπάρα Λειτουργίας. Ουσιαστικά αυτή η λειτουργία θα έχει νόημα μόνο εφόσον προηγουμένως έχει επιλέγει η λειτουργία που αντιστοιχεί στο πάγωμα της προσομοίωσης.

#### **4.2.4 Επανεκκίνηση (Restart)**

Εδώ ο χρήστης θα μπορεί να σβήσει οτιδήποτε είχε πριν εισάγει στον χάρτη και να ξεκινάει μια νέα προσομοίωση. Όλοι οι οργανισμοί που βρίσκονται στο χάρτη καθώς και οι πληροφορίες που βρίσκονται στο κάτω μέρος της οθόνης της προσομοίωσης θα χάνονται και θα ξεκινάει μια εκ νέου προσομοίωση. Η λειτουργία αυτή θα γίνεται πατώντας το κουμπί 'Restart' από τη Μπάρα Λειτουργιών. Από εκεί και πέρα θα ισχύουν τα ίδια με τη λειτουργία 'Start'.

### **4.3 Πληροφορίες ανά πληθυσμό οργανισμών**

Με αυτή τη λειτουργία ο χρήστης θα μπορεί να δει γενικές πληροφορίες που θα αφορούν την κατηγορία οργανισμών που αυτός θα επιλέγει από τη Μπάρα Αντικειμένων. Πρώτα θα πρέπει ο χρήστης να πατάει το κουμπί 'Statistics' από τη Μπάρα Λειτουργιών. Στη συνέχεια θα επιλέγει την κατηγορία οργανισμών του οποίου θέλει να δει πληροφορίες, πατώντας πάνω στο εικονίδιο που του αντιστοιχεί στη Μπάρα Αντικειμένων. Οι πληροφορίες που θα εμφανίζονται θα αφορούν στατιστικά, όπως το μέσο όρο μεγέθους των οργανισμών που είναι ζωντανοί την τρέχουσα στιγμή, τη μέση κατανάλωση τροφής την τελευταία εβδομάδα, ποσοστά θανάτων σε σχέση με το σύνολο των οργανισμών του χάρτη, στατιστικά των ηλικιών ανά οργανισμό αλλά και τον αριθμό των ζωντανών οργανισμών την τρέχουσα στιγμή. Οι πληροφορίες αυτές θα φαίνονται στο κάτω μέρος της οθόνης.

#### **4.4 Πληροφορίες ανά οργανισμό**

Επιπλέον ο χρήστης θα μπορεί να βλέπει πληροφορίες που αφορούν αποκλειστικά κάποιον οργανισμό και όχι την ευρύτερη κατηγορία στη οποία ανήκει. Αυτό θα είναι εφικτό μέσω της επιλογής κάποιου συγκεκριμένου οργανισμού από το χάρτη της προσομοίωσης. Αυτό θα γίνεται πατώντας πάνω στο εικονίδιο του επιθυμητού οργανισμού μέσα στο χάρτη όποτε και το αντίστοιχο εικονίδιο θα τοποθετείται σε ένα ορθογώνιο πλαίσιο, δείχνοντας ποιον οργανισμό αφορούν οι συγκεκριμένες πληροφορίες. Στη συνέχεια πληροφορίες που αφορούν χαρακτηριστικά όπως την ηλικία, το μέγεθος, το βαθμό ανάπτυξης, την ποσότητα τροφής που χρειάζεται ανά εβδομάδα ο οργανισμός για να αναπτυχθεί και την ποσότητα τροφής που έχει καταναλώσει ο οργανισμός την τρέχουσα εβδομάδα θα εμφανίζονται σε ένα πλαίσιο διακριτού χρώματος στο κάτω μέρος της οθόνης. Οι πληροφορίες αυτές θα αλλάζουν συνεχώς καθώς ο οργανισμός θα κινείται στο χάρτη.

### **5. Περιγραφή των κλάσεων και των packages που υλοποιούνται**

#### **5.1 Το package organisms**

Σε αυτό το package υπάρχουν όλες οι κλάσεις που υλοποιούν τους οργανισμούς που θα έχουμε στο οικοσύστημά μας. Το package organisms περιέχει τις αφηρημένες κλάσεις organisms, animals, plants, carnivores, herbivores και τις συγκεκριμένες κλάσεις lions, tigers, bears, elephants, giraffes, zebras, bushes και trees. Η ιεραρχία αυτών των κλάσεων συμπίπτει με την ιεραρχία των οργανισμών όπως την περιγράψαμε στο 3 και φαίνεται στο σχήμα που κάναμε εκεί. Οι κλάσεις animals και plants είναι υποκλάσεις της organisms, οι κλάσεις carnivores και herbivores είναι υποκλάσεις της animals, οι τελικές κλάσεις bushes και trees είναι υποκλάσεις της plants, οι τελικές κλάσεις lions, tigers και bears είναι υποκλάσεις της carnivores και τέλος οι τελικές κλάσεις elephants, giraffes και zebras είναι υποκλάσεις της herbivores. Παρακάτω περιγράφονται αναλυτικά όλες οι παραπάνω κλάσεις που αναφέραμε.

