



Πανεπιστήμιο Πειραιώς

Τμήμα Πληροφορικής

Ατομική εργασία μαθήματος

“Αντικειμενοστρεφής Προγραμματισμός”

Ουντράκης Στέφανος

A.M: Π16104

Ημερομηνία: 27/04/2020

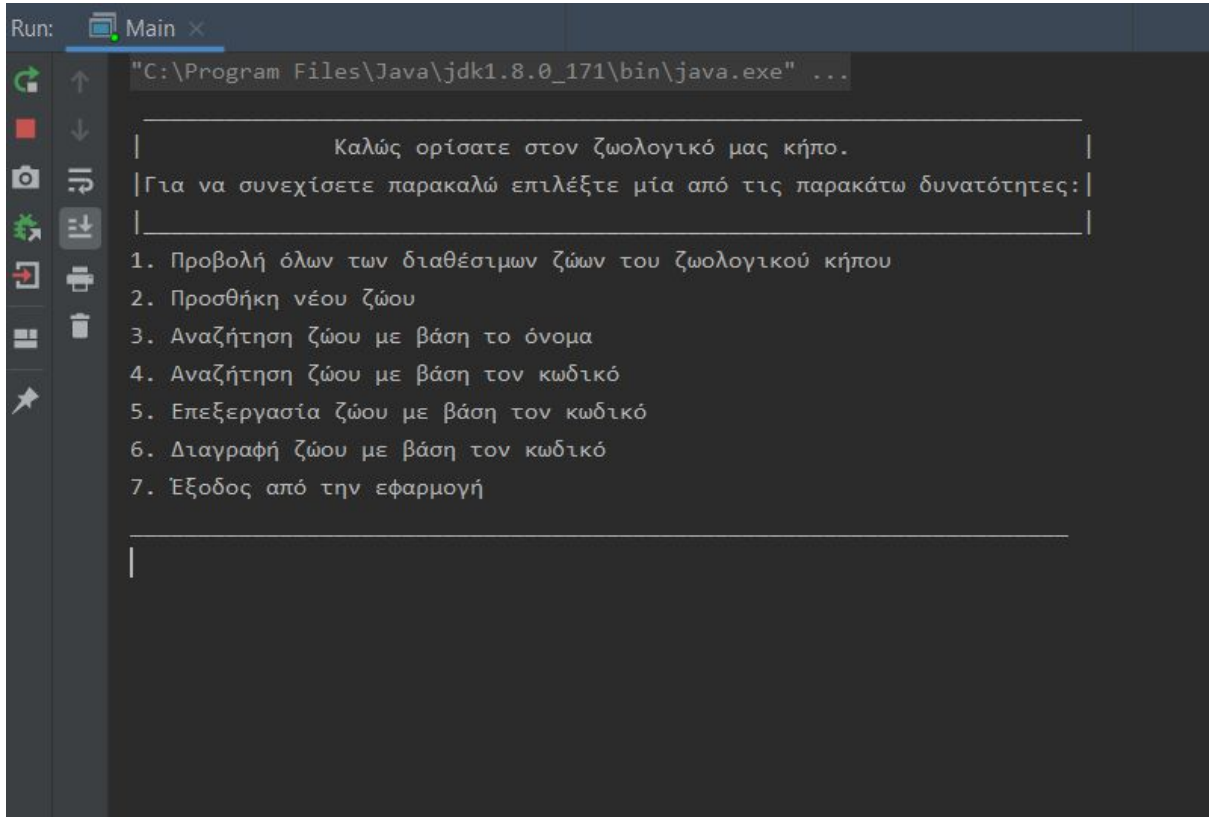
Επεξεργασία: 28/04/2020

Περιεχόμενα

- 1.) Συνοπτική παρουσίαση εφαρμογής.
- 2.) Συνοπτική παρουσίαση κλάσεων.
- 3.) Συνοπτική παρουσίαση κώδικα.

Παρουσίαση Εφαρμογής.

1) Όταν ξεκινήσει η εφαρμογή παρουσιάζει το μενού δυνατοτήτων:



The screenshot shows a Java application window titled "Main". The command prompt at the top shows the execution of "C:\Program Files\Java\jdk1.8.0_171\bin\java.exe". The application displays a menu with the following text:

```
|          Καλώς ορίσατε στον ζωολογικό μας κήπο.          |  
| Για να συνεχίσετε παρακαλώ επιλέξτε μία από τις παρακάτω δυνατότητες: |  
|_____|  
1. Προβολή όλων των διαθέσιμων ζώων του ζωολογικού κήπου  
2. Προσθήκη νέου ζώου  
3. Αναζήτηση ζώου με βάση το όνομα  
4. Αναζήτηση ζώου με βάση τον κωδικό  
5. Επεξεργασία ζώου με βάση τον κωδικό  
6. Διαγραφή ζώου με βάση τον κωδικό  
7. Έξοδος από την εφαρμογή  
|_____|  
|
```

- 2) Αν εισαχθεί το 1, παρουσιάζεται στον χρήστη ο κατάλογος με τα ζώα που υπάρχουν στο αρχείο μας.

Επιπλέον, να σημειωθεί πως εδώ, όπως και μετά από την εκτέλεση κάθε δυνατότητας εκτός της 7ης, ο χρήστης ερωτάται αν θέλει να γυρίσει στο μενού ή να τερματίσει την εφαρμογή.

```
1
Κωδικός:1
Όνομα:elephant
Ομοταξία:mammals
Βάρος:8000.0
Μέγιστη ηλικία:80
-----
Κωδικός:2
Όνομα:elephant
Ομοταξία:mammals
Βάρος:7800.0
Μέγιστη ηλικία:80
-----
Κωδικός:3
Όνομα:crocodile
Ομοταξία:serptiles
Βάρος:1000.0
Μέγιστη ηλικία:70
-----
Κωδικός:4
Όνομα:crocodile
Ομοταξία:serptiles
Βάρος:900.0
Μέγιστη ηλικία:70
-----
Κωδικός:5
Όνομα:bear
Ομοταξία:mammals
Βάρος:250.0
Μέγιστη ηλικία:25
-----
Κωδικός:7
Όνομα:giraffe
Ομοταξία:mammals
Βάρος:2000.0
Μέγιστη ηλικία:80
-----
Θέλετε να γυρίσετε πίσω στο αρχικό μενού? y/n
```

Στη συνέχεια με την εισαγωγή του 2, ο χρήστης έχει την δυνατότητα να προσθέσει στο αρχείο ένα νέο ζώο.

```
Θέλετε να γυρίσετε πίσω στο αρχικό μενού? y/n
y
|
| Καλώς ορίσατε στον ζωολογικό μας κήπο.
| Για να συνεχίσετε παρακαλώ επιλέξτε μία από τις παρακάτω δυνατότητες:|
|
1. Προβολή όλων των διαθέσιμων ζώων του ζωολογικού κήπου
2. Προσθήκη νέου ζώου
3. Αναζήτηση ζώου με βάση το όνομα
4. Αναζήτηση ζώου με βάση τον κωδικό
5. Επεξεργασία ζώου με βάση τον κωδικό
6. Διαγραφή ζώου με βάση τον κωδικό
7. Έξοδος από την εφαρμογή
|
2
Συμπληρώστε Κωδικό. Η τιμή πρέπει να είναι θετικός ακέραιος αριθμός, μικρότερο των 4 ψηφίων και ΜΟΝΑΔΙΚΟΣ.
8
Συμπληρώστε ένα όνομα
elephant
Συμπληρώστε μια ομοταξία
mammals
Συμπληρώστε βάρος. Το βάρος πρέπει να είναι θετικό, μεγαλύτερο από 0.1 κιλά και μικρότερο από 10000 κιλά
8000
Συμπληρώστε μέγιστη ηλικία. Η ηλικία πρέπει να είναι θετικός ακέραιος αριθμός, μεγαλύτερος από 0 και μικρότερος από 100
80
Επιτυχής ενέργεια!
Θέλετε να γυρίσετε πίσω στο αρχικό μενού? y/n
|
```

Έπειτα πληκτρολογούμε το 3 για την εύρεση ζώου με συγκεκριμένο όνομα (αποτελέσματα μπορεί να είναι και >1).

```
|           Καλώς ορίσατε στον ζωολογικό μας κήπο.           |
| Για να συνεχίσετε παρακαλώ επιλέξτε μία από τις παρακάτω δυνατότητες: |
|_____|
1. Προβολή όλων των διαθέσιμων ζώων του ζωολογικού κήπου
2. Προσθήκη νέου ζώου
3. Αναζήτηση ζώου με βάση το όνομα
4. Αναζήτηση ζώου με βάση τον κωδικό
5. Επεξεργασία ζώου με βάση τον κωδικό
6. Διαγραφή ζώου με βάση τον κωδικό
7. Έξοδος από την εφαρμογή

3
Αναζήτηση ζώου με βάση το όνομα:
elephant

Κωδικός:1
Όνομα:elephant
Ομοταξία:mammals
Βάρος:8000.0
Μέγιστη ηλικία:80
-----

Κωδικός:2
Όνομα:elephant
Ομοταξία:mammals
Βάρος:7800.0
Μέγιστη ηλικία:80
-----

Κωδικός:8
Όνομα:elephant
Ομοταξία:mammals
Βάρος:8000.0
Μέγιστη ηλικία:80
-----

Θέλετε να γυρίσετε πίσω στο αρχικό μενού? y/n
|
```

