Μέσα στο σώμα της μεθόδου act(), που βρίσκεται ανάμεσα στα δύο άγκιστρα { }, γράφονται οι εντολές οι οποίες καθορίζουν τις δυνατότητες (συμπεριφορές), δηλαδή τις ενέργειες, που το αντικείμενο μπορεί να πραγματοποιήσει όταν εκτελεστεί η act().

Στον επεξεργαστή κώδικα μπορούμε να:

- Γράφουμε κώδικα για να προγραμματίζουμε τις ενέργειες που θα εκτελούν τα αντικείμενα των κλάσεων.

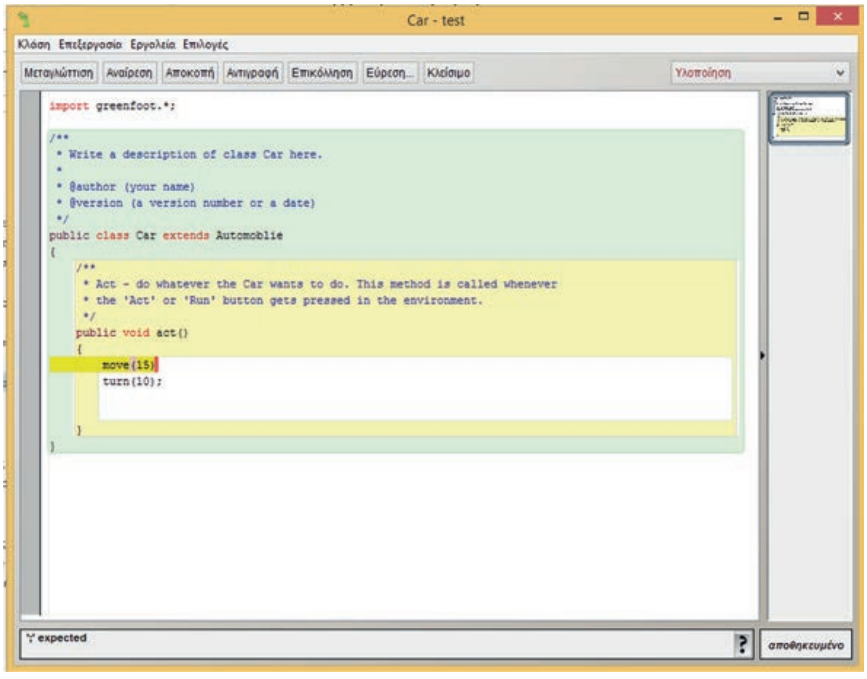
Τροποποιούμε τον κώδικα για ν' αλλάζουμε την συμπεριφορά ενός αντικειμένου

Ας γράψουμε τον παρακάτω κώδικα ανάμεσα στα δύο άγκιστρα { } της μεθόδου act() του αντικειμένου Frog:

```
move(15);
turn(10);
```

Κλείνουμε τον επεξεργαστή, πατάμε ***Μεταγλώττιση Όλων*** και δημιουργούμε ένα νέο αντικείμενο Frog. Πατάμε το κουμπί ***Εκκίνηση*** και βλέπουμε το βάτραχο να διαγράφει μικρούς κύκλους.

Κατά τη διαδικασία της μεταγλώττισης υπάρχει πιθανότητα να εντοπίσουμε κάποιο λάθος που έχουμε κάνει κατά τη συγγραφή του προγράμματος. Στην παρακάτω εικόνα έχουμε ξεχάσει το ερωτηματικό στο τέλος της εντολής move (Εικόνα 1.7.2).



Εικόνα 1.7.2 Συντακτικό λάθος της Java

Η ύπαρξη λαθών, κατά τη συγγραφή προγραμμάτων, αποτελεί σύνηθες φαινόμενο. Ορισμένα από αυτά μπορεί να εμποδίζουν τη μεταγλώττιση τους. Το λάθος που είδαμε παραπάνω λέγεται **συντακτικό** και ανιχνεύεται κατά τη μεταγλώττιση. Ο προγραμματιστής μπορεί να διαβάζει τα τυχόν μηνύματα λάθους που του εμφανίζει το περιβάλλον του Greenfoot, να τα διορθώνει και να μεταγλωττίζει ξανά το πρόγραμμα του. Ένας χαρακτήρας που λείπει ή έχει γραφεί λάθος, η παράλειψη π.χ. του συμβόλου του ερωτηματικού (;) στο τέλος μιας εντολής, μπορεί να προκαλέσει την αποτυχία εκτέλεσης του προγράμματος μας.

Η δήλωση import γίνεται στην αρχή του αρχείου της κλάσης πριν ξεκινήσει ο ορισμός της. Αν απαιτείται η εισαγωγή περισσοτέρων πακέτων, μπορούμε να χρησιμοποιήσουμε περισσότερες δηλώσεις import, πάντα στην αρχή του αρχείου.

Μια κλάση που θα χρησιμοποιούμε συχνά είναι η Math, του πακέτου java.lang, η οποία περιέχει όλες τις μαθηματικές μεθόδους που θα χρειαστούμε.

