Αναφορά Project Οντοκεντρικού Προγραμματισμού 2020-2021

Στοιχεία Ομάδας:

Εφραίμ Χαριλάου, 1056638, 4ο έτος, up1056638@upnet.gr

Γεωργία – Μαρία Φωτοπούλου, 1059597, 4ο έτος, up1059597@upnet.gr

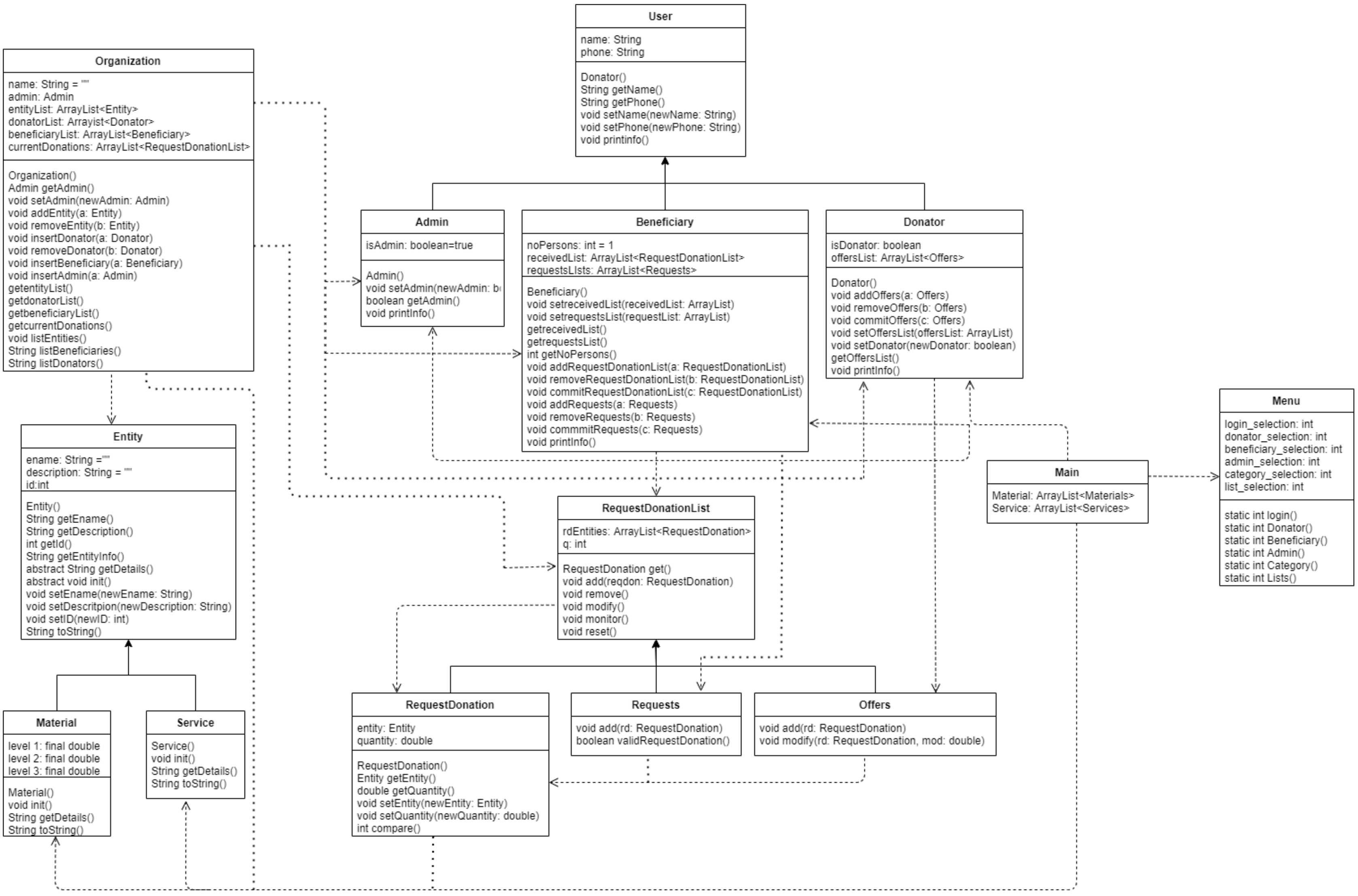
Link του drive στο οποίο περιέχεται ο κώδικας:

https://drive.google.com/drive/folders/14i-YvbOyKKRmCsCzEJk6ab0StDAeFdQx

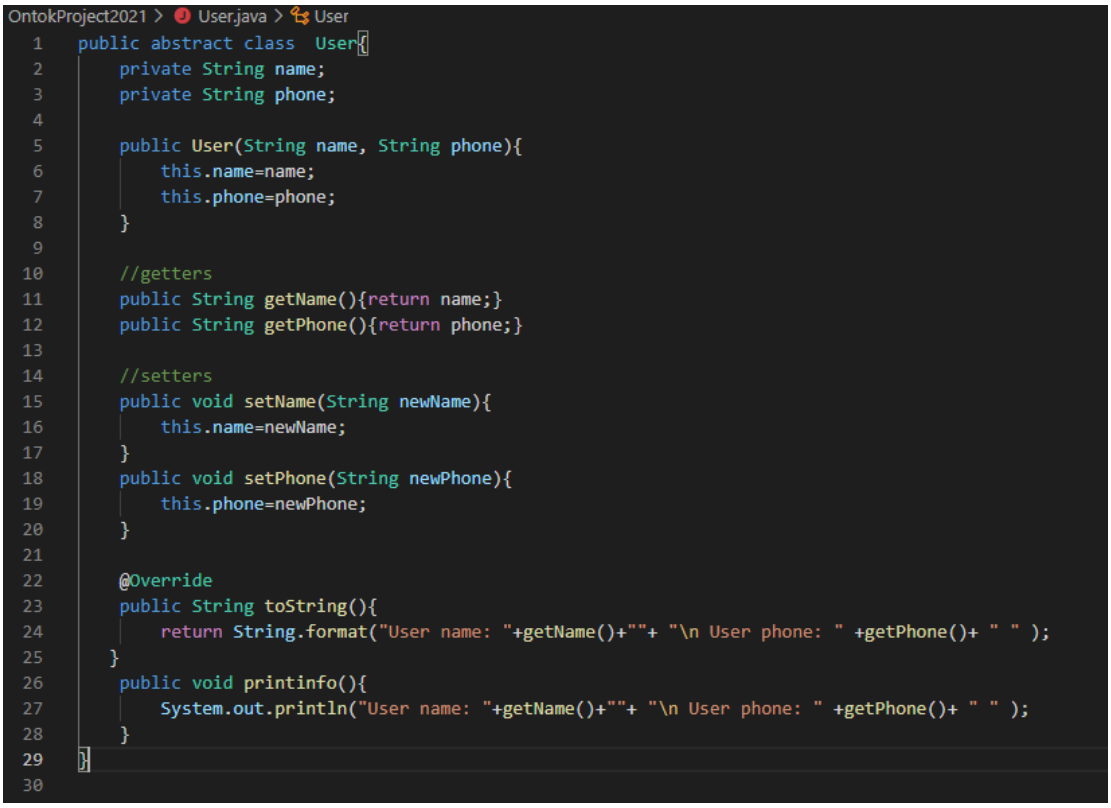
Ολοκληρωμένος κώδικας στο τέλος της αναφοράς.

Το project υλοποιήθηκε σε Java.

Το διάγραμμα κλάσεων UML:


User
name: String
phone: String
Donator()
String getName()
String getPhone()
void setName(newName: String)
void setPhone(newPhone: String)
void printinfo()
Organization
name: String = ""
admin: Admin
entityList: ArrayList<Entity>
donatorList: Arrayist<Donator>
beneficiaryList: ArrayList<Beneficiary>
currentDonations: ArrayList<RequestDonationList>
Organization()
Admin getAdmin()
void setAdmin(newAdmin: Admin)
void addEntity(a: Entity)
void removeEntity(b: Entity)
void insertDonator(a: Donator)
void removeDonator(b: Donator)
void insertBeneficiary(a: Beneficiary)
void insertAdmin(a: Admin)
getentityList()
getdonatorList()
getbeneficiaryList()
getcurrentDonations()
void listEntities()
String listBeneficiaries()
String listDonators()
Admin
isAdmin: boolean=true
Admin()
void setAdmin(newAdmin: b
boolean getAdmin()
void printInfo()
Beneficiary
noPersons: int = 1
receivedList: ArrayList<RequestDonationList>
requestsLIsts: ArrayList<Requests>
Beneficiary()
void setreceivedList(receivedList: ArrayList)
void setrequestsList(requestList: ArrayList)
getreceivedList()
getrequestsList()
int getNoPersons()
void addRequestDonationList(a: RequestDonationList)
void removeRequestDonationList(b: RequestDonationList)
void commitRequestDonationList(c: RequestDonationList)
void addRequests(a: Requests)
void removeRequests(b: Requests)
void commmitRequests(c: Requests)
void printInfo()
Donator
isDonator: boolean
offersList: ArrayList<Offers>
Donator()
void addOffers(a: Offers)
void removeOffers(b: Offers)
void commitOffers(c: Offers)
void setOffersList(offersList: ArrayList)
void setDonator(newDonator: boolean)
getOffersList()
void printInfo()
Menu
login_selection: int
donator_selection: int
beneficiary_selection: int
admin_selection: int
category_selection: int
list_selection: int
static int login()
static int Donator()
static int Beneficiary()
static int Admin()
static int Category()
static int Lists()
Main
Material: ArrayList<Materials>
Service: ArrayList<Services>
Entity
ename: String =""
description: String = ""
id:int
Entity()
String getEname()
String getDescription()
int getId()
String getEntityInfo()
abstract String getDetails()
abstract void init()
void setEname(newEname: String)
void setDescritpion(newDescription: String)
void setID(newID: int)
String toString()
RequestDonationList
rdEntities: ArrayList<RequestDonation>
q: int
RequestDonation get()
void add(reqdon: RequestDonation)
void remove()
void modify()
void monitor()
void reset()
Material
level 1: final double
level 2: final double
level 3: final double
Material()
void init()
String getDetails()
String toString()
Service
Service()
void init()
String getDetails()
String toString()
RequestDonation
entity: Entity
quantity: double
RequestDonation()
Entity getEntity()
double getQuantity()
void setEntity(newEntity: Entity)
void setQuantity(newQuantity: double)
int compare()
Requests
void add(rd: RequestDonation)
boolean validRequestDonation()
Offers
void add(rd: RequestDonation)
void modify(rd: RequestDonation, mod: double)

