

Στατικές Μέθοδοι και Αντικείμενα στη Java

Μια πρακτική παρουσίαση για την κατανόηση των στατικών μεθόδων, της δημιουργίας μεθόδων και της αλληλεπίδρασης αντικειμένων στη Java



Δύο Τύποι Μεθόδων στη Java

Μέθοδοι Στιγμιότυπου

Ανήκουν στο αντικείμενο και καθορίζουν τη συμπεριφορά του.
Καλούνται μέσω ενός αντικειμένου.

```
kermit.move(100);
```

Μέθοδοι Κλάσης

Ανήκουν στην κλάση και δεν αναφέρονται σε συγκεκριμένο αντικείμενο. Ορίζονται με το **static**.

```
Frog.numberOfFrogs();
```

Οι στατικές μέθοδοι μπορούν να κληθούν χωρίς να δημιουργηθεί αντικείμενο της κλάσης.

Παραδείγματα Στατικών Μεθόδων

Κλάση Math

Όλες οι μέθοδοι της είναι στατικές:

```
int x = Math.max(a, b);  
int y = Math.abs(a);  
int z = Math.min(a, b);
```

Κλάση Greenfoot

Παραγωγή τυχαίων αριθμών:

```
x =  
Greenfoot.getRandomNumber(limit)  
;
```



Εργαστηριακή Άσκηση 10: Τυχαία Συμπεριφορά

Προσθέτουμε τυχαία συμπεριφορά στην κίνηση των αντικειμένων χρησιμοποιώντας τη στατική μέθοδο `getRandomNumber`.

01

Κλήση Μεθόδου

`Greenfoot.getRandomNumber(100)` παράγει τυχαίους αριθμούς από 0 έως 99

02

Συνθήκη

Αν ο αριθμός είναι μικρότερος του 10 (10% πιθανότητα), ο βάτραχος στρίβει

03

Εκτέλεση

Η μέθοδος `turn(15)` στρίβει τον βάτραχο κατά 15 μοίρες

```
public void act() {  
    move(4);  
    if (Greenfoot.getRandomNumber(100) < 10) {  
        turn(15);  
    }  
    if (isAtEdge()) {  
        turn(5);  
    }  
}
```

Δημιουργία Δικτών μας Μεθόδων

Κατά την ανάπτυξη εφαρμογής, ορισμένα τμήματα κώδικα χρειάζεται να επαναλαμβάνονται πολλές φορές. Μπορούμε να δώσουμε ένα όνομα σε αυτά τα τμήματα κώδικα και να τα καλούμε όποτε τα χρειαζόμαστε.

Βελτίωση Αναγνωσιμότητας

Διαχωρισμός του κώδικα σε μικρότερα, κατανοητά μέρη

Επαναχρησιμοποίηση

Χρήση των ίδιων μεθόδων σε διαφορετικά σημεία του προγράμματος

Διαχείριση

Εύκολη τροποποίηση χωρίς αλλαγή του κύριου κώδικα



Εργαστηριακή Άσκηση 11: Οργάνωση Κώδικα

Δημιουργούμε μεθόδους για κάθε ομάδα εντολών που επιτελεί μια συγκεκριμένη λειτουργία.

1

randomTurn()