Παρακάτω δίνουμε μερικά παραδείγματα χρήσης μεθόδων της κλάσης Math.

```
x = Math.max(a, b) ;        // Επιστρέφει στην x το μέγιστο των a, b
root = Math.sqrt(x) ;       // Επιστρέφει την τετραγωνική ρίζα του x
```

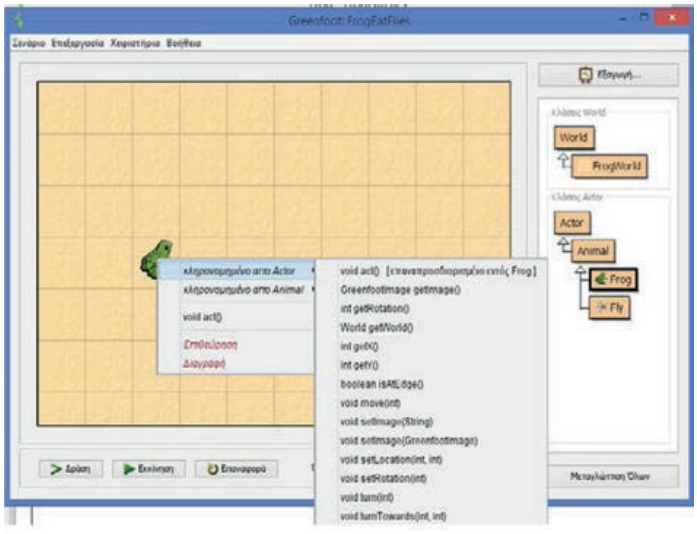
Πολύ συχνά χρησιμοποιούμε τη **μέθοδο println**, η οποία εμφανίζει στην οθόνη ένα μήνυμα. Η println είναι μια μέθοδος του **αντικειμένου out** που αναπαριστά το ρεύμα εξόδου, το οποίο συνήθως κατευθύνεται στην οθόνη. Το αντικείμενο out ανήκει στην **κλάση System**. Έτσι αν θέλουμε να εμφανίσουμε το μήνυμα : *Καλημέρα γαλαξία* γράφουμε:

```
System.out.println("Καλημέρα γαλαξία");
```

Υπάρχει μεγάλο πλήθος βιβλιοθηκών της java που περιέχουν πάρα πολλές μεθόδους κλάσεων. Αυτό είναι ένα από τα βασικά πλεονεκτήματα της Java, καθώς μπορούμε να υλοποιήσουμε πολύπλοκες εφαρμογές, χρησιμοποιώντας έτοιμο κώδικα που αναζητούμε στην πληθώρα των βιβλιοθηκών της, γλυτώνοντας την εκ νέου συγγραφή του. Δεν είμαστε υποχρεωμένοι και ούτε έχει νόημα να γνωρίζουμε όλες τις κλάσεις και τις μεθόδους ενός πακέτου. Συνήθως, όπως θα δούμε στη συνέχεια, αναζητούμε τη μέθοδο που ψάχνουμε στην **τεκμηρίωση** της κλάσης της.

4η εργαστηριακή άσκηση:

Από το περιβάλλον του Greenfoot, ανοίγουμε το σενάριο FrogEatFlies. Αφού δημιουργήσουμε μερικά αντικείμενα των κλάσεων, που έχουμε ορίσει ως τώρα, μπορούμε να τους δώσουμε εντολές για να εκτελέσουν κάποιες ενέργειες. Αρκεί να κάνουμε δεξί κλικ επάνω τους, όπως φαίνεται στην Εικόνα 2.4.1 για το αντικείμενο της κλάσης Frog. Στο μενού που εμφανίζεται βλέπουμε την επιλογή «κληρονομημένο από Actor» και αν την επιλέξουμε με το ποντίκι εμφανίζονται όλες οι μέθοδοι που έχουν κληρονομηθεί από την κλάση Actor. Η κλάση Frog είναι υποκλάση της Actor και σύμφωνα με τον ορισμό της κληρονομικότητας που δώσαμε στο προηγούμενο κεφάλαιο, κληρονομεί όλες τις μεθόδους της.



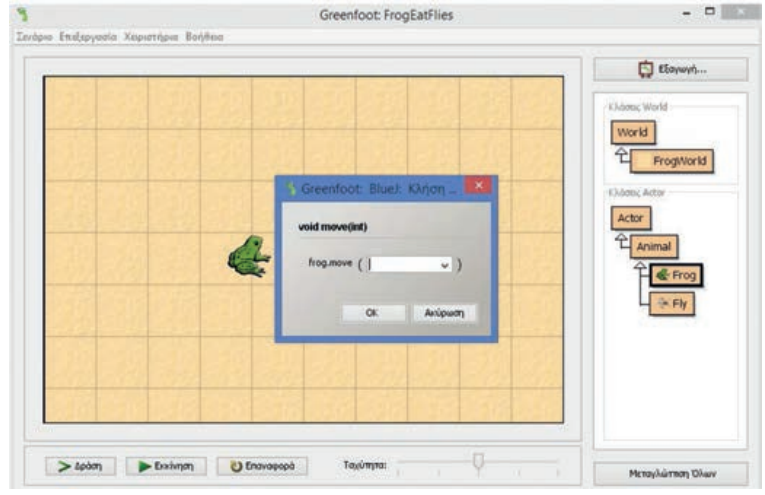
Εικόνα 2.4.1 Οι μέθοδοι του αντικειμένου Frog