Αν πληκτρολογηθεί το 4, ο χρήστης μπορεί να δει το ζώο με τον συγκεκριμένο κωδικό. (Το αποτέλεσμα είναι μοναδικό αν και εφόσον υπάρχει κωδικός, λόγω των ελέγχων που “αναγκάζουν” την μοναδική ύπαρξη κωδικού).

```
Θέλετε να γυρίσετε πίσω στο αρχικό μενού? y/n
y
|
| Καλώς ορίσατε στον ζωολογικό μας κήπο.
| Για να συνεχίσετε παρακαλώ επιλέξτε μία από τις παρακάτω δυνατότητες: |
|
1. Προβολή όλων των διαθέσιμων ζώων του ζωολογικού κήπου
2. Προσθήκη νέου ζώου
3. Αναζήτηση ζώου με βάση το όνομα
4. Αναζήτηση ζώου με βάση τον κωδικό
5. Επεξεργασία ζώου με βάση τον κωδικό
6. Διαγραφή ζώου με βάση τον κωδικό
7. Έξοδος από την εφαρμογή

4
Αναζήτηση ζώου με βάση τον κωδικό:
3
Κωδικός:3
Όνομα:crocodile
Ομοταξία:serptiles
Βάρος:1000.0
Μέγιστη ηλικία:70

Θέλετε να γυρίσετε πίσω στο αρχικό μενού? y/n
|
```

Με το 5 ο χρήστης επεξεργάζεται ένα ήδη υπάρχων ζώο.

```
Θέλετε να γυρίσετε πίσω στο αρχικό μενού? y/n
y

| Καλώς ορίσατε στον ζωολογικό μας κήπο. |
| Για να συνεχίσετε παρακαλώ επιλέξτε μία από τις παρακάτω δυνατότητες: |
|-----|
1. Προβολή όλων των διαθέσιμων ζώων του ζωολογικού κήπου
2. Προσθήκη νέου ζώου
3. Αναζήτηση ζώου με βάση το όνομα
4. Αναζήτηση ζώου με βάση τον κωδικό
5. Επεξεργασία ζώου με βάση τον κωδικό
6. Διαγραφή ζώου με βάση τον κωδικό
7. Έξοδος από την εφαρμογή

5
Συμπληρώστε Κωδικό. Η τιμή πρέπει να είναι θετικός ακέραιος αριθμός, μικρότερο
8
Συμπληρώστε ένα όνομα
bird
Συμπληρώστε μια ομοταξία
dontknow
Συμπληρώστε βάρος. Το βάρος πρέπει να είναι θετικό, μεγαλύτερο από 0.1 κιλά και
0.2
Συμπληρώστε μέγιστη ηλικία. Η ηλικία πρέπει να είναι θετικός ακέραιος αριθμός
1
Επιτυχής ενέργεια!
Θέλετε να γυρίσετε πίσω στο αρχικό μενού? y/n
|
```



```
Ομοταξία:serptiles
Βάρος:1000.0
Μέγιστη ηλικία:70
-----
Κωδικός:4
Όνομα:crocodile
Ομοταξία:serptiles
Βάρος:900.0
Μέγιστη ηλικία:70
-----
Κωδικός:5
Όνομα:bear
Ομοταξία:mammals
Βάρος:250.0
Μέγιστη ηλικία:25
-----
Κωδικός:7
Όνομα:giraffe
Ομοταξία:mammals
Βάρος:2000.0
Μέγιστη ηλικία:80
-----
Κωδικός:8
Όνομα:bird
Ομοταξία:dontknow
Βάρος:0.2
Μέγιστη ηλικία:1
-----
Θέλετε να γυρίσετε πίσω στο αρχικό μενού? y/n
```

Με το 6, διαγράφουμε όποιο ζώο θέλουμε , αρκεί να υπάρχει ο κωδικός..

Θέλετε να γυρίσετε πίσω στο αρχικό μενού? y/n

y

Καλώς φέρεστε στον ζωολογικό μας κήπο.

| Για να συνεχίσετε παρακαλώ επιλέξτε μία από τις παρακάτω δυνατότητες: |

1. Προβολή όλων των διαθέσιμων ζώων του ζωολογικού κήπου
2. Προσθήκη νέου ζώου
3. Αναζήτηση ζώου με βάση το όνομα
4. Αναζήτηση ζώου με βάση τον κωδικό
5. Επεξεργασία ζώου με βάση τον κωδικό
6. Διαγραφή ζώου με βάση τον κωδικό
7. Έξοδος από την εφαρμογή

6

Διαγράψτε ζώο με βάση τον κωδικό:

8

Επιτυχής ενέργεια!

Θέλετε να γυρίσετε πίσω στο αρχικό μενού? y/n

1

Κωδικός:1
Όνομα:elephant
Ομοταξία:mammals
Βάρος:8000.0
Μέγιστη ηλικία:80

Κωδικός:2
Όνομα:elephant
Ομοταξία:mammals
Βάρος:7800.0
Μέγιστη ηλικία:80

Κωδικός:3
Όνομα:crocodile
Ομοταξία:serptiles
Βάρος:1000.0
Μέγιστη ηλικία:70

Κωδικός:4
Όνομα:crocodile
Ομοταξία:serptiles
Βάρος:900.0
Μέγιστη ηλικία:70

Κωδικός:5
Όνομα:bear
Ομοταξία:mammals
Βάρος:250.0
Μέγιστη ηλικία:25

Κωδικός:7
Όνομα:giraffe
Ομοταξία:mammals
Βάρος:2000.0
Μέγιστη ηλικία:80

Θέλετε να γυρίσετε πίσω στο αρχικό μενού? y/n

|

Τέλος με το 7 (όπως και με την επιλογή n) η εφαρμογή κλείνει.

```
| Καλώς ορίσατε στον ζωολογικό μας κήπο. |
| Για να συνεχίσετε παρακαλώ επιλέξτε μία από τις παρακάτω δυνατότητες: |
|-----|
1. Προβολή όλων των διαθέσιμων ζώων του ζωολογικού κήπου
2. Προσθήκη νέου ζώου
3. Αναζήτηση ζώου με βάση το όνομα
4. Αναζήτηση ζώου με βάση τον κωδικό
5. Επεξεργασία ζώου με βάση τον κωδικό
6. Διαγραφή ζώου με βάση τον κωδικό
7. Έξοδος από την εφαρμογή

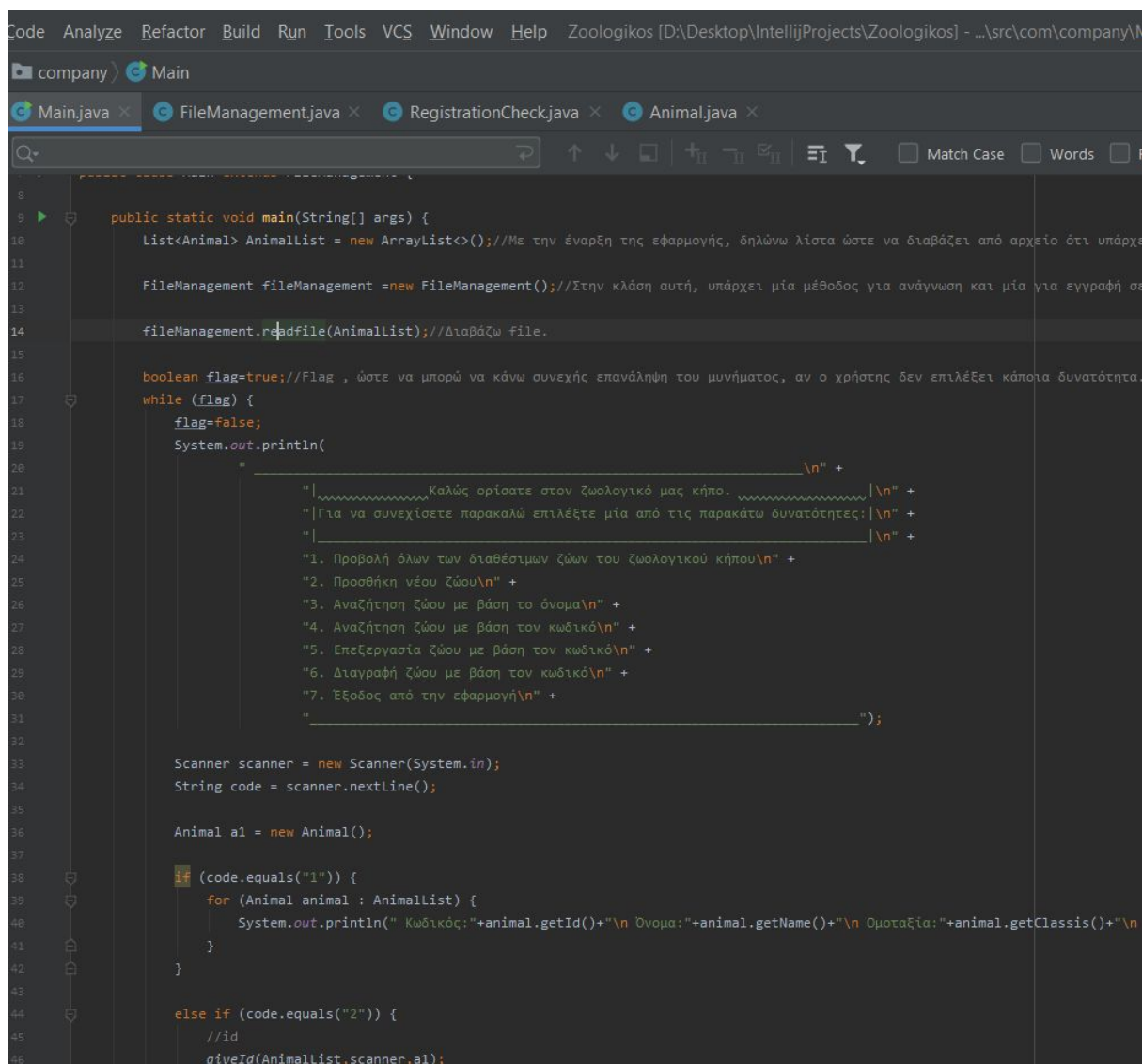
7
Πραγματοποιήθηκε έξοδος από το σύστημα.
Ευχαριστούμε για την χρήση της εφαρμογής!

Process finished with exit code 0
|
```