Στην αρχή του κώδικα υλοποιήσαμε την κλάση **User** η οποία είναι η υπερκλάση των Admin, Beneficiary και Donator. Η User είναι abstract έτσι ώστε να μπορούμε να χρησιμοποιήσουμε μέθοδο που θα δέχεται αντικείμενα υποκλάσεών της. Έχει setters και getters για να μπορούμε να δηλώσουμε και να πάρουμε τιμές αργότερα.

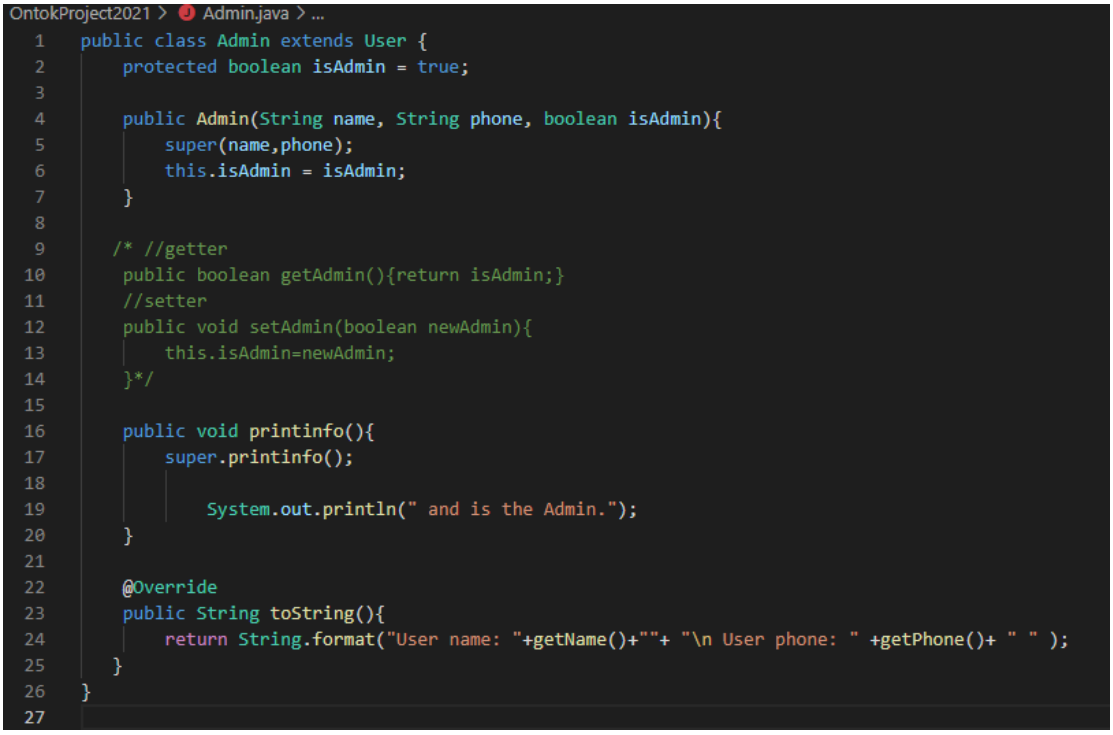


```
public abstract class  User{
    private String name;
    private String phone;

    public User(String name, String phone){
        this.name=name;
        this.phone=phone;
    }

    //getters
    public String getName(){return name;}
    public String getPhone(){return phone;}

    //setters
    public void setName(String newName){
        this.name=newName;
    }
    public void setPhone(String newPhone){
        this.phone=newPhone;
    }

    @Override
    public String toString(){
        return String.format("User name: "+getName()+""+ "\n User phone: " +getPhone()+ " " );
   }
    public void printinfo(){
        System.out.println("User name: "+getName()+""+ "\n User phone: " +getPhone()+ " " );
    }
}

```

Όπως φαίνεται, δηλώνουμε τα στοιχεία του κάθε χρήστη (όνομα και τηλέφωνο) από τα οποία θα αναγνωρίζουμε αν υπάρχει στο πρόγραμμα μας και τι είδος χρήστη είναι. Δηλώσαμε και μέθοδο printinfo() η οποία θα καλείται στη main για να εμφανίζονται μέσω των objects οι υπάρχοντες χρήστες. Πχ., όπως θα δούμε και στην πορεία, αν δηλώσουμε στην Main objects της μορφής Admin adm1 = new Admin (“George”, 6980574755, true) να εμφανίζεται “User name: George

User phone: 6980574755 and is an Admin” κτλ. Υπερκαλύψαμε και μια μέθοδο String toString έτσι ώστε στο τέλος του κώδικα να εμφανίζονται σε πίνακα οι αντίστοιχοι χρήστες με τα στοιχεία τους (Admin, Donators, Beneficiaries).