### 5.1.1 Η κλάση organisms

Είναι αφηρημένη κλάση και όλοι οι οργανισμοί βρίσκονται κάτω από αυτήν στην ιεραρχία των κλάσεων. Είναι δηλαδή υπερκλάση όλων των υπολοίπων κλάσεων που υπάρχουν σε αυτό το package είτε άμεσα είτε έμμεσα. Τα πεδία αυτής της κλάσης είναι:

- age: (protected int age) Η ηλικία του οργανισμού.
- energy: (protected int energy) Η ενέργεια του οργανισμού.
- growth\_rate: (protected final int growth\_rate) Ο ρυθμός ανάπτυξης του οργανισμού, είναι σταθερά.
- position: (protected Position position) Η θέση του οργανισμού στον χάρτη.
- alive\_organisms: (protected static int alive\_organisms) Πόσοι οργανισμοί είναι ζωντανοί συνολικά – στατική μεταβλητή.
- dead\_organisms: (protected static int dead\_organisms) Πόσοι οργανισμοί έχουν πεθάνει συνολικά – στατική μεταβλητή.

Τα πεδία της κλάσης organisms είναι protected μεταβλητές επειδή πρέπει να είναι κρυφά από το έξω κόσμο αλλά να κληρονομούνται στις υποκλάσεις της organisms. Οι μέθοδοι της κλάσης organisms είναι:

- public organisms(int g\_r) : Constructor – καλείται κατά την δημιουργία ενός νέου οργανισμού.
- public abstract void draw(Graphics g) : Abstract, Accesor-Selector. Ζωγραφίζει τον οργανισμό στον χάρτη με την κατάλληλη εικόνα.
- public boolean isAlive() : Accesor-Observer. Επιστρέφει true αν ο οργανισμός είναι ζωντανός, αλλιώς false.
- public int age() : Accesor-Selector. Επιστρέφει την ηλικία του οργανισμού.
- public int energy() : Accesor-Selector. Επιστρέφει την ενέργεια του οργανισμού.
- public int growth\_rate() : Accesor-Selector. Επιστρέφει τον ρυθμό ανάπτυξης του οργανισμού.
- public Position position() : Accesor-Selector. Επιστρέφει την θέση του οργανισμού.
- public int alive\_organisms() : Accesor-Selector. Επιστρέφει το πλήθος των ζωντανών οργανισμών.
- public int dead\_organisms() : Accesor-Selector. Επιστρέφει το πλήθος των νεκρών οργανισμών.

- `public void increase_age() : Transformer-Mutative`. Αυξάνει την ηλικία κατά ένα.
- `public void increase_energy() : Transformer-Mutative`. Αυξάνει την ενέργεια κατά `growth_rate`(ρυθμός ανάπτυξης).
- `public void decrease_energy() : Transformer-Mutative`. Μειώνει την ενέργεια κατά `growth_rate`(ρυθμός ανάπτυξης).
- `public abstract void restoreEnergy() : Abstract, Transformer-Mutative`. Ανανεώνει την ενέργεια ανάλογα με το είδος του οργανισμού.
- `public abstract void die() : Abstract, Transformer-Mutative`. Καλείται όταν πεθαίνει ένας οργανισμός και αυξομειώνει τις στατικές μεταβλητές για το πλήθος των ζωντανών και νεκρών οργανισμών.

### 5.1.2 Η κλάση `animals`

Είναι αφηρημένη και κάθε ζώο βρίσκεται κάτω από αυτήν στην ιεραρχία των κλάσεων. Είναι υποκλάση της κλάσης `organisms`. Τα πεδία αυτής της κλάσης, εκτός από τα κληρονομημένα, είναι:

- `foodPerWeek : (protected final int foodPerWeek)` Το φαγητό που πρέπει να φάει το ζώο ανά εβδομάδα.
- `eatenThisWeek : (protected int eatenThisWeek)` Το φαγητό που έχει φάει το ζώο μέχρι στιγμής αυτήν την εβδομάδα.
- `alive_animals : (protected static int alive_animals)` Πόσα ζώα είναι ζωντανά συνολικά – στατική μεταβλητή.
- `dead_animals : (protected static int dead_animals)` Πόσα ζώα είναι νεκρά συνολικά – στατική μεταβλητή.

Τα πεδία της κλάσης `animals` είναι `protected` μεταβλητές επειδή πρέπει να είναι κρυφά από το έξω κόσμο αλλά να κληρονομούνται στις υποκλάσεις της `animals`. Οι μέθοδοι της κλάσης `animals` είναι:

- `public animals(int g_r, int fpw) : Constructor` – καλείται κατά την δημιουργία ενός νέου ζώου.
- `public int foodPerWeek() : Accesor-Selector`. Επιστρέφει το φαγητό που πρέπει να φάει το ζώο ανά εβδομάδα.
- `public int eatenThisWeek() : Accesor-Selector`. Επιστρέφει το φαγητό που έχει φάει το ζώο μέχρι στιγμής.
- `public int alive_animals() : Accesor-Selector`. Επιστρέφει το πλήθος των ζωντανών ζώων.
- `public int dead_animals() : Accesor-Selector`. Επιστρέφει το πλήθος των νεκρών ζώων.