Κλάσεις

Χρησιμοποιώ 4 κλάσεις για την εκτέλεση του κώδικα:

- 1) Την main, η οποία περιέχει όλο τον κώδικα και εκτελεί την εργασία..



```
Code Analyze Refactor Build Run Tools VCS Window Help Zoologikos [D:\Desktop\IntelliJProjects\Zoologikos] - ...src\com\company\l
company > Main
Main.java x FileManagement.java x RegistrationCheck.java x Animal.java x
Q-
8
9 public static void main(String[] args) {
10     List<Animal> AnimalList = new ArrayList<>();//Με την έναρξη της εφαρμογής, δηλώνω λίστα ώστε να διαβάζει από αρχείο ότι υπάρχει
11
12     FileManagement fileManagement = new FileManagement();//Στην κλάση αυτή, υπάρχει μία μέθοδος για ανάγνωση και μία για εγγραφή σε
13
14     fileManagement.readFile(AnimalList);//Διαβάζω file.
15
16     boolean flag=true;//Flag , ώστε να μπορώ να κάνω συνεχής επανάληψη του μηνύματος, αν ο χρήστης δεν επιλέξει κάποια δυνατότητα.
17     while (flag) {
18         flag=false;
19         System.out.println(
20             "
21             ~~~~~Καλώς ορίσατε στον ζωολογικό μας κήπο. ~~~~~\n" +
22             "|Για να συνεχίσετε παρακαλώ επιλέξτε μία από τις παρακάτω δυνατότητες:\n" +
23             "|~~~~~\n" +
24             "1. Προβολή όλων των διαθέσιμων ζώων του ζωολογικού κήπου\n" +
25             "2. Προσθήκη νέου ζώου\n" +
26             "3. Αναζήτηση ζώου με βάση το όνομα\n" +
27             "4. Αναζήτηση ζώου με βάση τον κωδικό\n" +
28             "5. Επεξεργασία ζώου με βάση τον κωδικό\n" +
29             "6. Διαγραφή ζώου με βάση τον κωδικό\n" +
30             "7. Έξοδος από την εφαρμογή\n" +
31             "~~~~~");
32
33         Scanner scanner = new Scanner(System.in);
34         String code = scanner.nextLine();
35
36         Animal a1 = new Animal();
37
38         if (code.equals("1")) {
39             for (Animal animal : AnimalList) {
40                 System.out.println(" Κωδικός:"+animal.getId()+" Όνομα:"+animal.getName()+" Όμοιαξία:"+animal.getClassis()+"\n");
41             }
42         }
43
44         else if (code.equals("2")) {
45             //id
46             giveId(AnimalList,scanner,a1);
47         }
48     }
49 }
```

- 2) Την κλάση Animal όπου περιέχει τους απαραίτητους getters and setters για την δημιουργία κάποιου ζώου ώστε να αποθηκευτεί στη συνέχεια σε λίστα και σε αρχείο.

```
1 package com.company;
2
3 import java.io.Serializable;
4
5 public class Animal implements Serializable {
6     private int id,max_age;
7     private String name,classis;
8     private double weight;
9
10    public int getId() {
11        return id;
12    }
13
14    public void setId(int id) { this.id = id; }
15
16
17
18    public int getMax_age() { return max_age; }
19
20
21
22    public void setMax_age(int max_age) {
23        this.max_age = max_age;
24    }
25
26    public String getName() { return name; }
27
28
29
30    public void setName(String name) { this.name = name; }
31
32
33
34    public String getClassis() { return classis; }
35
36
37
38    public void setClassis(String classis) { this.classis = classis; }
39
40
41
42    public double getWeight() { return weight; }
43
44
45
46    public void setWeight(double weight) { this.weight = weight; }
47
48
49 }
```

3) Την κλάση FileManagement η οποία περιέχει 2 μεθόδους για το διάβασμα από αρχείο και την εγγραφή σε αυτό.

```
Main.java × Animal.java × FileManagement.java × RegistrationCheck.java ×
3  import java.io.*;
4  import java.util.List;
5
6  public class FileManagement {
7
8      public void readfile(List<Animal> animallist){
9
10         try {
11             FileInputStream fileInputStream = new FileInputStream( name: "myzoo.ser");
12             ObjectInputStream objectInputStream = new ObjectInputStream(fileInputStream);
13             boolean revise=true; // Διαβάζω αρχείο μέχρι να τελειώσουν τα αποθηκευμένα objects.
14             while (revise){
15                 Animal a2=(Animal) objectInputStream.readObject();
16                 if(a2 != null){
17                     animallist.add(a2);
18                 }
19                 else {
20                     revise=false;
21                     objectInputStream.close();
22                     fileInputStream.close();
23                 }
24             }
25         } catch (FileNotFoundException e) {
26             //e.printStackTrace();
27         } catch (IOException e) {
28             // e.printStackTrace();
29         } catch (ClassNotFoundException e) {
30             // e.printStackTrace();
31         }
32     }
33     @ public void WriteFile(List<Animal> Animallist){
34
35         try {
36             FileOutputStream fileOutputStream = new FileOutputStream( name: "myzoo.ser");
37             ObjectOutputStream objectOutputStream = new ObjectOutputStream(fileOutputStream);
38             for (Animal animal : Animallist) {
39                 objectOutputStream.writeObject(animal);
```


- 4) Τέλος την κλάση RegistrationCheck, η οποία περιέχει όλους τους ελέγχους που πρέπει να γίνουν για την επιτυχή εγγραφή και επεξεργασία κάποιου ζώου.

```
ain.java × Animal.java × FileManagement.java × RegistrationCheck.java ×
import java.util.Scanner;
import java.util.regex.Pattern;

public class RegistrationCheck {

    public static void giveId(List<Animal> AnimalList, Scanner scanner, Animal a1){
        boolean flag_id = true;
        while (flag_id) {
            try {
                System.out.println("Συμπληρώστε Κωδικό. Η τιμή πρέπει να είναι θετικός α
                String input = scanner.nextLine();
                int id = Integer.parseInt(input);
                boolean has_id=false;// Έλεγχος αν ο κωδικός υπάρχει ήδη στον κατάλογο.
                for (Animal animal : AnimalList) {
                    if (id == animal.getId()){
                        has_id=true;
                    }
                }
                if (!has_id){
                    if (id > 0 && id < 1000) {
                        a1.setId(id);
                        flag_id = false;
                    } else {
                        flag_id = true;
                    }
                }
            } catch (NumberFormatException e) {
                flag_id = true;
            }
        }
    }

    public static void giveName(Scanner scanner,Animal a1){
        boolean flag_name = true;
        while(flag_name){
            System.out.println("Συμπληρώστε ένα όνομα");
```