Μερικές από τις μεθόδους, που κληρονομεί το αντικείμενο Frog δίνουν, όταν εκτελούνται, εντολή στο αντικείμενο να κάνει κάποια ενέργεια, όπως βλέπουμε στον Πίνακα που ακολουθεί.

move(int)	Κινεί το αντικείμενο προς την κατεύθυνση που βρίσκεται κατά συγκεκριμένα βήματα
turn(int)	Στρίβει το αντικείμενο κατά συγκεκριμένες μοίρες
act()	Το αντικείμενο κάνει μια ενέργεια
setLocation(int, int)	Ορίζει νέα θέση του αντικειμένου
setRotation(int)	Ορίζει την περιστροφή του αντικειμένου

Πίνακας 6 Μέθοδοι που δίνουν εντολή για να ενεργήσει το αντικείμενο

Παρατηρούμε ότι κάποιες μέθοδοι έχουν τη λέξη int μεταξύ των παρενθέσεων π.χ. η μέθοδος move(int). Η λέξη int μέσα στην παρένθεση ονομάζεται **παράμετρος** και σημαίνει ότι θα πρέπει να ορίσουμε κάποια τιμή, όταν επικαλούμαστε αυτή τη μέθοδο. Ο όρος 'int' υποδηλώνει ότι ένας ακέραιος αριθμός αναμένεται να δοθεί που θα προσδιορίσει πόσα βήματα θα κινηθεί ο βάτραχος όταν επικαλεστούμε τη συγκεκριμένη μέθοδο. Όταν λοιπόν την επιλέξουμε θα εμφανιστεί ένα παράθυρο διαλόγου που αναμένει να εισάγουμε μια τιμή για αυτήν την παράμετρο (Εικόνα 2.4.2). Πληκτρολογήστε έναν αριθμό (π.χ.: 20) και πατήστε Ok. Παρατηρούμε ότι ο βάτραχος θα κινηθεί 20 βήματα.



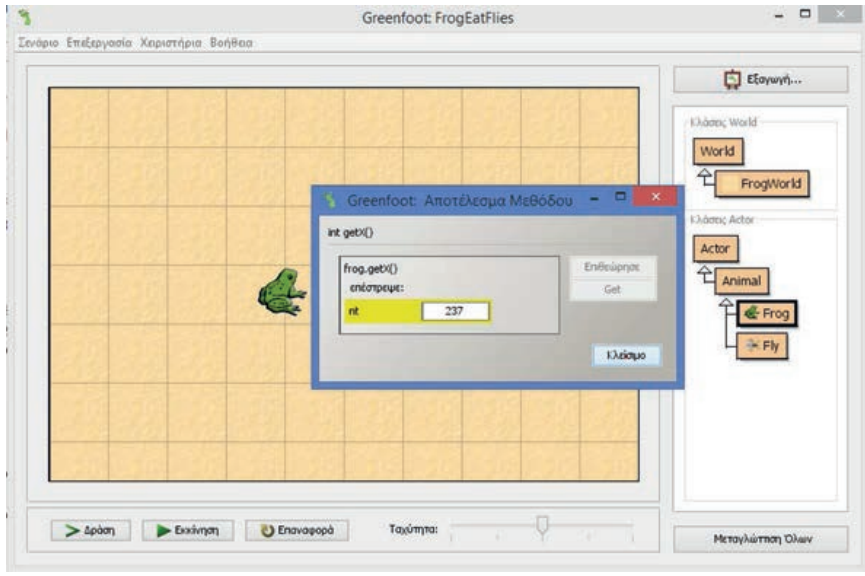
Εικόνα 2.4.2 Κλήση μεθόδου

Μερικές από τις μεθόδους, που κληρονομεί το αντικείμενο Frog δίνουν, όταν εκτελούνται, πληροφορίες για το αντικείμενο. Αυτές είναι οι:

getRotation()	Δίνει την τρέχουσα περιστροφή του αντικειμένου
getWorld()	Δίνει το όνομα του Κόσμου που βρίσκεται το αντικείμενο
getX()	Δίνει την συντεταγμένη χ της τρέχουσας θέσης του αντικειμένου
getY()	Δίνει την συντεταγμένη y της τρέχουσας θέσης του αντικειμένου

Πίνακας 7 Μέθοδοι που δίνουν πληροφορίες για το αντικείμενο

Στις μεθόδους αυτές, που δεν δέχονται παραμέτρους, η παρένθεση δεν περιέχει τίποτα. Όταν επιλέξουμε μια από αυτές θα εμφανιστεί στην οθόνη μας ένα παράθυρο διαλόγου που μας δίνει μια συγκεκριμένη πληροφορία για το αντικείμενο. Π.χ. αν επιλέξουμε τη μέθοδο getX() θα δούμε την τρέχουσα θέση του αντικειμένου στον άξονα του x (Εικόνα 2.4.3).