- `public void reset_eatenThisWeek() : Transformer-Mutative.` Μηδενίζει το φαγητό που έχει φάει το ζώο.
- `public void restoreEnergy() : Transformer-Mutative.` Υλοποιεί την `restoreEnergy()` της `organisms`. Ανανεώνει κατάλληλα την ενέργεια του ζώου.
- `public void move(int x, int y) : Transformer-Mutative.` Αλλάζει την θέση του ζώου.
- `public abstract void eat(organisms o) : Abstract, Transformer-Mutative.` Ελέγχει αν το ζώο τρώει τον οργανισμό `o` και αν τον τρώει αυξάνει κατάλληλα το `eatenThisWeek` του ζώου και μειώνει το `energy` του οργανισμού `o`.

### 5.1.3 Η κλάση `plants`

Είναι αφηρημένη και κάθε φυτό βρίσκεται κάτω από αυτήν στην ιεραρχία των κλάσεων. Είναι υποκλάση της κλάσης `organisms`. Τα πεδία αυτής της κλάσης, εκτός από τα κληρονομημένα, είναι:

- `alive_plants : (protected static int alive_plants)` Πόσα φυτά είναι ζωντανά συνολικά – στατική μεταβλητή.
- `dead_plants : (protected static int dead_plants)` Πόσα φυτά είναι νεκρά συνολικά – στατική μεταβλητή.

Οι μέθοδοι της κλάσης `plants` είναι:

- `public plants (int g_r) : Constructor` – καλείται κατά την δημιουργία ενός νέου φυτού.
- `public int alive_plants() : Accesor-Selector.` Επιστρέφει το πλήθος των ζωντανών φυτών.
- `public int dead_plants() : Accesor-Selector.` Επιστρέφει το πλήθος των νεκρών φυτών.
- `public void restoreEnergy() : Transformer-Mutative.` Υλοποιεί την `restoreEnergy()` της `organisms`. Ανανεώνει κατάλληλα την ενέργεια του φυτού.

### 5.1.4 Η κλάση `carnivores`

Είναι αφηρημένη και κάθε σαρκοφάγο ζώο βρίσκεται κάτω από αυτήν στην ιεραρχία των κλάσεων. Είναι υποκλάση της κλάσης `animals`. Τα πεδία αυτής της κλάσης, εκτός από τα κληρονομημένα, είναι:

- `alive_carnivores` : (protected static int `alive_carnivores`) Πόσα σαρκοφάγα ζώα είναι ζωντανά συνολικά – στατική μεταβλητή.
- `dead_carnivores`: (protected static int `dead_carnivores`) Πόσα σαρκοφάγα ζώα είναι νεκρά συνολικά – στατική μεταβλητή.

Οι μέθοδοι της κλάσης `carnivores` είναι:

- `public carnivores(int g_r, int fpw)` : Constructor – καλείται κατά την δημιουργία ενός νέου σαρκοφάγου ζώου.
- `public int alive_carnivores()` : Accessor-Selector. Επιστρέφει το πλήθος των ζωντανών σαρκοφάγων ζώων.
- `public int dead_carnivores()` : Accessor-Selector. Επιστρέφει το πλήθος των νεκρών σαρκοφάγων ζώων.
- `public abstract void eat(organisms o)` : Transformer-Mutative. Υλοποιεί την `eat()` της `animals`. Ελέγχει αν το σαρκοφάγο ζώο τρώει τον οργανισμό `o` και αν τον τρώει αυξάνει κατάλληλα το `eatenThisWeek` του ζώου και μειώνει το `energy` του οργανισμού `o`.

### 5.1.5 Η κλάση `herbivores`

Είναι αφηρημένη και κάθε φυτοφάγο ζώο βρίσκεται κάτω από αυτήν στην ιεραρχία των κλάσεων. Είναι υποκλάση της κλάσης `animals`. Τα πεδία αυτής της κλάσης, εκτός από τα κληρονομημένα, είναι:

- `alive_herbivores` : (protected static int `alive_herbivores`) Πόσα φυτοφάγα ζώα είναι ζωντανά συνολικά – στατική μεταβλητή.
- `dead_herbivores`: (protected static int `dead_herbivores`) Πόσα φυτοφάγα ζώα είναι νεκρά συνολικά – στατική μεταβλητή.

Οι μέθοδοι της κλάσης `herbivores` είναι:

- `public herbivores (int g_r, int fpw)` : Constructor – καλείται κατά την δημιουργία ενός νέου φυτοφάγου ζώου.
- `public int alive_herbivores()` : Accessor-Selector. Επιστρέφει το πλήθος των ζωντανών φυτοφάγων ζώων.
- `public int dead_herbivores()` : Accessor-Selector. Επιστρέφει το πλήθος των νεκρών φυτοφάγων ζώων.
- `public abstract void eat(organisms o)` : Transformer-Mutative. Υλοποιεί την `eat()` της `animals`. Ελέγχει αν το φυτοφάγο ζώο τρώει τον οργανισμό `o` και αν τον τρώει αυξάνει κατάλληλα το `eatenThisWeek` του ζώου και μειώνει το `energy` του οργανισμού `o`.