Επεξήγηση Κώδικα

Στην αρχή της εφαρμογής διαβάζω από το αρχείο μου ότι αποθηκευμένο αντικείμενο υπάρχει και το τοποθετώ στην λίστα που έχω δημιουργήσει. Αν δεν υπάρχει κάποιο άλλο αντικείμενο να διαβάσω, βγαίνω από την επανάληψη.

```
public class Main extends FileManagement {  
  
    public static void main(String[] args) {  
        List<Animal> AnimalList = new ArrayList<>(); //Με την  
  
        FileManagement fileManagement = new FileManagement();  
  
        fileManagement.readfile(AnimalList); //Διαβάζω file.
```

```
    public void readfile(List<Animal> animalList){  
  
        try {  
            FileInputStream fileInputStream = new FileInputStream( name: "myzoo.ser");  
            ObjectInputStream objectInputStream = new ObjectInputStream(fileInputStream);  
            boolean revise=true; // Διαβάζω αρχείο μέχρι να τελειώσουν τα αποθηκευμένα ob  
            while (revise){  
                Animal a2=(Animal) objectInputStream.readObject();  
                if(a2 != null){  
                    animalList.add(a2);  
                }  
                else {  
                    revise=false;  
                    objectInputStream.close();  
                    fileInputStream.close();  
                }  
            }  
        } catch (FileNotFoundException e) {  
            //e.printStackTrace();  
        } catch (IOException e) {  
            // e.printStackTrace();  
        } catch (ClassNotFoundException e) {  
            // e.printStackTrace();  
        }  
    }  
}
```

Στη συνέχεια δημιουργώ μία επανάληψη με flag, ώστε να εμφανίζεται το αρχικό μενού μία φορά στην αρχή (boolean flag=true), κάθε φορά που ο χρήστης δεν δίνει την απαιτούμενη είσοδο και έπειτα από κάθε εκτέλεση κάποιας ενέργειας, όπου ο χρήστης έχει την δυνατότητα να δώσει θετική απάντηση όταν ερωτάται για επιστροφή στο μενού επιλογών (εικόνα 2). Συμπλήρωση: Το συγκεκριμένο flag γίνεται αρνητικό αν δοθεί σωστή είσοδος, ώστε να μην εμφανίζεται συνεχώς το μενού.

```
16      boolean flag=true;//Flag , ώστε να μπορώ να κάνω συνεχής επανάληψη του μηνύματος, αν ο χρήστης δ
17      while (flag) {
18          flag=false;
19          System.out.println(
20              "
21              |~~~~~Καλώς ορίσατε στον ζωολογικό μας κήπο. ~~~~~|
22              |Για να συνεχίσετε παρακαλώ επιλέξτε μία από τις παρακάτω δυνατότητες:|
23              |~~~~~|
24              "1. Προβολή όλων των διαθέσιμων ζώων του ζωολογικού κήπου\n" +
25              "2. Προσθήκη νέου ζώου\n" +
26              "3. Αναζήτηση ζώου με βάση το όνομα\n" +
27              "4. Αναζήτηση ζώου με βάση τον κωδικό\n" +
28              "5. Επεξεργασία ζώου με βάση τον κωδικό\n" +
29              "6. Διαγραφή ζώου με βάση τον κωδικό\n" +
30              "7. Έξοδος από την εφαρμογή\n" +
31              "~~~~~");
32
33          Scanner scanner = new Scanner(System.in);
34          String code = scanner.nextLine();
35
36          Animal a1 = new Animal();
37
38          if (code.equals("1")) {
39              for (Animal animal : Animallist) {
40                  System.out.println(" Κωδικός:"+animal.getId()+"\n Όνομα:"+animal.getName()+"\n Ομοταξ
41              }
42          }
43
44          else if (code.equals("2")) {
45              //id
46              giveId(Animallist,scanner,a1);
47              //name
```

```
193
194      //Ερώτηση για να γυρίσω σε μενού η να τερματίσω την εφαρμογή.
195      // Διευκολύνει τον χρήστη να δει πρώτα τα αποτελέσματα και να του εμφανισ
196
197      while (!flag) {
198          System.out.println("Θέλετε να γυρίσετε πίσω στο αρχικό μενού? y/n");
199          String go_back = scanner.nextLine();
200          if (go_back.equals("y")) {
201              flag = true;
202          } else if (go_back.equals("n")) {
203              break;
204          } else {
205              System.out.println("Λάθος είσοδος");
206              flag = false;
207          }
208      }
209  }
210 }
```

Από εδώ και στο εξής αναφέρονται αναλυτικά οι ενέργειες:

- 1) Αν ο χρήστης εισάγει τον αριθμό 1, τότε θα διαβαστεί από την λίστα(η οποία περιέχει τα αποθηκευμένα ζώα) και θα εμφανίσει στον χρήστη όλα τα ζώα με όλα τα στοιχεία τους.

```
37
38         if (code.equals("1")) {
39             for (Animal animal : Animallist) {
40                 System.out.println(" Κωδικός:" + animal.getId() + "\n Ό
41             }
42         }
43
```

- 2) Αν ο χρήστης εισάγει τον αριθμό 2, τότε θα δημιουργήσει μια νέα εγγραφή η οποία ζητάει συγκεκριμένα πράγματα.

```
else if (code.equals("2")) {
    //id
    giveId(Animallist, scanner, a1);
    //name
    giveName(scanner, a1);
    //classis
    giveClassis(scanner, a1);
    //weight
    giveWeight(scanner, a1);
    //max_age
    giveMax_Age(scanner, a1);
    //add to list, write on file:
    Animallist.add(a1);
    fileManagement.WriteFile(Animallist);
}
```

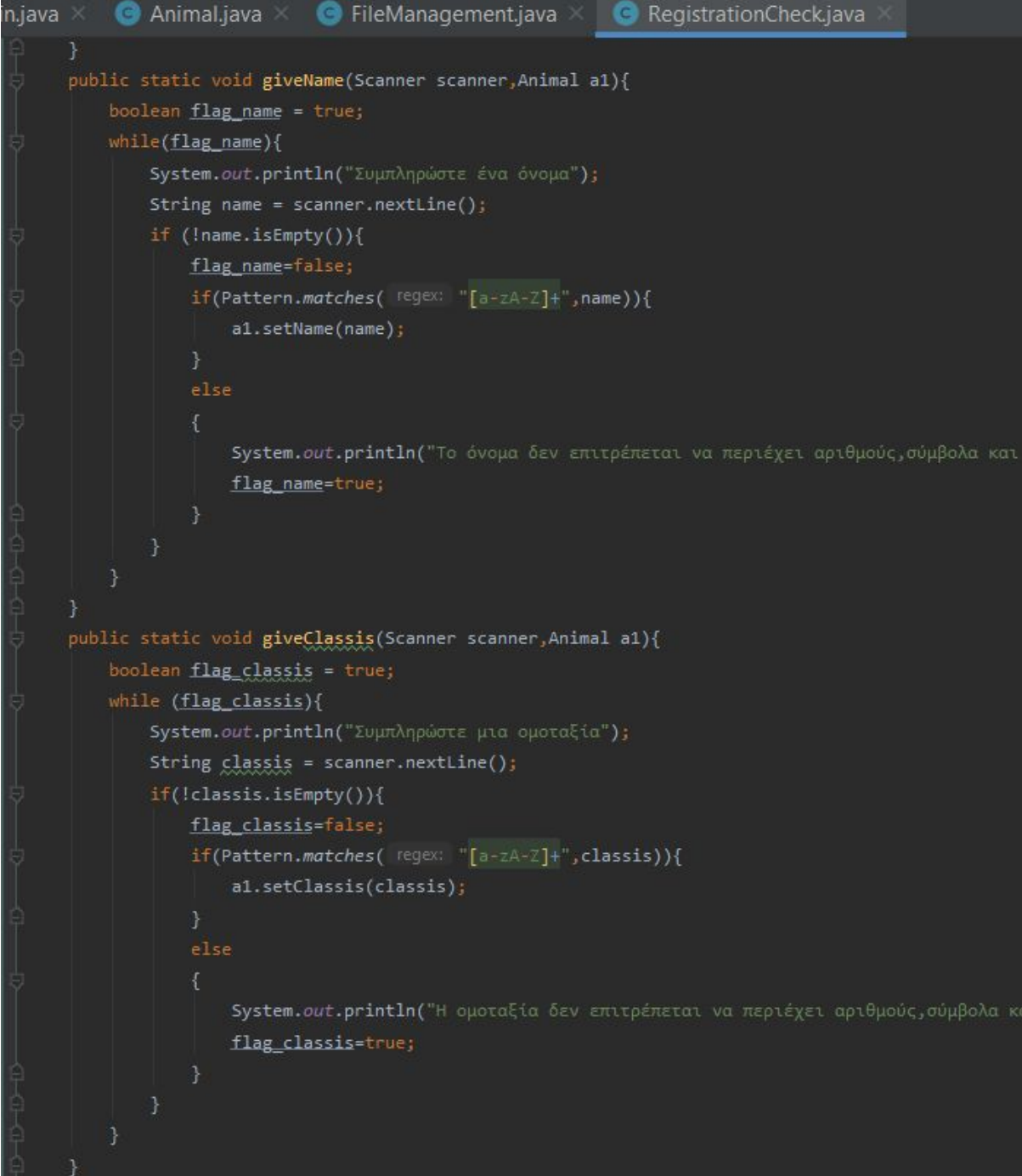
Αρχικά, ζητάει έναν κωδικό ο οποίος ελέγχεται αν είναι θετικός ακέραιος. Έπειτα αν είναι μικρότερος των 4 ψηφίων και κυρίως αν υπάρχει ήδη στην λίστα, ώστε να μην γίνουν 2 ή και παραπάνω εγγραφές με τον ίδιο κωδικό.

```
java × Animal.java × FileManagement.java × RegistrationCheck.java ×
import java.util.Scanner;
import java.util.regex.Pattern;

public class RegistrationCheck {

    public static void giveId(List<Animal> AnimalList, Scanner scanner, Animal a1){
        boolean flag_id = true;
        while (flag_id) {
            try {
                System.out.println("Συμπληρώστε Κωδικό. Η τιμή πρέπει να είναι θετικός");
                String input = scanner.nextLine();
                int id = Integer.parseInt(input);
                boolean has_id=false; // Έλεγχος αν ο κωδικός υπάρχει ήδη στον κατάλογο
                for (Animal animal : AnimalList) {
                    if (id == animal.getId()){
                        has_id=true;
                    }
                }
                if (!has_id){
                    if (id > 0 && id < 1000) {
                        a1.setId(id);
                        flag_id = false;
                    } else {
                        flag_id = true;
                    }
                }
                else
                {
                    System.out.println("Ο κωδικός υπάρχει ήδη στον κατάλογο, παρακαλώ");
                }
            } catch (NumberFormatException e) {
                flag_id = true;
            }
        }
    }
}
```