Στροφή κατά τυχαία κλίση

2

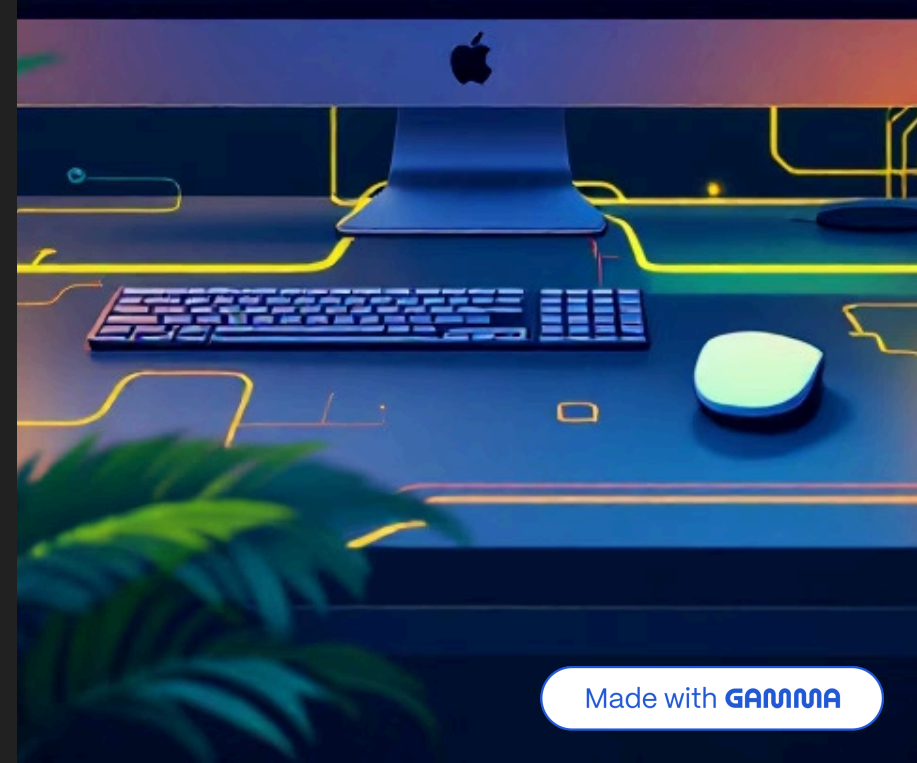
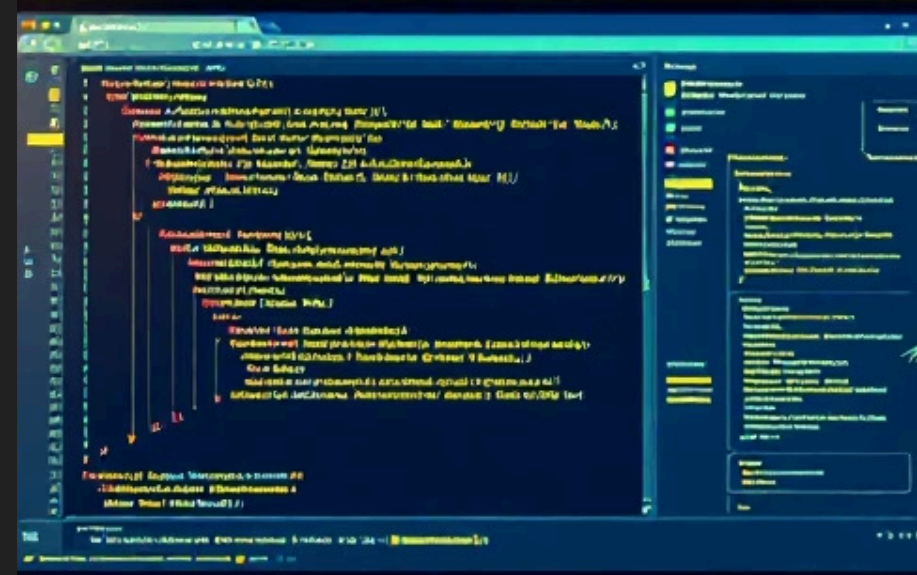
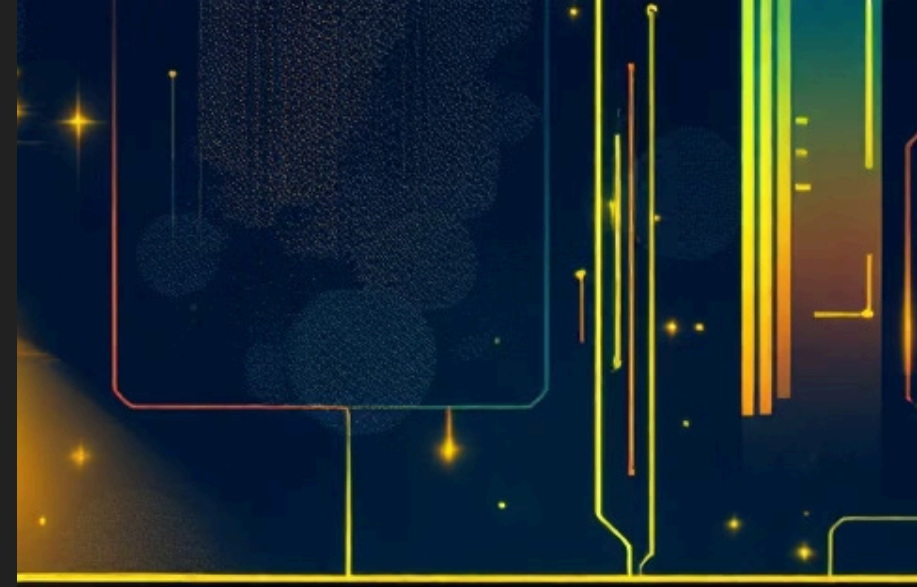
turnAtEdge()