### 5.1.6 Η κλάση lions

Είναι τελική κλάση από την οποία δημιουργούνται όλα τα λιοντάρια. Είναι υποκλάση της κλάσης carnivores. Τα πεδία αυτής της κλάσης, εκτός από τα κληρονομημένα, είναι:

- `alive_lions` : (protected static int `alive_lions`) Πόσα λιοντάρια είναι ζωντανά συνολικά – στατική μεταβλητή.
- `dead_lions` : (protected static int `dead_lions`) Πόσα λιοντάρια είναι νεκρά συνολικά – στατική μεταβλητή.

Οι μέθοδοι της κλάσης lions είναι:

- `public lions(int g_r, int fpw)` : Constructor – καλείται κατά την δημιουργία ενός νέου λιονταριού.
- `public int alive_lions()`: Accessor-Selector. Επιστρέφει το πλήθος των ζωντανών λιονταριών.
- `public int dead_lions()`: Accessor-Selector. Επιστρέφει το πλήθος των νεκρών λιονταριών.
- `public void draw(Graphics g)` : Accessor-Selector. Υλοποιεί την `draw(Graphics g)` της `organisms`. Ζωγραφίζει μια εικόνα ενός λιονταριού στην θέση που αυτό βρίσκεται πάνω στον χάρτη.
- `public void die()` : Transformer-Mutative. Υλοποιεί την `die()` της `organisms`. Καλείται όταν πεθαίνει ένα λιοντάρι και μειώνει τις στατικές μεταβλητές για τους ζωντανούς οργανισμούς ενώ αυξάνει τις στατικές μεταβλητές για τους νεκρούς οργανισμούς.

### 5.1.7 Η κλάση tigers

Είναι τελική κλάση από την οποία δημιουργούνται όλες οι τίγρεις. Είναι υποκλάση της κλάσης carnivores. Τα πεδία αυτής της κλάσης, εκτός από τα κληρονομημένα, είναι:

- `alive_tigers` : (protected static int `alive_tigers`) Πόσες τίγρεις είναι ζωντανές συνολικά – στατική μεταβλητή.
- `dead_tigers` : (protected static int `dead_tigers`) Πόσες τίγρεις είναι νεκρές συνολικά – στατική μεταβλητή.

Οι μέθοδοι της κλάσης tigers είναι:

- `public tigers(int g_r, int fpw)` : Constructor – καλείται κατά την δημιουργία μιας νέας τίγρης.
- `public int alive_tigers()`: Accessor-Selector. Επιστρέφει το πλήθος των ζωντανών τίγρεων.

- `public int dead_tigers():` Accessor-Selector. Επιστρέφει το πλήθος των νεκρών τίγρεων.
- `public void draw(Graphics g) :` Accessor-Selector. Υλοποιεί την `draw(Graphics g)` της `organisms`. Ζωγραφίζει μια εικόνα μιας τίγρης στην θέση που αυτή βρίσκεται πάνω στον χάρτη.
- `public void die() :` Transformer-Mutative. Υλοποιεί την `die()` της `organisms`. Καλείται όταν πεθαίνει μια τίγρη και μειώνει τις στατικές μεταβλητές για τους ζωντανούς οργανισμούς ενώ αυξάνει τις στατικές μεταβλητές για τους νεκρούς οργανισμούς.

### 5.1.8 Η κλάση `bears`

Είναι τελική κλάση από την οποία δημιουργούνται όλες οι αρκούδες. Είναι υποκλάση της κλάσης `carnivores`. Τα πεδία αυτής της κλάσης, εκτός από τα κληρονομημένα, είναι:

- `alive_bears :` (`protected static int alive_bears`) Πόσες αρκούδες είναι ζωντανές συνολικά – στατική μεταβλητή.
- `dead_bears :` (`protected static int dead_bears`) Πόσες αρκούδες είναι νεκρές συνολικά – στατική μεταβλητή.

Οι μέθοδοι της κλάσης `bears` είναι:

- `public bears (int g_r, int fpw) :` Constructor – καλείται κατά την δημιουργία μιας νέας αρκούδας.
- `public int alive_bears():` Accessor-Selector. Επιστρέφει το πλήθος των ζωντανών αρκούδων.
- `public int dead_bears():` Accessor-Selector. Επιστρέφει το πλήθος των νεκρών αρκούδων.
- `public void draw(Graphics g) :` Accessor-Selector. Υλοποιεί την `draw(Graphics g)` της `organisms`. Ζωγραφίζει μια εικόνα μιας αρκούδας στην θέση που αυτή βρίσκεται πάνω στον χάρτη.
- `public void die() :` Transformer-Mutative. Υλοποιεί την `die()` της `organisms`. Καλείται όταν πεθαίνει μια αρκούδα και μειώνει τις στατικές μεταβλητές για τους ζωντανούς οργανισμούς ενώ αυξάνει τις στατικές μεταβλητές για τους νεκρούς οργανισμούς.