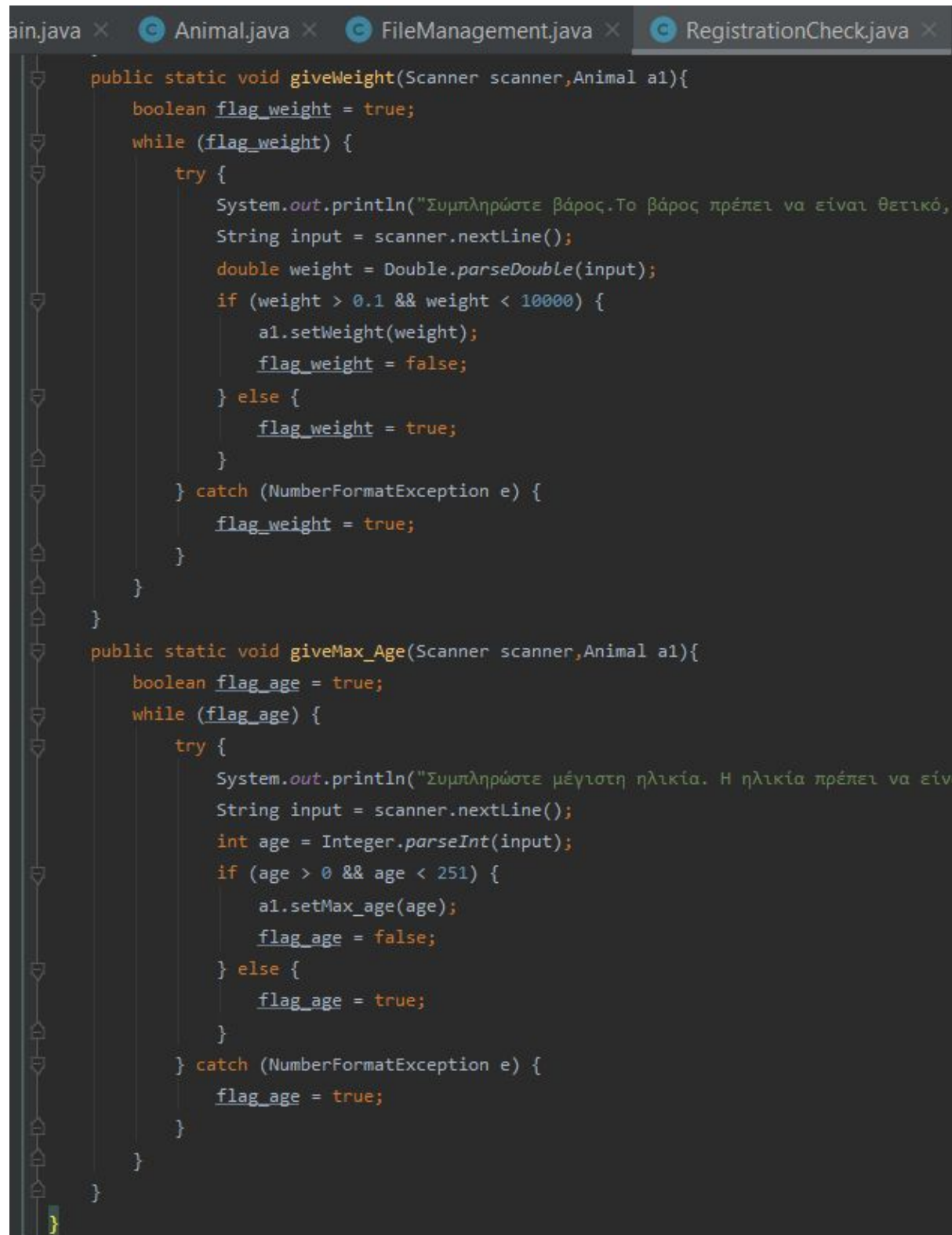
Στη συνέχεια ζητείται όνομα και ομοταξία, όπου αφήνουμε τον χρήστη να βάλει ότι θέλει, εκτός από αριθμό κενά και σύμβολα. Επίσης δεν επιτρέπεται να είναι null.



```
h.java x Animal.java x FileManagement.java x RegistrationCheck.java x
}
public static void giveName(Scanner scanner,Animal a1){
    boolean flag_name = true;
    while(flag_name){
        System.out.println("Συμπληρώστε ένα όνομα");
        String name = scanner.nextLine();
        if (!name.isEmpty()){
            flag_name=false;
            if(Pattern.matches( regex: "[a-zA-Z]+",name)){
                a1.setName(name);
            }
            else
            {
                System.out.println("Το όνομα δεν επιτρέπεται να περιέχει αριθμούς,σύμβολα και
                flag_name=true;
            }
        }
    }
}

public static void giveClassis(Scanner scanner,Animal a1){
    boolean flag_classis = true;
    while (flag_classis){
        System.out.println("Συμπληρώστε μια ομοταξία");
        String classis = scanner.nextLine();
        if(!classis.isEmpty()){
            flag_classis=false;
            if(Pattern.matches( regex: "[a-zA-Z]+",classis)){
                a1.setClassis(classis);
            }
            else
            {
                System.out.println("Η ομοταξία δεν επιτρέπεται να περιέχει αριθμούς,σύμβολα και
                flag_classis=true;
            }
        }
    }
}
}
```