Έπειτα, σχεδιάζουμε την κλάση του Διαχειρηστή (**Admin**) όπου και καλούμε την printinfo() της υπερκλάσης με διαφορά ότι ελέγχεται εάν ο χρήστης είναι Admin μέσω της Boolean μεταβλητής isAdmin. Επίσης έχουμε και τον constructor ο οποίος καλεί τα δεδομένα της User. Το ίδιο γίνεται και στις υποκλάσεις Beneficiary και Donator.

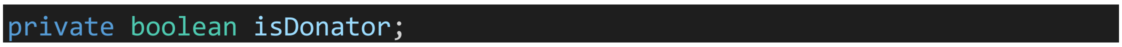


OntokProject2021 > Admin.java > ...

```java
public class Admin extends User {
    protected boolean isAdmin = true;

    public Admin(String name, String phone, boolean isAdmin){
        super(name,phone);
        this.isAdmin = isAdmin;
    }

   /* //getter
    public boolean getAdmin(){return isAdmin;}
    //setter
    public void setAdmin(boolean newAdmin){
        this.isAdmin=newAdmin;
    }*/

    public void printinfo(){
        super.printinfo();

            System.out.println(" and is the Admin.");
    }

    @Override
    public String toString(){
        return String.format("User name: "+getName()+""+ "\n User phone: " +getPhone()+ " " );
   }
}

```

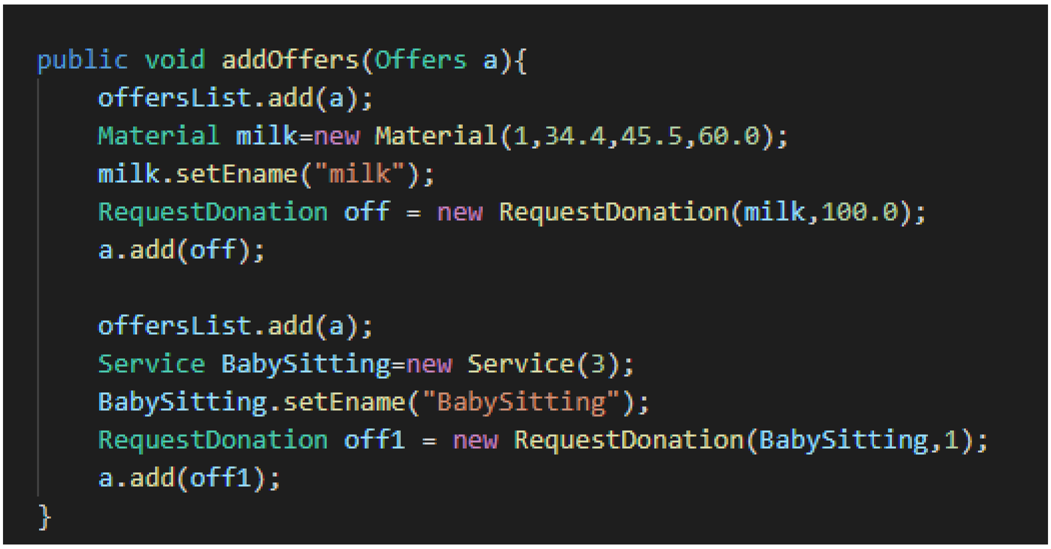
Οι setters/getters της κλάσης δεν χρησιμοποιούνται.