### 5.1.9 Η κλάση zebras

Είναι τελική κλάση από την οποία δημιουργούνται όλες οι ζέβρες. Είναι υποκλάση της κλάσης herbivores. Τα πεδία αυτής της κλάσης, εκτός από τα κληρονομημένα, είναι:

- `alive_zebras` : (protected static int `alive_zebras`) Πόσες ζέβρες είναι ζωντανές συνολικά – στατική μεταβλητή.
- `dead_zebras`: (protected static int `dead_zebras`) Πόσες ζέβρες είναι νεκρές συνολικά – στατική μεταβλητή.

Οι μέθοδοι της κλάσης zebras είναι:

- `public zebras(int g_r, int fpw)` : Constructor – καλείται κατά την δημιουργία μιας νέας ζέβρας.
- `public int alive_zebras()`: Accessor-Selector. Επιστρέφει το πλήθος των ζωντανών ζέβρων.
- `public int dead_zebras()`: Accessor-Selector. Επιστρέφει το πλήθος των νεκρών ζέβρων.
- `public void draw(Graphics g)` : Accessor-Selector. Υλοποιεί την `draw(Graphics g)` της `organisms`. Ζωγραφίζει μια εικόνα μιας ζέβρας στην θέση που αυτή βρίσκεται πάνω στον χάρτη.
- `public void die()` : Transformer-Mutative. Υλοποιεί την `die()` της `organisms`. Καλείται όταν πεθαίνει μια ζέβρα και μειώνει τις στατικές μεταβλητές για τους ζωντανούς οργανισμούς ενώ αυξάνει τις στατικές μεταβλητές για τους νεκρούς οργανισμούς.

### 5.1.10 Η κλάση elephants

Είναι τελική κλάση από την οποία δημιουργούνται όλες οι ελέφαντες. Είναι υποκλάση της κλάσης herbivores. Τα πεδία αυτής της κλάσης, εκτός από τα κληρονομημένα, είναι:

- `alive_elephants` : (protected static int `alive_elephants`) Πόσοι ελέφαντες είναι ζωντανοί συνολικά – στατική μεταβλητή.
- `dead_elephants`: (protected static int `dead_elephants`) Πόσοι ελέφαντες είναι νεκροί συνολικά – στατική μεταβλητή.

Οι μέθοδοι της κλάσης zebras είναι:

- `public elephants(int g_r, int fpw)` : Constructor – καλείται κατά την δημιουργία ενός νέου ελέφαντα.
- `public int alive_elephants()`: Accessor-Selector. Επιστρέφει το πλήθος των ζωντανών ελεφάντων.

- `public int dead_ elephants():` Accesor-Selector. Επιστρέφει το πλήθος των νεκρών ελεφάντων.
- `public void draw(Graphics g) :` Accesor-Selector. Υλοποιεί την `draw(Graphics g)` της `organisms`. Ζωγραφίζει μια εικόνα ενός ελέφαντα στην θέση που αυτός βρίσκεται πάνω στον χάρτη.
- `public void die() :` Transformer-Mutative. Υλοποιεί την `die()` της `organisms`. Καλείται όταν πεθαίνει ένας ελέφαντας και μειώνει τις στατικές μεταβλητές για τους ζωντανούς οργανισμούς ενώ αυξάνει τις στατικές μεταβλητές για τους νεκρούς οργανισμούς.

### 5.1.11 Η κλάση `giraffes`

Είναι τελική κλάση από την οποία δημιουργούνται όλες οι καμηλοπαρδάλεις. Είναι υποκλάση της κλάσης `herbivores`. Τα πεδία αυτής της κλάσης, εκτός από τα κληρονομημένα, είναι:

- `alive_ giraffes :` (`protected static int alive_ giraffes`) Πόσες καμηλοπαρδάλεις είναι ζωντανές συνολικά – στατική μεταβλητή.
- `dead_ giraffes:` (`protected static int dead_ giraffes`) Πόσες καμηλοπαρδάλεις είναι νεκρές συνολικά – στατική μεταβλητή.

Οι μέθοδοι της κλάσης `giraffes` είναι:

- `public giraffes(int g_r, int fpw) :` Constructor – καλείται κατά την δημιουργία μιας νέας καμηλοπάρδαλης.
- `public int alive_ giraffes():` Accesor-Selector. Επιστρέφει το πλήθος των ζωντανών καμηλοπαρδάλεων.
- `public int dead_ giraffes():` Accesor-Selector. Επιστρέφει το πλήθος των νεκρών καμηλοπαρδάλεων.
- `public void draw(Graphics g) :` Accesor-Selector. Υλοποιεί την `draw(Graphics g)` της `organisms`. Ζωγραφίζει μια εικόνα μιας καμηλοπάρδαλης στην θέση που αυτή βρίσκεται πάνω στον χάρτη.
- `public void die() :` Transformer-Mutative. Υλοποιεί την `die()` της `organisms`. Καλείται όταν πεθαίνει μια καμηλοπάρδαλη και μειώνει τις στατικές μεταβλητές για τους ζωντανούς οργανισμούς ενώ αυξάνει τις στατικές μεταβλητές για τους νεκρούς οργανισμούς.