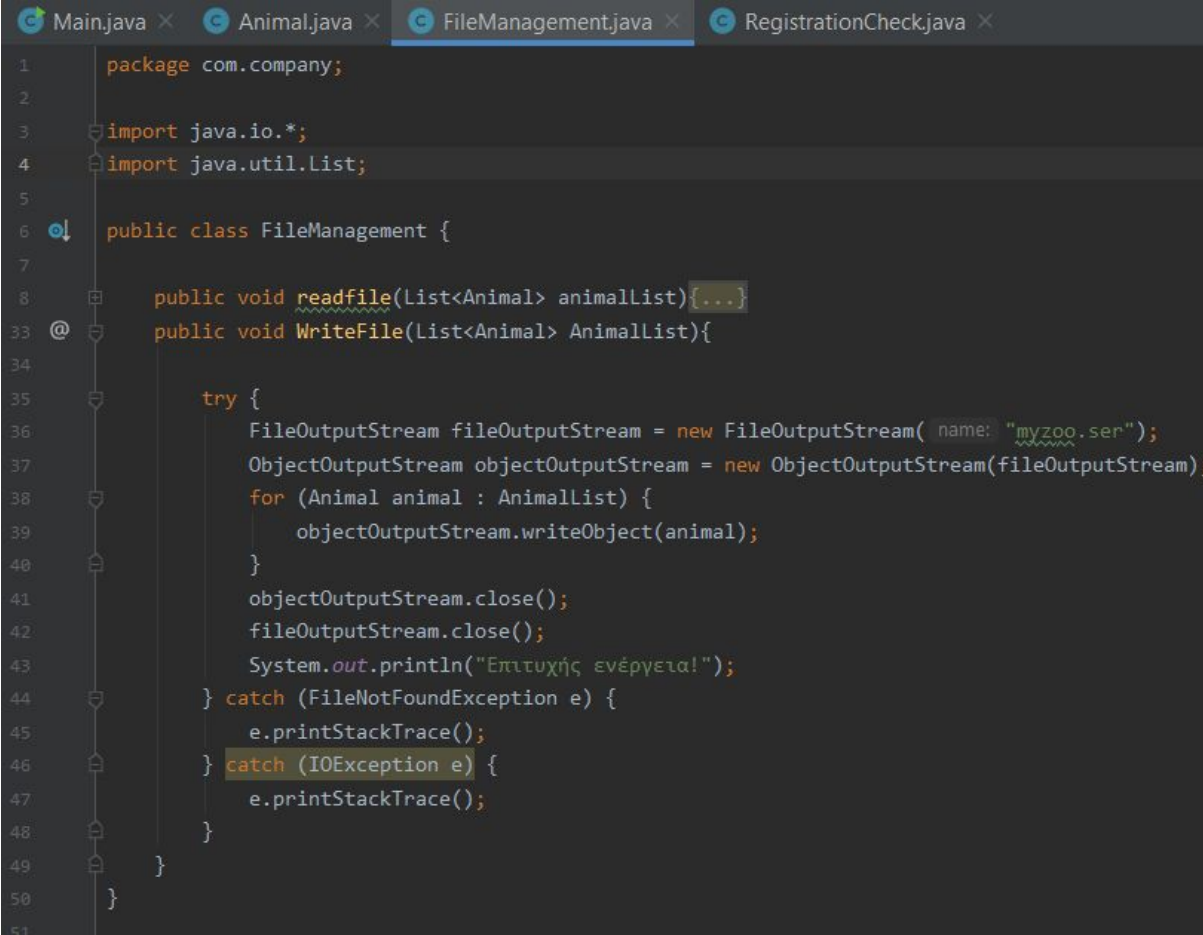

Έπειτα το βάρος και η μέγιστη ηλικία ελέγχονται με παρόμοιο τρόπο όπως τον κωδικό. Η μεγάλη διαφορά είναι ότι δεν ζητάμε μοναδική τιμή και η λογική για το καθένα (όσον αφορά τα όρια) είναι διαφορετικά , και κυρίως για το βάρος αφού είναι δεκαδικός και όχι ακέραιος.

The image shows a screenshot of an IDE with four tabs: 'ain.java', 'Animal.java', 'FileManagement.java', and 'RegistrationCheck.java'. The 'RegistrationCheck.java' tab is active, displaying two methods. The first method, 'giveWeight', prompts the user to enter a weight, validates it against a range of 0.1 to 10000, and sets the weight on an 'Animal' object if valid. The second method, 'giveMax_Age', prompts the user to enter a maximum age, validates it against a range of 0 to 251, and sets the max age on an 'Animal' object if valid. Both methods use a 'while' loop and a 'try-catch' block to handle input errors.

```
public static void giveWeight(Scanner scanner,Animal a1){
    boolean flag_weight = true;
    while (flag_weight) {
        try {
            System.out.println("Συμπληρώστε βάρος.Το βάρος πρέπει να είναι θετικό,");
            String input = scanner.nextLine();
            double weight = Double.parseDouble(input);
            if (weight > 0.1 && weight < 10000) {
                a1.setWeight(weight);
                flag_weight = false;
            } else {
                flag_weight = true;
            }
        } catch (NumberFormatException e) {
            flag_weight = true;
        }
    }
}

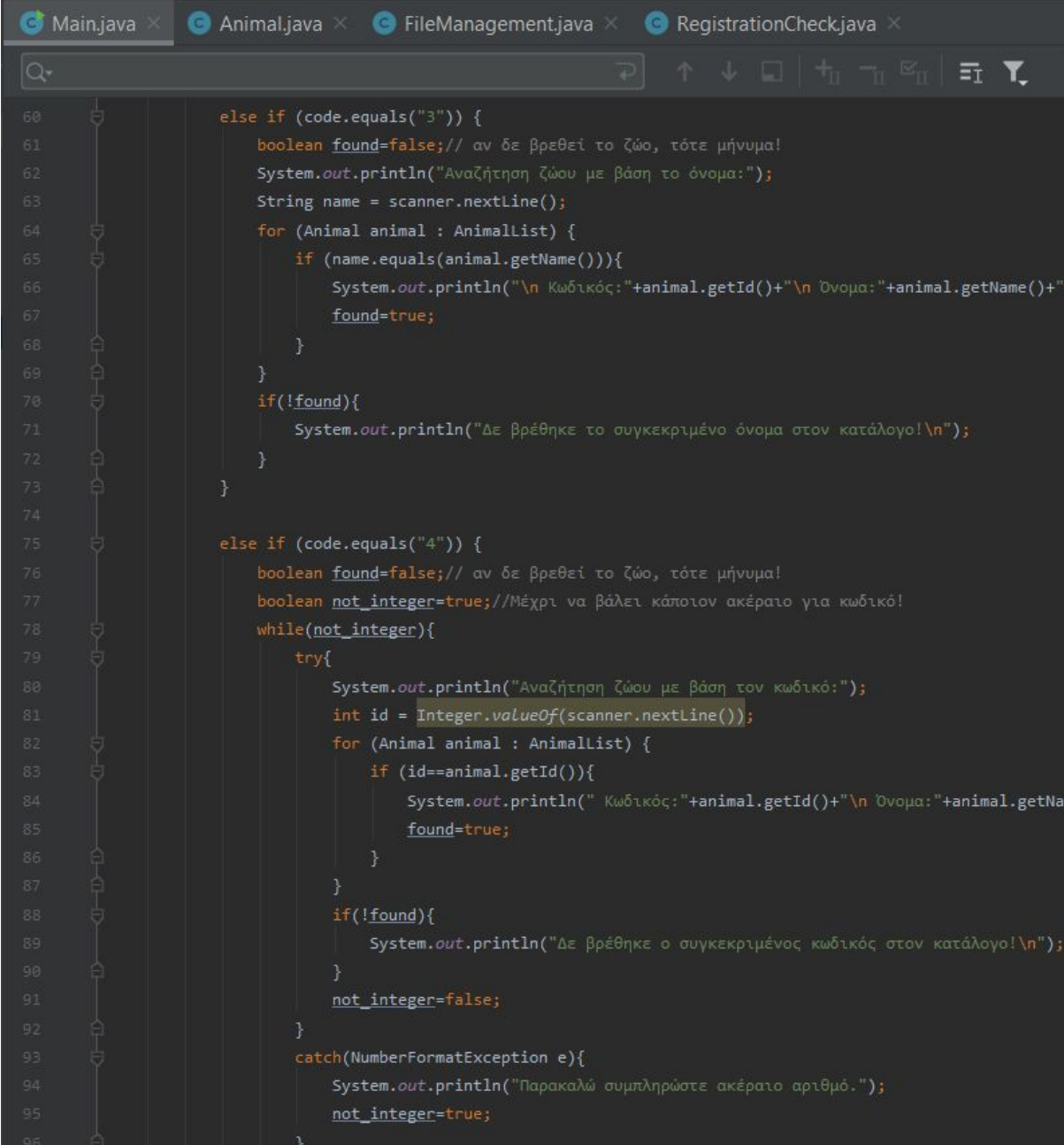
public static void giveMax_Age(Scanner scanner,Animal a1){
    boolean flag_age = true;
    while (flag_age) {
        try {
            System.out.println("Συμπληρώστε μέγιστη ηλικία. Η ηλικία πρέπει να είναι θετική.");
            String input = scanner.nextLine();
            int age = Integer.parseInt(input);
            if (age > 0 && age < 251) {
                a1.setMax_age(age);
                flag_age = false;
            } else {
                flag_age = true;
            }
        } catch (NumberFormatException e) {
            flag_age = true;
        }
    }
}
```

Τέλος , τοποθετώ στη λίστα και στο αρχείο την νέα εγγραφή..



```
1 package com.company;
2
3 import java.io.*;
4 import java.util.List;
5
6 public class FileManagement {
7
8     public void readfile(List<Animal> animallist){...}
9
10    @
11    public void WriteFile(List<Animal> Animallist){
12
13        try {
14            FileOutputStream fileOutputStream = new FileOutputStream( name: "myzoo.ser");
15            ObjectOutputStream objectOutputStream = new ObjectOutputStream(fileOutputStream);
16            for (Animal animal : Animallist) {
17                objectOutputStream.writeObject(animal);
18            }
19            objectOutputStream.close();
20            fileOutputStream.close();
21            System.out.println("Επιτυχής ενέργεια!");
22        } catch (FileNotFoundException e) {
23            e.printStackTrace();
24        } catch (IOException e) {
25            e.printStackTrace();
26        }
27    }
28
29 }
30
31 }
```

3,4) Αν ο χρήστης εισάγει τον αριθμό 3, θα μπορεί να πραγματοποιήσει την αναζήτηση ενός ζώου με βάση το όνομα, ενώ με τον αριθμό 4, με βάση τον κωδικό. Ο κώδικας είναι παρόμοιος με την διαφορά ότι η αναζήτηση με κωδικό έχει περισσότερους ελέγχους και θεωρητικά με το 4 θα βγάζει πάντα ένα αποτέλεσμα, ενώ με το 3 μπορούν να εμφανιστούν πολλαπλά αποτελέσματα.