Εικόνα 2.4.3 Αποτέλεσμα μεθόδου

Εκτελέστε την μέθοδο setLocation(int, int) στο αντικείμενο Frog δίνοντας στο παράθυρο διαλόγου που εμφανίζεται τις συντεταγμένες των αξόνων x και y της νέας θέσης που επιθυμείτε να μετακινηθεί το αντικείμενο. Κατόπιν εκτελέστε τη μέθοδο getX() και επιβεβαιώστε ότι η τιμή που θα εμφανιστεί είναι η συντεταγμένη του άξονα x που δώσατε στην προηγούμενη μέθοδο setLocation. Δοκιμάστε την μέθοδο setLocation με διαφορετικές τιμές. Τι συμβαίνει αν πληκτρολογείτε στις παραμέτρους μια λέξη η έναν δεκαδικό αριθμό;

2.5. Τεκμηρίωση Λογισμικού

Η τεκμηρίωση μιας κλάσης είναι πολύ σημαντική γιατί αποτελεί την εικόνα της προς τον έξω κόσμο. Όποιος θέλει να χρησιμοποιήσει μεθόδους της κλάσης διαβάζει την τεκμηρίωση ***(Documentation)*** για να πληροφορηθεί σχετικά με τη λειτουργία των μεθόδων και τον τρόπο κλήσης τους. Η τεκμηρίωση των βιβλιοθηκών που έρχονται μαζί με τη γλώσσα Java είναι διαθέσιμη μέσω του παγκόσμιου ιστού. Είναι επίσης ενσωματωμένη μέσα στα περισσότερα ολοκληρωμένα προγραμματιστικά περιβάλλοντα (IDEs).

5η εργαστηριακή άσκηση:

Ας γυρίσουμε πίσω στο σενάριο μας. Με δεξί κλικ πάνω στην κλάση Actor επιλέγουμε ***Open Documentation*** για να μας ανοίξει στο φυλλομετρητή, την τεκμηρίωση της κλάσης Actor (Εικόνα 2.5.1).

Package Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: NESTED | FIELD | CONSTR | METHOD

greenfoot

Class Actor

```
java.lang.Object
  └ greenfoot.Actor
```

```
public abstract class Actor
extends java.lang.Object
```

An Actor is an object that exists in the Greenfoot world. Every Actor has a location in the world, and an appearance (that is: an icon).

An Actor is not normally instantiated, but instead used as a superclass to more specific objects in the world. Every object that is intended to appea

One of the most important aspects of this class is the 'act' method. This method is called when the 'Act' or 'Run' buttons are activated in the Green

Version:
2.5
Author:
Poul Henriksen

Constructor Summary

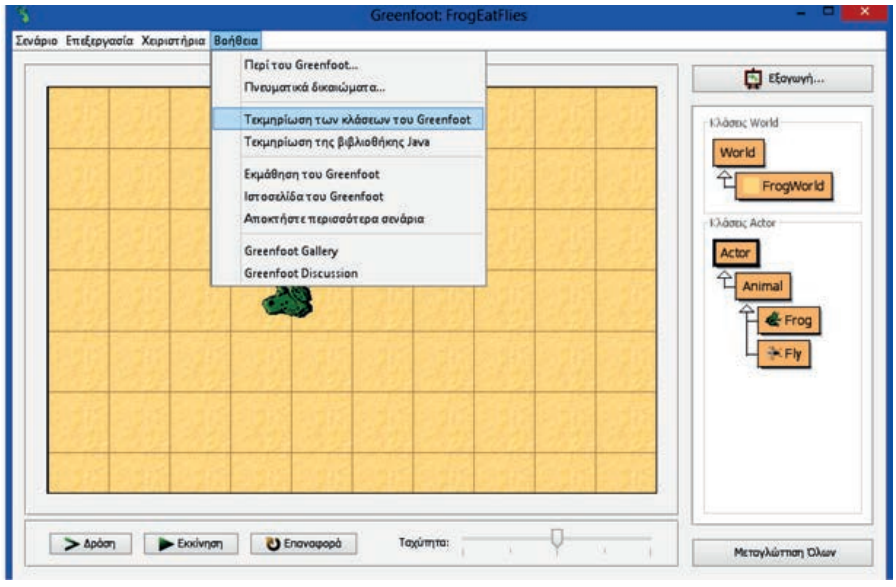
Actor()	Construct an Actor.

Method Summary

void	act() The act method is called by the greenfoot framework to give actors a chance to perform some action.
protected void	addedToWorld(World world) This method is called by the Greenfoot system when this actor has been inserted into the world.
protected java.util.List	getIntersectingObjects(java.lang.Class cls) Return all the objects that intersect this object.
protected java.util.List	getNeighbours(int distance, boolean diagonal, java.lang.Class cls) Return the neighbours to this object within a given distance.
protected java.util.List	getObjectsAtOffset(int dx, int dy, java.lang.Class cls) Return all objects that intersect the center of the given location (relative to this object's location).
protected java.util.List	getObjectsInRange(int radius, java.lang.Class cls) Return all objects within range 'radius' around this object.
protected Actor	getOneIntersectingObject(java.lang.Class cls) Return an object that intersects this object.
protected Actor	getOneObjectAtOffset(int dx, int dy, java.lang.Class cls) Return one object that is located at the specified cell (relative to this objects location).
int	getRotation() Return the current rotation of this actor.
World	getWorld() Return the world that this actor lives in.
int	getX() Return the x-coordinate of the actor's current location.
int	getY() Return the y-coordinate of the object's current location.
protected boolean	intersects(Actor other) Check whether this object intersects with another given object.
boolean	isAtEdge() Detect whether the actor has reached the edge of the world.
protected boolean	isTouching(java.lang.Class cls) Checks whether this actor is touching any other objects of the given class.
void	move(int distance) Move this actor the specified distance in the direction it is currently facing.
protected void	removeTouching(java.lang.Class cls) Removes one object of the given class that this actor is currently touching (if any exist).