Στη συνέχεια, κατασκευάζοντας την κλάση **Donator** θέτουμε μία έξτρα μεταβλητή



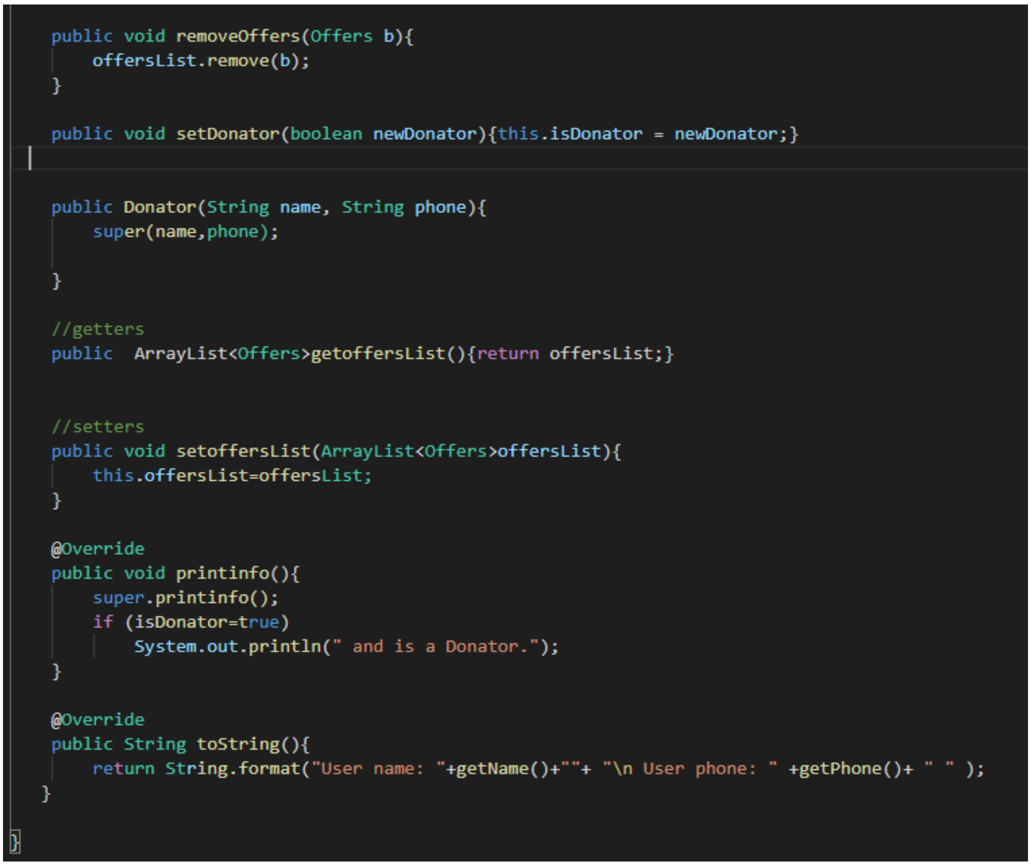
```
private boolean isDonator;
```

την οποία και θα χρησιμοποιήσουμε στη συνέχεια για να εξακριβώσουμε το είδος του χρήστη. Δημιουργούμε την λίστα Offers (που αναφέρεται στην κλάση Offers) η οποία θα περιέχει τα στοιχεία (entities, Materials ή Services) που θέλει να προσφέρει ο Δωρητής. Μέσα στην κλάση, προσθέτουμε wrappers για την ανανέωση της λίστας με κάθε δωρεά και βάζουμε κάποια παραδείγματα υλικών και υπηρεσιών που μπορούν π.χ. να προστεθούν. Π.χ. στην addOffers:



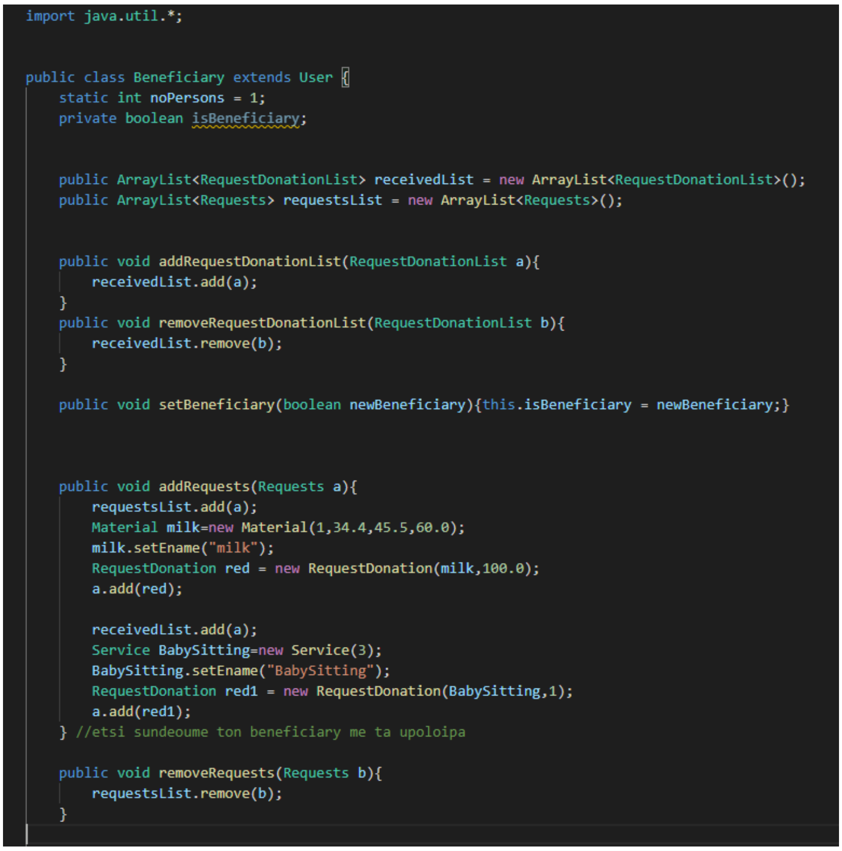
```
public void addOffers(Offers a){
    offersList.add(a);
    Material milk=new Material(1,34.4,45.5,60.0);
    milk.setEname("milk");
    RequestDonation off = new RequestDonation(milk,100.0);
    a.add(off);

    offersList.add(a);
    Service BabySitting=new Service(3);
    BabySitting.setEname("BabySitting");
    RequestDonation off1 = new RequestDonation(BabySitting,1);
    a.add(off1);
}
```

Όπου έχουμε σαν ένα είδος προσφοράς το γάλα (milk) και ένα άλλο την υπηρεσία babysitting. Προχωρώντας, την ίδια τακτική ακολουθούμε στην κλάση του Beneficiary με τη διαφορά ότι πλέον δημιουργούμε λίστες για τις αιτήσεις Requests των beneficiaries με τα αντίστοιχα wrappers.