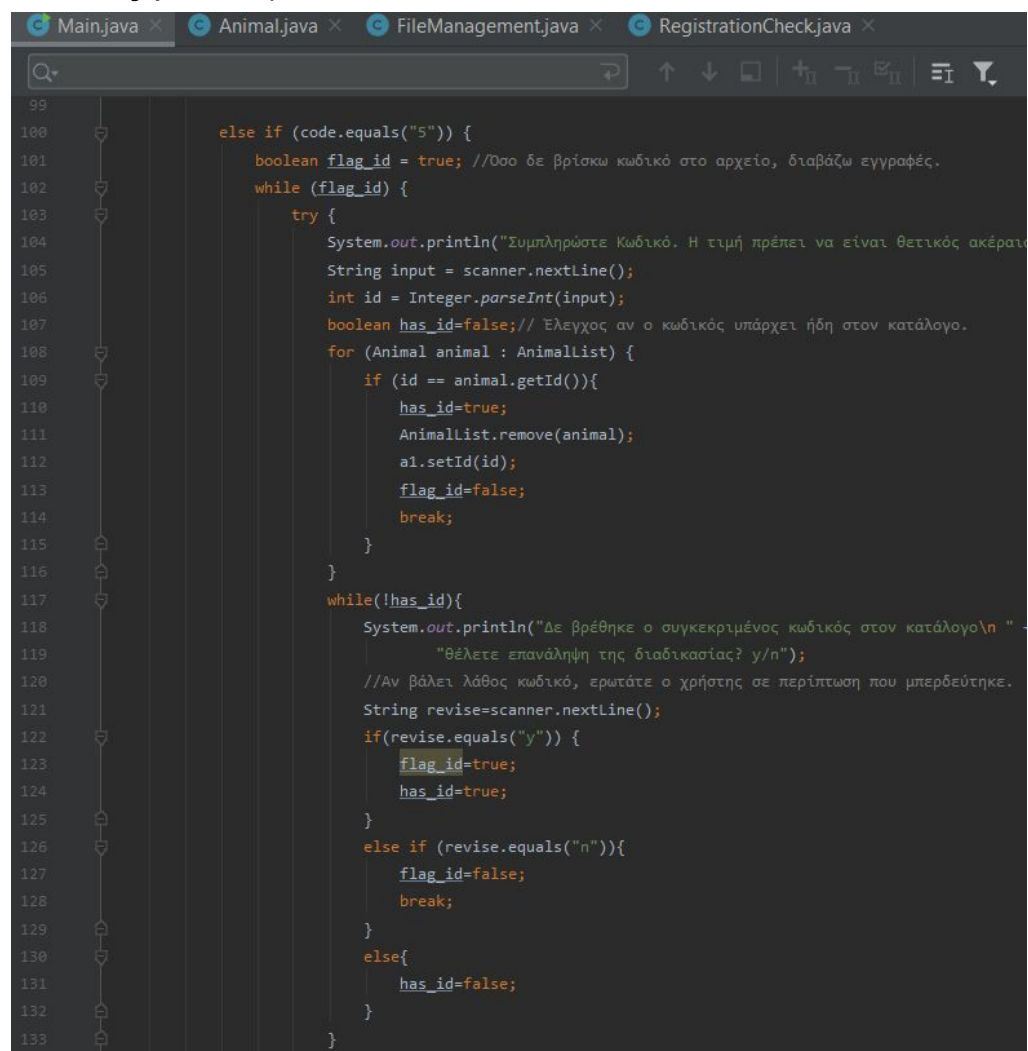


```
60     else if (code.equals("3")) {
61         boolean found=false; // αν δε βρεθεί το ζώο, τότε μήνυμα!
62         System.out.println("Αναζήτηση ζώου με βάση το όνομα:");
63         String name = scanner.nextLine();
64         for (Animal animal : AnimalList) {
65             if (name.equals(animal.getName())){
66                 System.out.println("\n Κωδικός:"+animal.getId()+"\n Όνομα:"+animal.getName());
67                 found=true;
68             }
69         }
70         if(!found){
71             System.out.println("Δε βρέθηκε το συγκεκριμένο όνομα στον κατάλογο!\n");
72         }
73     }
74
75     else if (code.equals("4")) {
76         boolean found=false; // αν δε βρεθεί το ζώο, τότε μήνυμα!
77         boolean not_integer=true; // Μέχρι να βάλει κάποιον ακέραιο για κωδικό!
78         while(not_integer){
79             try{
80                 System.out.println("Αναζήτηση ζώου με βάση τον κωδικό:");
81                 int id = Integer.valueOf(scanner.nextLine());
82                 for (Animal animal : AnimalList) {
83                     if (id==animal.getId()){
84                         System.out.println(" Κωδικός:"+animal.getId()+"\n Όνομα:"+animal.getName());
85                         found=true;
86                     }
87                 }
88                 if(!found){
89                     System.out.println("Δε βρέθηκε ο συγκεκριμένος κωδικός στον κατάλογο!\n");
90                 }
91                 not_integer=false;
92             }
93             catch(NumberFormatException e){
94                 System.out.println("Παρακαλώ συμπληρώστε ακέραιο αριθμό.");
95                 not_integer=true;
96             }
97         }
98     }
```

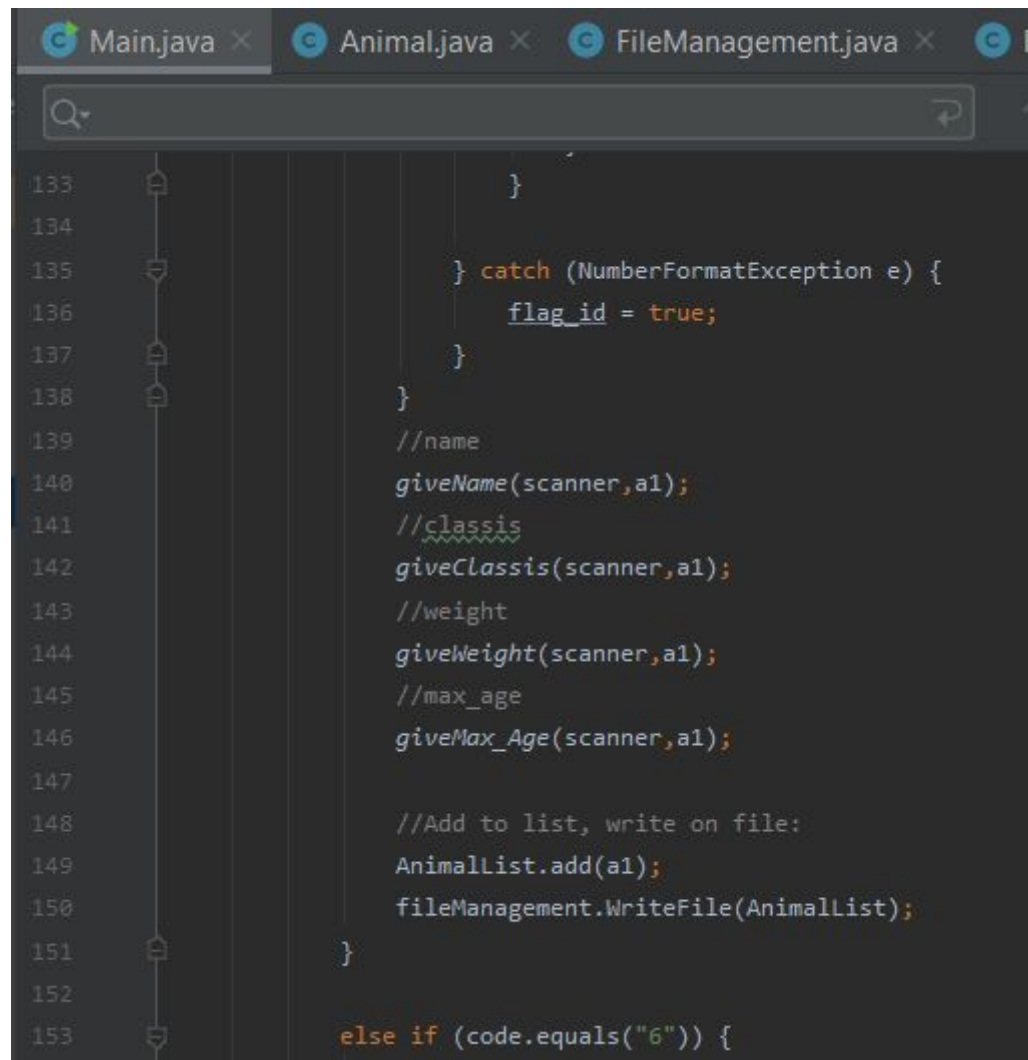

5) Αν ο χρήστης εισάγει τον αριθμό 5 , τότε μπορεί να επεξεργαστεί ένα αντικείμενο του πίνακα με τον συγκεκριμένου κωδικό που επιλέξει.