### 5.1.12 Η κλάση bushes

Είναι τελική κλάση από την οποία δημιουργούνται όλες οι θάμνοι. Είναι υποκλάση της κλάσης plants. Τα πεδία αυτής της κλάσης, εκτός από τα κληρονομημένα, είναι:

- `alive_bushes` : (protected static int `alive_bushes`) Πόσοι θάμνοι είναι ζωντανοί συνολικά – στατική μεταβλητή.
- `dead_bushes`: (protected static int `dead_bushes`) Πόσοι θάμνοι είναι νεκροί συνολικά – στατική μεταβλητή.

Οι μέθοδοι της κλάσης bushes είναι:

- `public bushes(int g_r, int fpw)` : Constructor – καλείται κατά την δημιουργία ενός νέου θάμνου.
- `public int alive_bushes()`: Accessor-Selector. Επιστρέφει το πλήθος των ζωντανών θάμνων.
- `public int dead_bushes()`: Accessor-Selector. Επιστρέφει το πλήθος των νεκρών θάμνων.
- `public void draw(Graphics g)` : Accessor-Selector. Υλοποιεί την `draw(Graphics g)` της `organisms`. Ζωγραφίζει μια εικόνα ενός θάμνου στην θέση που αυτός βρίσκεται πάνω στον χάρτη.
- `public void die()` : Transformer-Mutative. Υλοποιεί την `die()` της `organisms`. Καλείται όταν πεθαίνει ένας ελέφαντας και μειώνει τις στατικές μεταβλητές για τους ζωντανούς οργανισμούς ενώ αυξάνει τις στατικές μεταβλητές για τους νεκρούς οργανισμούς.

### 5.1.13 Η κλάση trees

Είναι τελική κλάση από την οποία δημιουργούνται όλα τα δέντρα. Είναι υποκλάση της κλάσης plants. Τα πεδία αυτής της κλάσης, εκτός από τα κληρονομημένα, είναι:

- `alive_trees` : (protected static int `alive_trees`) Πόσα δέντρα είναι ζωντανά συνολικά – στατική μεταβλητή.
- `dead_trees`: (protected static int `dead_trees`) Πόσα δέντρα είναι νεκρά συνολικά – στατική μεταβλητή.

Οι μέθοδοι της κλάσης trees είναι:

- `public trees(int g_r, int fpw)` : Constructor – καλείται κατά την δημιουργία ενός νέου δέντρου.
- `public int alive_trees()`: Accessor-Selector. Επιστρέφει το πλήθος των ζωντανών δέντρων.

- `public int dead_trees():` Accessor-Selector. Επιστρέφει το πλήθος των νεκρών δέντρων.
- `public void draw(Graphics g) :` Accessor-Selector. Υλοποιεί την `draw(Graphics g)` της `organisms`. Ζωγραφίζει μια εικόνα ενός δέντρου στην θέση που αυτό βρίσκεται πάνω στον χάρτη.
- `public void die() :` Transformer-Mutative. Υλοποιεί την `die()` της `organisms`. Καλείται όταν πεθαίνει ένα δέντρο και μειώνει τις στατικές μεταβλητές για τους ζωντανούς οργανισμούς ενώ αυξάνει τις στατικές μεταβλητές για τους νεκρούς οργανισμούς.

## 5.2 Το package Position

Το package Position περιέχει μια μόνο κλάση, την Position. Η κλάση Position χρησιμοποιείται για να προσδιορίζουμε την θέση ενός αντικειμένου πάνω στον χάρτη. Τα πεδία αυτής της κλάσης είναι:

- `x : (private int x)` Η συντεταγμένη x της θέσης.
- `y : (private int y)` Η συντεταγμένη y της θέσης.
- `max_distance : (private final int max_distance)` Η μέγιστη απόσταση από την οποία ένα ζώο μπορεί να φάει έναν άλλον οργανισμό.
- `width : (private final int width)` Το πλάτος του κάθε αντικειμένου.
- `length : (private final int length)` Το μήκος του κάθε αντικειμένου.

Τα πεδία της κλάσης Position είναι private μεταβλητές επειδή πρέπει να είναι κρυφά από το έξω κόσμο. Οι μέθοδοι της κλάσης Position είναι:

- `public Position(int x, int y) :` Constructor. Δημιουργεί μια νέα θέση με συντεταγμένες x,y.
- `public int get_x() :` Accessor-Selector. Επιστρέφει την συντεταγμένη x.
- `public int get_y() :` Accessor-Selector. Επιστρέφει την συντεταγμένη y.
- `public void set_x(int x) :` Transformer-Mutative. Θέτουμε το x.
- `public void set_y(int y) :` Transformer-Mutative. Θέτουμε το y.
- `public boolean checkIfClose(Position p) :` Accessor-Observer. Ελέγχει αν οι δυο θέσεις των οργανισμών είναι τόσο κοντά ώστε να μπορεί ο ένας να φάει τον άλλο.
- `public boolean isEqual(Position p) :` Accessor-Observer. Ελέγχει αν δυο θέσεις είναι ίδιες.
- `public boolean isInside(Position p) :` Accessor-Observer. Ελέγχει αν ένα αντικείμενο είναι μέσα σε ένα άλλο.