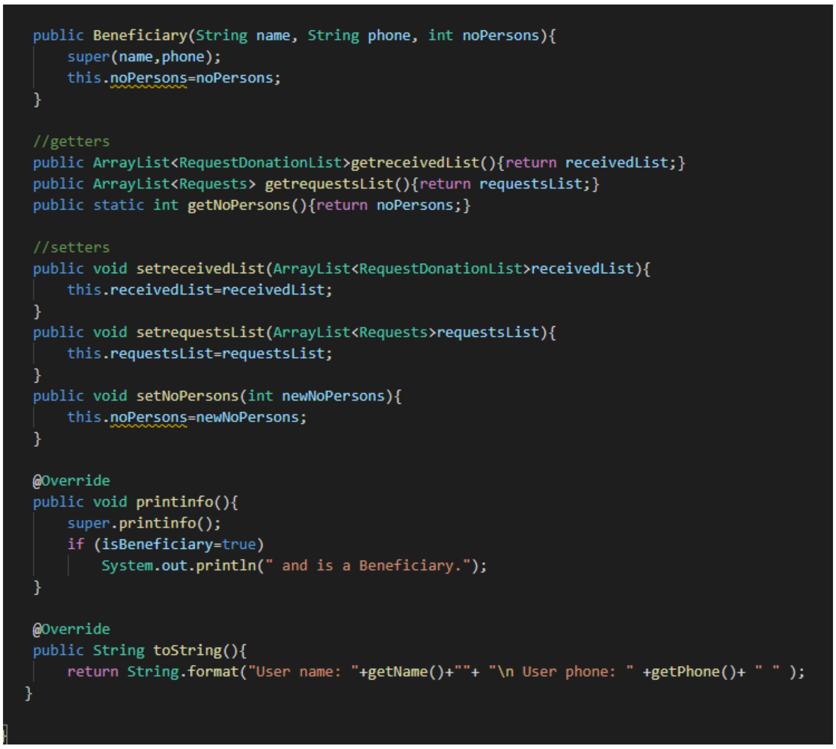


```java
    public void removeOffers(Offers b){
        offersList.remove(b);
    }

    public void setDonator(boolean newDonator){this.isDonator = newDonator;}


    public Donator(String name, String phone){
        super(name,phone);

    }

    //getters
    public  ArrayList<Offers>getoffersList(){return offersList;}


    //setters
    public void setoffersList(ArrayList<Offers>offersList){
        this.offersList=offersList;
    }

    @Override
    public void printinfo(){
        super.printinfo();
        if (isDonator=true)
            System.out.println(" and is a Donator.");
    }

    @Override
    public String toString(){
        return String.format("User name: "+getName()+""+ "\n User phone: " +getPhone()+ " " );
   }

}
```

Για τον **Beneficiary** έχουμε:



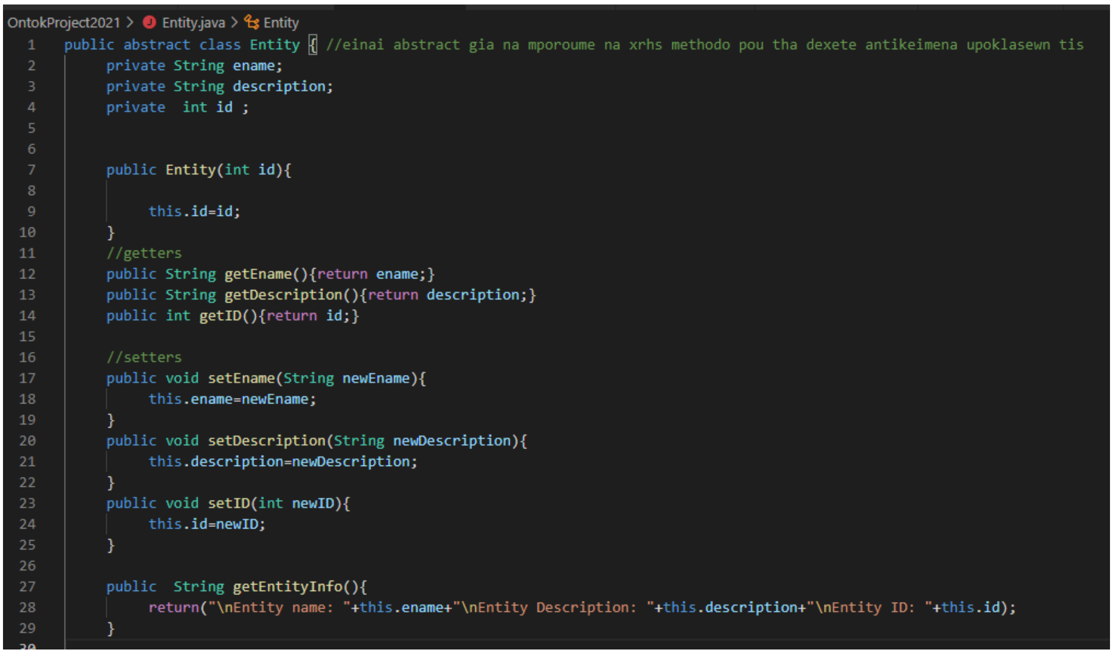
```java
import java.util.*;


public class Beneficiary extends User {
    static int noPersons = 1;
    private boolean isBeneficiary;


    public ArrayList<RequestDonationList> receivedList = new ArrayList<RequestDonationList>();
    public ArrayList<Requests> requestsList = new ArrayList<Requests>();


    public void addRequestDonationList(RequestDonationList a){
        receivedList.add(a);
    }
    public void removeRequestDonationList(RequestDonationList b){
        receivedList.remove(b);
    }

    public void setBeneficiary(boolean newBeneficiary){this.isBeneficiary = newBeneficiary;}



    public void addRequests(Requests a){
        requestsList.add(a);
        Material milk=new Material(1,34.4,45.5,60.0);
        milk.setEname("milk");
        RequestDonation red = new RequestDonation(milk,100.0);
        a.add(red);

        receivedList.add(a);
        Service BabySitting=new Service(3);
        BabySitting.setEname("BabySitting");
        RequestDonation red1 = new RequestDonation(BabySitting,1);
        a.add(red1);
    } //etsi sundeoume ton beneficiary me ta upoloipa

    public void removeRequests(Requests b){
        requestsList.remove(b);
    }
```



```java
    public Beneficiary(String name, String phone, int noPersons){
        super(name,phone);
        this.noPersons=noPersons;
    }

    //getters
    public ArrayList<RequestDonationList>getreceivedList(){return receivedList;}
    public ArrayList<Requests> getrequestsList(){return requestsList;}
    public static int getNoPersons(){return noPersons;}

    //setters
    public void setreceivedList(ArrayList<RequestDonationList>receivedList){
        this.receivedList=receivedList;
    }
    public void setrequestsList(ArrayList<Requests>requestsList){
        this.requestsList=requestsList;
    }
    public void setNoPersons(int newNoPersons){
        this.noPersons=newNoPersons;
    }

    @Override
    public void printinfo(){
        super.printinfo();
        if (isBeneficiary=true)
            System.out.println(" and is a Beneficiary.");
    }

    @Override
    public String toString(){
        return String.format("User name: "+getName()+""+ "\n User phone: " +getPhone()+ " " );
   }
```