Εικόνα 2.5.1 Τεκμηρίωση της κλάσης Actor

Για να δούμε την τεκμηρίωση όλων των κλάσεων του Greenfoot πηγαίνουμε στο μενού Βοήθεια και επιλέγουμε ***Τεκμηρίωση των κλάσεων του Greenfoot***, όπως φαίνεται στην Εικόνα 2.5.2.



Εικόνα 2.5.2 Το μενού Βοήθεια

Στην ιστοσελίδα που ανοίγει εμφανίζονται οι κλάσεις του πακέτου greenfoot με μια μικρή περίληψη του τι κάνει η κάθε μια από αυτές, όπως φαίνεται στην Εικόνα 2.5.3.

Class Summary	
Actor	Ο ηθοποιός (Actor) είναι ένα αντικείμενο το οποίο υπάρχει μέσα στον κόσμο του Greenfoot.
Greenfoot	Αυτή η κλάση είναι γενικής χρήσης και παρέχει μεθόδους για τον έλεγχο της εξομοίωσης και την αλληλεπίδραση με το σύστημα.
GreenfootImage	Μία εικόνα που θα εμφανίζεται στην οθόνη.
MouseInfo	Αυτή η κλάση περιέχει πληροφορίες για την κατάσταση του ποντικιού.
World	World είναι ο κόσμος μέσα στον οποίο ζουν οι ηθοποιοί (Actors).

Εικόνα 2.5.3 Κλάσεις του πακέτου greenfoot

Για να ρίξουμε μια αναλυτικότερη ματιά σε κάποια από τις κλάσεις, πατάμε με το ποντίκι πάνω στο όνομα της κλάσης και θα οδηγηθούμε στην ιστοσελίδα της, όπου θα δούμε ποιες μεθόδους παρέχει.

Περιηγηθείτε και στις πέντε κλάσεις του πακέτου greenfoot και προσπαθήστε να καταλάβετε τι λειτουργίες (μεθόδους) υποστηρίζει η κάθε μια.

Από το μενού Βοήθεια μπορούμε να δούμε επίσης, την τεκμηρίωση της βιβλιοθήκης Java που περιλαμβάνει όλα τα πακέτα και τις κλάσεις της βιβλιοθήκης.

2.5.1. Σχόλια

Ένα σημαντικό στοιχείο που πρέπει ν' αναφερθεί είναι η δυνατότητα να προσθέτουμε σχόλια στο πρόγραμμα μας.

Τα σχόλια χρησιμοποιούνται για να:

- Τεκμηριώνουμε: τον σκοπό και τους στόχους του προγράμματος μας, τον συγγραφέα του προγράμματος, τις εκδόσεις του κώδικά μας (revision history) κτλ.
- Περιγράφουμε: πεδία, μεθόδους, κλάσεις και κατασκευαστές.

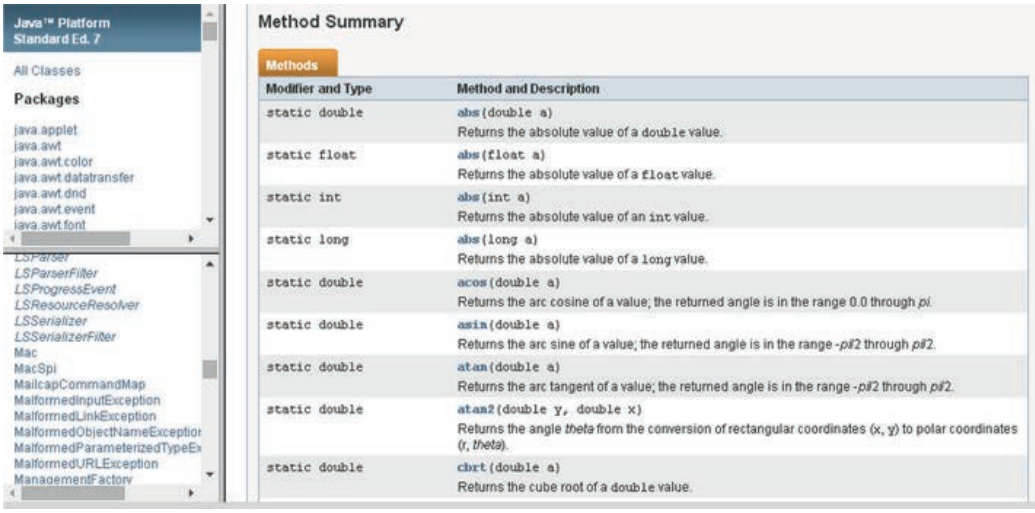
Τα σχόλια που γράφουμε στον πηγαίο κώδικα δεν εμφανίζονται στην εκτέλεση του προγράμματος μας.