## 5.3 Το package Operations

### 5.3.1 Η κλάση Task

Είναι η κλάση η οποία αφορά τη δημιουργία μιας ανάθεσης εργασίας, η οποία αργότερα θα μπαίνει σαν όρισμα σε ένα Object μιας κλάσης Timer. Έτσι θα καθορίζεται τι θα κάνει ένα Object αυτής της κλάσης (Timer) σε τακτά χρονικά διαστήματα. Η κλάση αυτή κληρονομεί την abstract κλάση Time Task έτσι ώστε να κάνει override την abstract μέθοδο run() και υλοποιεί το interface Runnable εν μέσω της κλάσης Time Task που κληρονομεί. Ουσιαστικά θα καθορίζει την κίνηση και την αλληλεπίδραση ενός οργανισμού με τους υπόλοιπους οργανισμούς του χάρτη προσομοίωσης σε καθημερινή βάση.

Τα πεδία, εκτός των πεδίων που αυτή η κλάση κληρονομεί, είναι :

- Vector orgs: (private) Το διάνυσμα μέσα στο οποίο θα κρατούνται όλοι οι οργανισμοί που υπάρχουν στο χάρτη της προσομοίωσης.

Οι μέθοδοι αυτής της κλάσης είναι:

- Public void run (): Transformer-mutative –Κάνει override την abstract μέθοδο της super class. Ουσιαστικά αλλάζει τα περιεχόμενα του orgs αφού ελέγχει επί καθημερινής βάσης τι αλληλεπιδράσεις μεταξύ των οργανισμών και ανάλογα ρυθμίζει τις συντεταγμένες τις θέσης τους καθώς και τα στατιστικά τους .
- Public void load Vector (Vector v): Transformer-mutative – Φορτώνει στον orgs τα περιεχόμενα του v, σβήνοντας τα παλιά περιεχόμενα του.

### 5.3.2 Η κλάση Operations

Η κλάση αυτή συνοψίζει τις λειτουργίες που αφορούν την προσομοίωση του οικοσυστήματος μας. Τέτοιες λειτουργίες είναι η δημιουργία ενός νέου οργανισμού, η έναρξη ,η παύση , η συνεχεία και η επανεκκίνηση της προσομοίωσης. Επιπλέον αφορά την απεικόνιση στατιστικών πληροφοριών που αφορούν τόσο μεμονωμένους οργανισμούς όσο και γενικότερα είδη οργανισμών.

Τα πεδία αυτής της κλάσης είναι:

- `Vector orgs (private)`: Σε αυτό το διάνυσμα αποθηκεύεται το σύνολο των οργανισμών που είναι ζωντανοί στο χάρτη του οικοσυστήματος.
- `timer t (private)`: Ένα instance της κλάσης `Timer` όπου θα καλούνται σε μεθόδους της, instances της `Task`.
- `int delay (private)`: Η χρονική καθυστέρηση που θέλουμε να έχει η προσομοίωση. Δηλαδή ο χρόνος σε milliseconds που θέλουμε να αντιστοιχεί σε μια νοητή ημέρα, οπότε και θα κινούνται οι οργανισμοί του οικοσυστήματος.
- `unsigned long int total Time (private)`: Ο συνολικός χρόνος (σε milliseconds) που αντιστοιχεί στη διάρκεια προσομοίωσης. Δηλαδή ο χρόνος που 'τρέχει' η προσομοίωση από τη στιγμή της εκκίνησης ή της επανεκκίνησης, χωρίς να λαμβάνουμε υπόψη το χρόνο που είναι 'παγωμένη' η προσομοίωση.
- `int week (private)`: Ο αριθμός της νοητής εβδομάδας της προσομοίωσης.
- `boolean running (private)`: Δηλώνει αν η προσομοίωση 'τρέχει' η είναι παγωμένη.