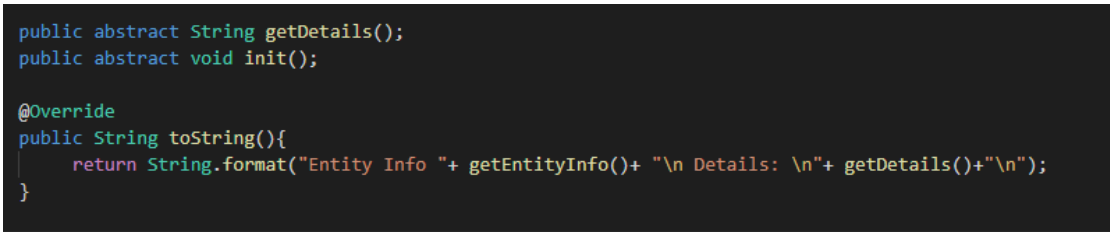
Έπειτα, δημιουργούμε την υπερκλάση **Entity** την οποία και δηλώνουμε ως abstract για τους ίδιους λόγους με την User. Η κλάση περιέχει το όνομα, την περιγραφή και το ID ενός entity (material ή service) με τους κατάλληλους setters getters και έναν κατασκευαστή τον οποίο χρειαζόμαστε για τη δήλωση του ID (μοναδικού) το οποίο θα δηλώνουμε στη main με κάθε αντίστοιχο object. Έπειτα, ορίζουμε την συνάρτηση `public abstract String getDetails();` την οποία και ορίζουμε ως abstract καθώς δεν θέλουμε να ορίσουμε κάτι στο σώμα της στην κλάση Entity ενώ την χρειαζόμαστε για να ορίσουμε τα δεδομένα σε άλλες υποκλάσεις όπου και θα την υπερκαλύψουμε. Έπειτα, την αρχικοποιούμε μέσω της `public abstract void init();`.

Στη συνέχεια, κάνουμε @Override σε μία String to String μέθοδο για να επιστρέψουμε τα στοιχεία (info & details) κάθε Entity.