Ένα σχόλιο πολλών γραμμών γράφεται ανάμεσα στα σύμβολα /* και */.

Ένα σχόλιο μιας γραμμής γράφεται μετά από το σύμβολο // μέχρι το τέλος της γραμμής.

Η δημιουργία τεκμηρίωσης για τις δικές μας κλάσεις στην Java είναι πολύ απλή και γίνεται εντελώς αυτοματοποιημένα, με χρήση του εργαλείου **javadoc** της γλώσσας. Το javadoc είναι ένα πρόγραμμα για αυτόματη κατασκευή τεκμηρίωσης σε μορφή HTM. Είναι αρκετά χρήσιμο στην κατασκευή βοηθημάτων και τεχνικών αναφορών για εφαρμογές οποιουδήποτε μεγέθους. Παρέχεται ως τμήμα του καθιερωμένου Πακέτου Ανάπτυξης (Development Kit), διατρέχει τον κώδικα ενός πακέτου και δημιουργεί τεκμηρίωση για κάθε μια από τις κλάσεις που αυτό περιέχει.

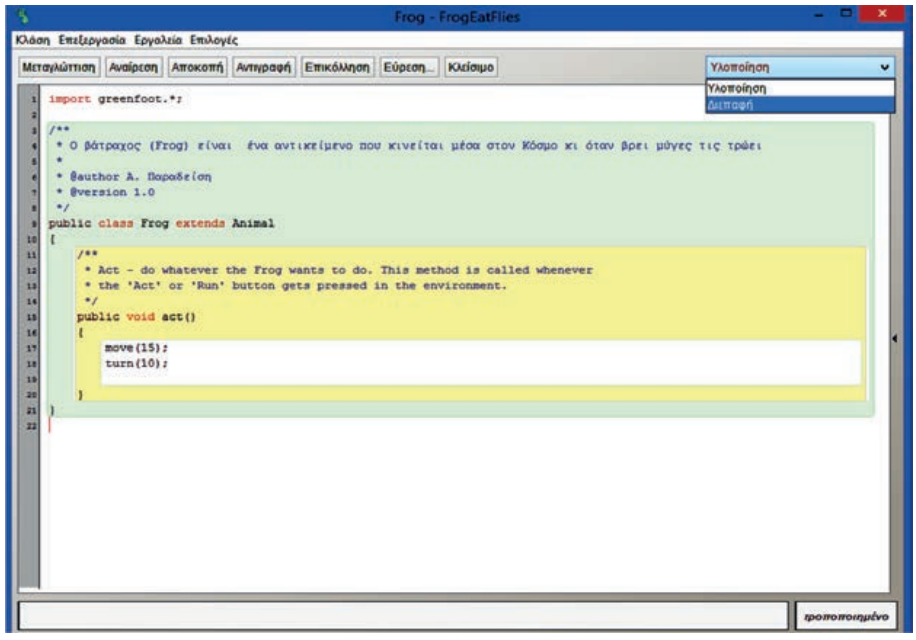
Παρακάτω στην Εικόνα 2.5.4, φαίνεται η τεκμηρίωση της κλάσης Math μέσα από τον παγκόσμιο ιστό, όπως έχει ανακτηθεί από την διεύθυνση http://docs.oracle.com/javase/7/docs/api/



Εικόνα 2.5.4 Τεκμηρίωση κλάσης Math

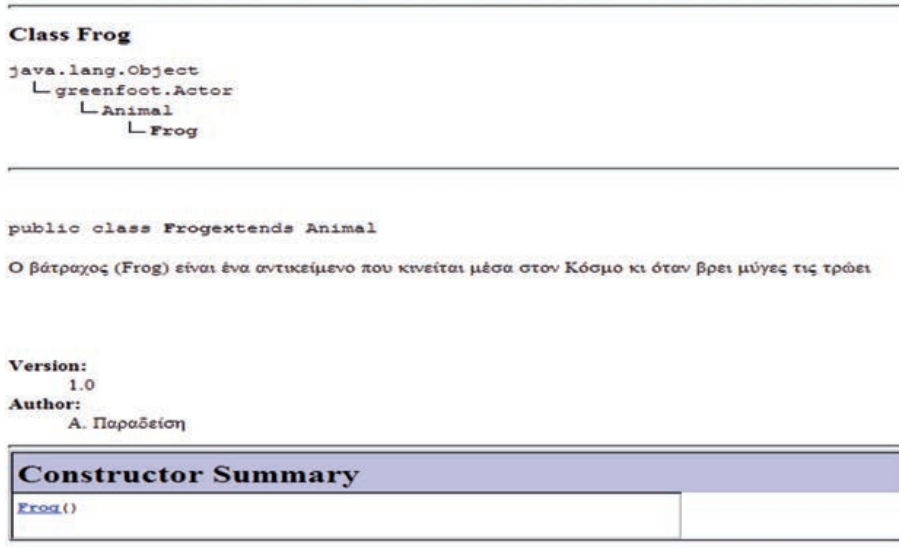
6η εργαστηριακή άσκηση:

Ας γυρίσουμε στο σενάριο μας κι ας προσθέσουμε κάποια σχόλια μέσα στις κλάσεις που έχουμε δημιουργήσει. Στην κλάση Frog ανοίγουμε τον επεξεργαστή και στο πάνω μέρος του αρχείου, στην περιοχή των σχολίων, γράφουμε μια μικρή περιγραφή της κλάσης όπως: «Ο βάτραχος (Frog) είναι ένα αντικείμενο που κινείται μέσα στον Κόσμο κι όταν βρει μύγες τις τρώει». Παρακάτω δίπλα στη λέξη @author θα γράψετε το όνομα σας και δίπλα στη λέξη @version θα γράψετε τον αριθμό 1.0 που δηλώνει ότι αυτή είναι η πρώτη έκδοση του σεναρίου σας (Εικόνα 2.5.5)



Εικόνα 2.5.5 Εισαγωγή σχολίων

Τα σχόλια δημιουργήθηκαν και τώρα θα εμφανίζονται στην τεκμηρίωση της κλάσης Frog. Μπορούμε να δούμε την τεκμηρίωση της κλάσης επιλέγοντας τη **Διεπαφή** από το αναδυόμενο μενού στην πάνω δεξιά γωνία του επεξεργαστή (Εικόνα 2.5.5).



Εικόνα 2.5.6 Τεκμηρίωση της κλάσης Frog

2.6. Διαγνωστικά Μηνύματα

Όλες οι γλώσσες προγραμματισμού, που παρέχουν τη δυνατότητα δημιουργίας παραθυρικών εφαρμογών, διαθέτουν μια σειρά διαλόγων που ονομάζονται **κοινοί διάλογοι (common dialogs)** και διευκολύνουν την εισαγωγή δεδομένων από τον χρήστη, την προβολή μηνυμάτων κλπ. Η Java παρέχει το πακέτο **javax.swing** για τη δημιουργία γραφικού περιβάλλοντος διεπαφής (GUI). Η κλάση **JOptionPane** του πακέτου διευκολύνει την χρήση παραθύρων διαλόγου για διαγνωστικά μηνύματα, είσοδο και έξοδο δεδομένων. Για μια ολοκληρωμένη γνώση της κλάσης JOptionPane και τις μεθόδους της, μπορείτε να ανατρέξετε στο documentation της java στην διεύθυνση: http://docs.oracle.com/javase/7/docs/api/.

Στην κλάση JOptionPane υπάρχουν τέσσερεις τύποι παραθύρων διαλόγου:

- Επιβεβαίωσης (ConfirmDialog)
- Εισόδου (InputDialog)
- Μηνύματος (MessageDialog)
- Επιλογής (OptionDialog)

7η εργαστηριακή άσκηση:

Ανοίγουμε το σενάριο FrogEatFlies στο Greenfoot. Αυτό που θέλουμε να κάνουμε είναι να προσθέσουμε στο πρόγραμμα μας τη λειτουργία να δείχνει ένα καλωσόρισμα στο χρήστη με παραθυρικό τρόπο, όταν αυτός πατήσει με το ποντίκι του πάνω στο αντικείμενο Frog.

Ανοίγουμε τον επεξεργαστή της κλάσης Frog για να γράψουμε τον κώδικα που χρειάζεται.

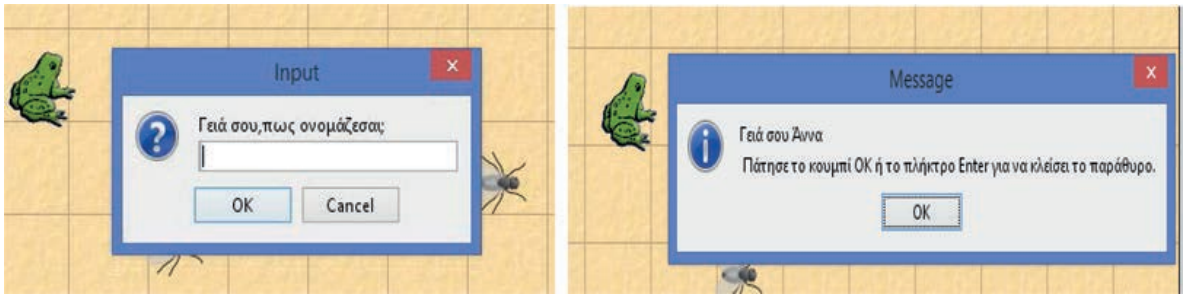
Η πρώτη μέθοδος της κλάσης JOptionPane που θα χρησιμοποιήσουμε είναι η showInputDialog που κάνει δυνατή την εισαγωγή δεδομένων, ζητώντας παράλληλα τα δεδομένα αυτά από τον χρήστη. Η δεύτερη μέθοδος της κλάσης JOptionPane που θα χρησιμοποιήσουμε είναι η showMessageDialog που εμφανίζει ένα μήνυμα με το καλωσόρισμα στο συγκεκριμένο χρήστη.