Η λογική είναι ότι εισάγει τον κωδικό, τον ψάχνουμε και αν τον βρούμε, τον κρατάμε! Σβήνουμε όλο το αντικείμενο που υπάρχει στην λίστα με τον συγκεκριμένου κωδικό, και ζητάμε από τον χρήστη να εισάγει όλα τα υπόλοιπα χαρακτηριστικά όπως γίνεται με το 2. Έτσι δημιουργείται ένα νέο αντικείμενο με τον ίδιο κωδικό.

Δηλαδή όσον αφορά τον κώδικα, έχουμε μια διαγραφή (ακριβώς ίδιος κώδικας στο 6.) και προσθήκη στοιχείο (ίδιος κώδικας με το 2).



```
99
100     else if (code.equals("5")) {
101         boolean flag_id = true; //Όσο δε βρισκω κωδικό στο αρχείο, διαβάζω εγγραφές.
102         while (flag_id) {
103             try {
104                 System.out.println("Συμπληρώστε Κωδικό. Η τιμή πρέπει να είναι θετικός ακέραιος");
105                 String input = scanner.nextLine();
106                 int id = Integer.parseInt(input);
107                 boolean has_id=false; // Έλεγχος αν ο κωδικός υπάρχει ήδη στον κατάλογο.
108                 for (Animal animal : AnimalList) {
109                     if (id == animal.getId()){
110                         has_id=true;
111                         AnimalList.remove(animal);
112                         al.setId(id);
113                         flag_id=false;
114                         break;
115                     }
116                 }
117                 while(!has_id){
118                     System.out.println("Δε βρέθηκε ο συγκεκριμένος κωδικός στον κατάλογο\n "
119                         + "Θέλετε επανάληψη της διαδικασίας? y/n");
120                     //Αν βάλει λάθος κωδικό, ερωτάει ο χρήστης σε περίπτωση που μπερδεύτηκε.
121                     String revise=scanner.nextLine();
122                     if(revise.equals("y")) {
123                         flag_id=true;
124                         has_id=true;
125                     }
126                     else if (revise.equals("n")){
127                         flag_id=false;
128                         break;
129                     }
130                     else{
131                         has_id=false;
132                     }
133                 }
134             }
135         }
```



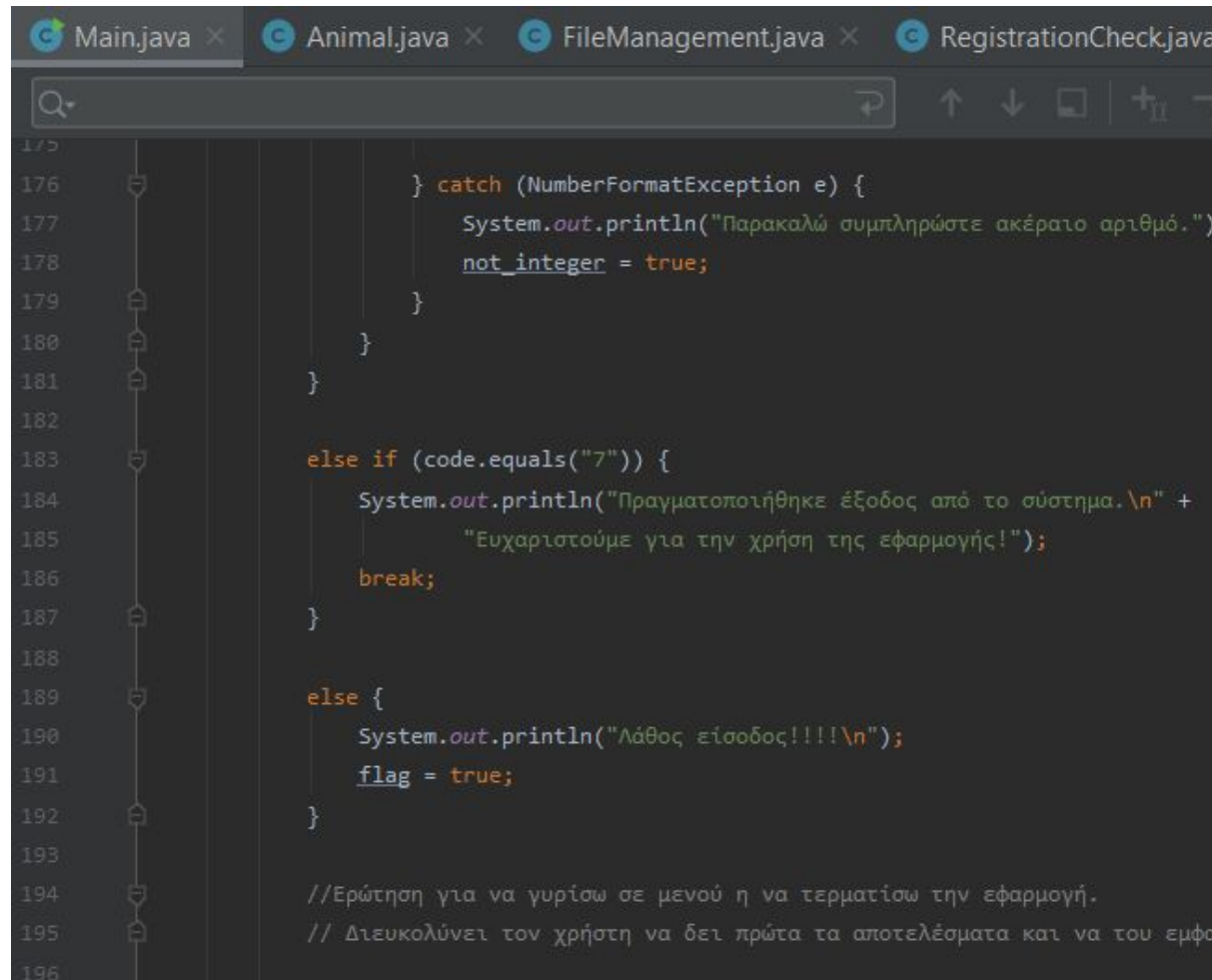
```
133     }
134
135     } catch (NumberFormatException e) {
136         flag_id = true;
137     }
138 }
139 //name
140 giveName(scanner,a1);
141 //classis
142 giveClassis(scanner,a1);
143 //weight
144 giveWeight(scanner,a1);
145 //max_age
146 giveMax_Age(scanner,a1);
147
148 //Add to list, write on file:
149 AnimalList.add(a1);
150 fileManagement.WriteFile(AnimalList);
151 }
152
153 else if (code.equals("6")) {
```

6) Ψάχνουμε το αντικείμενο στην λίστα, αν βρεθεί θα διαγραφεί από τη λίστα και θα αποθηκευτεί στο αρχείο η ενημερωμένη λίστα. Αλλιώς δε θα γίνει κάποια αλλαγή..



```
153     else if (code.equals("6")) {
154
155         boolean found=false;// αν δε βρεθεί το ζώο, τότε
156         boolean not_integer=true;//Μέχρι να βάλει κάποιος
157
158         while(not_integer) {
159             try {
160                 System.out.println("Διαγράψτε ζώο με βάση τον αριθμό:");
161                 String input = scanner.nextLine();
162                 int id = Integer.valueOf(input);
163                 for (Animal animal : AnimalList) {
164                     if (id == animal.getId()) {
165                         AnimalList.remove(animal);
166                         found = true;
167                         break;
168                     }
169                 }
170                 if (!found) {
171                     System.out.println("Δε βρέθηκε ο συγκεκριμένος αριθμός.");
172                 }
173                 not_integer = false;
174                 fileManagement.WriteFile(AnimalList);
175             } catch (NumberFormatException e) {
176                 System.out.println("Παρακαλώ συμπληρώστε έναν αριθμό.");
177                 not_integer = true;
178             }
179         }
180     }
181 }
```

7) Τέλος, αν επιλεγθεί η 7, τότε τερματίζω την λειτουργία απευθείας με ένα break; Για να βγω από την while.



```
175
176         } catch (NumberFormatException e) {
177             System.out.println("Παρακαλώ συμπληρώστε ακέραιο αριθμό.");
178             not_integer = true;
179         }
180     }
181 }
182
183 else if (code.equals("7")) {
184     System.out.println("Πραγματοποιήθηκε έξοδος από το σύστημα.\n" +
185         "Ευχαριστούμε για την χρήση της εφαρμογής!");
186     break;
187 }
188
189 else {
190     System.out.println("Λάθος είσοδος!!!!\n");
191     flag = true;
192 }
193
194 //Ερώτηση για να γυρίσω σε μενού η να τερματίσω την εφαρμογή.
195 // Διευκολύνει τον χρήστη να δει πρώτα τα αποτελέσματα και να του εμφα
196
```