Στροφή στην άκρη του Κόσμου

```
public void act() {  
    move(4);  
    randomTurn();  
    turnAtEdge();  
}
```

```
public void randomTurn() {  
    if (Greenfoot.getRandomNumber(100) < 10) {  
        int angle = Greenfoot.getRandomNumber(20);  
        turn(angle);  
    }  
}
```

```
public void turnAtEdge() {  
    if (isAtEdge()) {  
        turn(5);  
    }  
}
```



Πέρασμα Παραμέτρων

Αντί να δημιουργούμε διαφορετική μέθοδο για κάθε περίπτωση, χρησιμοποιούμε παραμέτρους.

```
public void turnAtEdge(int degrees) {  
    if (isAtEdge()) {  
        turn(degrees);  
    }  
}  
  
public void act() {  
    move(4);  
    for (int i = 0; i < 20; i++) {  
        turnAtEdge(i);  
    }  
}
```

Η μεταβίβαση παραμέτρων στη Java είναι **κατά τιμή** (pass-by-value).

Παράμετροι vs Ορίσματα

Οι μεταβλητές στον ορισμό της μεθόδου ονομάζονται **παραμέτροι**, ενώ οι τιμές που παίρνουν κατά την κλήση ονομάζονται **ορίσματα**.

Αναφορές και Αλληλεπίδραση Αντικειμένων

Όταν η παράμετρος αναφέρεται σε αντικείμενο κλάσης, δεν δημιουργείται αντίγραφο του αντικειμένου, αλλά μια **αναφορά** (reference) στη μνήμη.

1

Ανίχνευση

`isTouching(Fly.class)` ανιχνεύει αν ένα αντικείμενο αγγίζει κάποιο άλλο

2

Εξαφάνιση

`removeTouching(Fly.class)` εξαφανίζει το αντικείμενο

3

Επικοινωνία

Πέρασμα αναφοράς για αλληλεπίδραση



Εργαστηριακή Άσκηση 13: Μετρητής Σκορ

Προσθέτουμε ένα μετρητή για να κρατάμε το σκορ κάθε φορά που ο βάτραχος τρώει μια μύγα.



Μεταβλητή Αντικειμένου

Δήλωση `private int fliesEaten`; στην κλάση `Frog`



Αύξηση Μετρητή

Προσθήκη `fliesEaten++`; στη μέθοδο `eatFlies()`



Εμφάνιση Σκορ

Δημιουργία κλάσης `Counter` με δυναμικό κείμενο

Κλάση Counter: Δημιουργία Μετρητή

Η κλάση Counter επεκτείνει την Actor και χρησιμοποιεί την GreenfootImage για δυναμικό κείμενο.



GreenfootImage

Κλάση για χειρισμό εικόνων με μεθόδους `clear()`, `setColor()` και `drawString()`



Κατασκευαστής

Δημιουργία διαφανούς εικόνας και αρχικοποίηση μεταβλητών



Μέθοδος `update()`

Ανανέωση του κειμένου του μετρητή με τη νέα τιμή

```
public class Counter extends Actor {
    private int points;

    public Counter() {
        GreenfootImage counterImage = new GreenfootImage(200, 30);
        setImage(counterImage);
        points = 0;
        update();
    }

    public void update() {
        GreenfootImage img = getImage();
        img.clear();
        img.setColor(Color.BLACK);
        img.drawString("Σκορ: " + points, 4, 20);
    }

    public void addScore(int value) {
        points += value;
        update();
    }
}
```

Περνώντας την αναφορά του Counter στον κατασκευαστή του Frog, τα δύο αντικείμενα μπορούν να επικοινωνήσουν.