Στην αρχή του αρχείου της κλάσης θα γράψουμε την εντολή: import javax.swing.JOptionPane; για να ενημερώσουμε τον μεταγλωττιστή ότι θα χρησιμοποιήσουμε την κλάση JOptionPane που βρίσκεται στο πακέτο javax.swing.

Κατόπιν μέσα στο σώμα της μεθόδου act θα διαγράψουμε τις εντολές move και turn που υπάρχουν και θα γράψουμε τον παρακάτω κώδικα:

```
public void act()
  {
    if (Greenfoot.mousePressed(this)) {

        String inputStr = JOptionPane.showInputDialog("Γειά σου,πως ονομάζεσαι;");
        JOptionPane.showMessageDialog(null, "Γειά σου " + inputStr + "\n Πάτησε το κουμπί OK
                                          ή το πλήκτρο Enter για να κλείσει το παράθυρο.");
}
```

Κλείνουμε τον επεξεργαστή και κάνουμε μεταγλώττιση του προγράμματος. Κατόπιν, με δεξί κλικ πάνω στην κλάση Frog επιλέγουμε new Frog (), δημιουργούμε ένα νέο αντικείμενο Frog και το τοποθετούμε στον Κόσμο μας. Πατάμε το πλήκτρο **Εκκίνηση** και κάνουμε κλικ με το ποντίκι πάνω στο βάτραχο. Στο παράθυρο που εμφανίζεται δίνουμε το όνομα μας και κατόπιν εμφανίζεται το μήνυμα καλωσορίσματος, όπως φαίνεται στην Εικόνα 2.6.1.

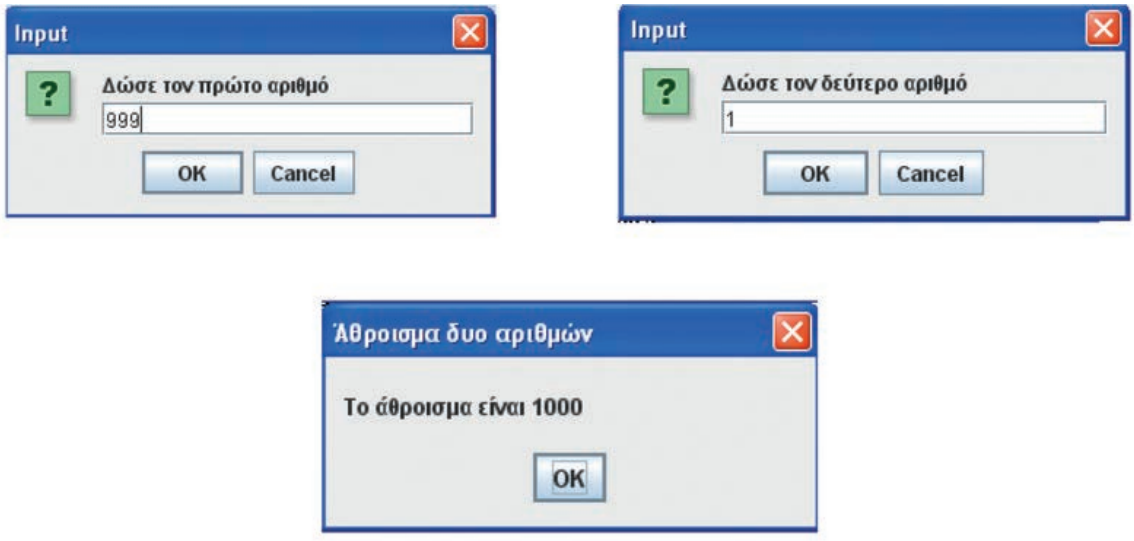


Εικόνα 2.6.1 Παράθυρα διαλόγου στο Greenfoot

Μπορούμε να τροποποιήσουμε την act έτσι ώστε να διαβάζει δύο (2) αριθμούς που θα εισάγει ο χρήστης και να εμφανίζει το άθροισμά τους.

```
import javax.swing.JOptionPane;

        public void act ( ) {
            String x = JOptionPane.showInputDialog("Δώσε τον πρώτο αριθμό");
            String y = JOptionPane.showInputDialog("Δώσε τον δεύτερο αριθμό");

            int number1 = Integer.parseInt(x);
            int number2 = Integer.parseInt(y);

            int sum = number1 + number2;

            JOptionPane.showMessageDialog(null, "Το άθροισμα είναι " + sum,
                                    "Άθροισμα δυο αριθμών", JOptionPane.PLAIN_MESSAGE);
    }
```

Τα παράθυρα διαλόγου που βλέπουμε στην οθόνη όταν τρέξουμε το πρόγραμμα φαίνονται στην Εικόνα 2.6.2.



Εικόνα 2.6.2 Παράθυρα διαλόγου της κλάσης JOptionPan

Αντί για PLAIN_MESSAGE για το είδος του μηνύματος υπάρχουν και οι εξής επιλογές:

τις οποίες μπορείτε να δοκιμάσετε και εσείς για να δείτε τα αντίστοιχα αποτελέσματα.