Οι μέθοδοι της κλάσης είναι:

- `public void create (position p, int organismType)`: Δημιουργεί ένα νέο αντικείμενο τύπου `Organisms` το οποίο χαρακτηρίζει μοναδικά (ως φύλλο στο δέντρο της κληρονομικότητας) και το οποίο έχει συντεταγμένες θέσης στο χάρτη που προσδιορίζονται από το `p`.
- `public void start ()`: Ξεκινάει την προσομοίωση και καθορίζει τις ενέργειες πάνω στους οργανισμούς που πρέπει να γίνονται τόσο σε καθημερινή όσο και σε εβδομαδιαία βάση.
- `public void pause ()`: Παγώνει την προσομοίωση και αφήνει τα χαρακτηριστικά των οργανισμών όπως ήταν και πριν το κάλεσμα της μεθόδου.
- `public void resume ()`: Επαναφέρει την προσομοίωση στο σημείο που ήταν πριν το κάλεσμα της `pause ()` και συνεχίζει από εκεί.
- `public void restart ()`: Σβήνει από τον `orgs` όλους τους οργανισμούς και από εκεί και πέρα ξεκινά μια νέα προσομοίωση όπως ορίζει και η `start ()`

- `public void population Info (int organismType)`: Προβάλλει στην οθόνη πληροφορίες που αφορούν στατιστικά του είδους των οργανισμών που καθορίζει η `organismType`.
- `public void organism Info (position p)`: Προβάλλει στην οθόνη πληροφορίες που αφορούν στατιστικά του μεμονωμένου οργανισμού που έχει θέση στο χάρτη, που καθορίζεται από το `p`.
- `public void drawAll (Graphics g)`: Διασχίζει το `orgs` και καλεί για κάθε οργανισμό τη μέθοδο `draw ()`.

#### 5.4 Οι κλάσεις του γραφικού περιβάλλοντος

Για το γραφικό περιβάλλον υλοποιήσαμε μία κλάση, την `Project`. Η κλάση αυτή θα είναι η κεντρική κλάση όλου του `Project` καθώς θα περιέχει και τη μέθοδο `main` η οποία θα εκτελεστεί. Είναι υποκλάση της `JApplet` και κάνει `implement` το `ActionListener`. Μέσα σε αυτή την κλάση ορίζονται όλα τα γραφικά αντικείμενα που θα χρησιμοποιηθούν στην εφαρμογή. Έτσι, σε αυτή την κλάση γίνεται η δημιουργία του χάρτη, των κουμπιών και των μάρων αντικειμένων και η τοποθέτησή τους πάνω στο κεντρικό παράθυρο της εφαρμογής. Χρησιμοποιεί μία μέθοδο `init()` η οποία ενώνει όλα τα επιμέρους στοιχεία του συστήματος σε ένα ενιαίο το οποίο προστίθεται στο `Frame`. Επίσης υπάρχει μία μέθοδος `actionPerformed` η οποία πιάνει τα πατήματα των κουμπιών μέσω του `ActionListener` και καλεί τις ανάλογες μεθόδους από την κλάση `Operations`.

Τα βασικά πεδία της κλάσης είναι:

- `name : (private static String name)` Το όνομα της εφαρμογής
- `livadi : (JFrame livadi)` Το βασικό `JFrame`.
- `Pane : (JPanel pane)` Ο χάρτης.
- Όλα τα κουμπιά. (`JButton start,pause,restart,resume,statistic;`)
- Τα πεδία κειμένου. (`JTextField individualinfo,statistics;`)

Οι βασικές μέθοδοι της κλάσης είναι:

- `public Component init()` : Καλείται μια φορά και αρχικοποιεί το κεντρικό παράθυρο.
- `public void actionPerformed(ActionEvent e)` : Καλεί τις επιτρεπτές λειτουργίες της εφαρμογής μέσω `events` – πατήματα κουμπιών.
- `public void paint(Graphics g)` : Ζωγραφίζει τον χάρτη με τα περιεχόμενά του.
- `public static void main(String[] args)` : Η κύρια μέθοδος της εφαρμογής.

## **6. Σχέσεις και επικοινωνία των κλάσεων**

Οι παραπάνω κλάσεις θα επικοινωνούν ως εξής: Πρώτα, η κεντρική κλάση Project θα φτιάχνει το γραφικό περιβάλλον της εφαρμογής, δηλαδή τον χάρτη, τα πεδία κειμένου και τις μπάρες με τα κουμπιά για τις λειτουργίες. Έπειτα, ανάλογα με την λειτουργία που εκτελείται κάθε φορά καλείται η κατάλληλη μέθοδος της κλάσης Operations. Η κλάση Operations χρησιμοποιεί την κλάση Task για τις λειτουργίες της προσομοίωσης. Η κλάση Operations και η κλάση Task χρησιμοποιούν την δομή δεδομένων Vector για να αποθηκεύουν τους ζωντανούς οργανισμούς που υπάρχουν στο οικοσύστημα. Για να υλοποιήσει τις λειτουργίες της η κλάση Operation χρησιμοποιεί τις μεθόδους των κλάσεων που ορίζονται στο package Organisms. Το ίδιο και η κλάση Task. Όλες οι κλάσεις χρησιμοποιούν την κλάση Position από το package Position για την θέση των αντικειμένων πάνω στον χάρτη. Οι κλάσεις στο package Organisms συνδέονται με σχέσεις κληρονομικότητας όπως ακριβώς περιγράφηκε στις παραγράφους 3 και 5.1. Σχηματικά, οι σχέσεις και η επικοινωνία των κλάσεων που υλοποιήθηκαν φαίνεται στο παρακάτω σχήμα.