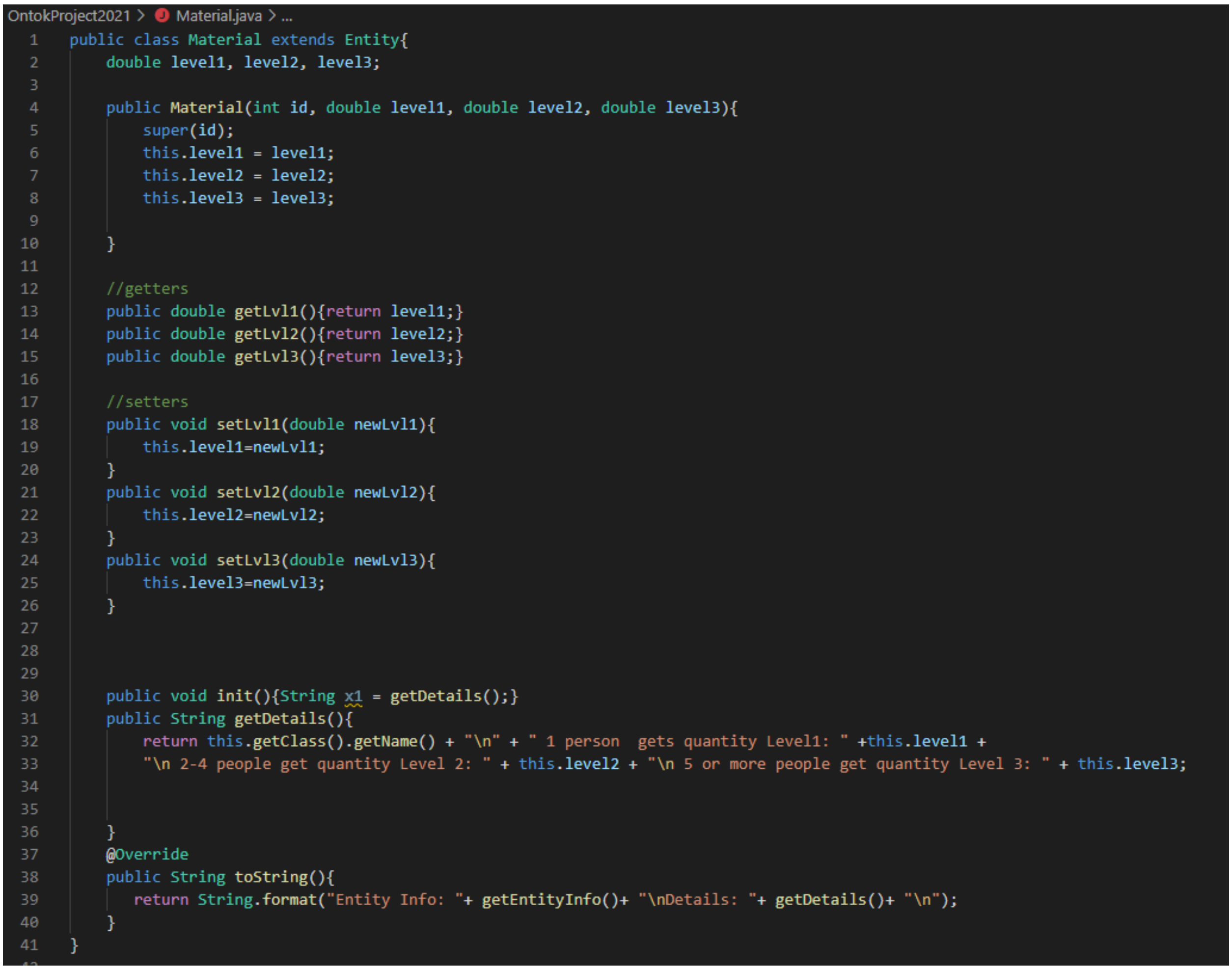
OntokProject2021 > Entity.java > Entity

```java
public abstract class Entity { //einai abstract gia na mporoume na xrhs methodo pou tha dexete antikeimena upoklasewn tis
     private String ename;
     private String description;
     private  int id ;


     public Entity(int id){

          this.id=id;
     }
     //getters
     public String getEname(){return ename;}
     public String getDescription(){return description;}
     public int getID(){return id;}

     //setters
     public void setEname(String newEname){
          this.ename=newEname;
     }
     public void setDescription(String newDescription){
          this.description=newDescription;
     }
     public void setID(int newID){
          this.id=newID;
     }

     public  String getEntityInfo(){
          return("\nEntity name: "+this.ename+"\nEntity Description: "+this.description+"\nEntity ID: "+this.id);
     }
```



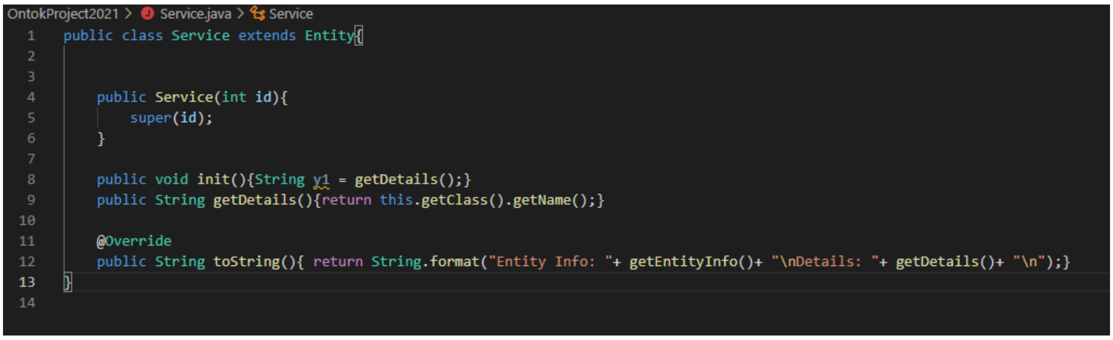
```java
public abstract String getDetails();
public abstract void init();

@Override
public String toString(){
    return String.format("Entity Info "+ getEntityInfo()+ "\n Details: \n"+ getDetails()+"\n");
}
```

Στη συνέχεια, φτιάχνουμε την υποκλάση της Entity, την **Material**, η οποία εκφράζει τα υλικά (π.χ. γάλα, ζάχαρη) και την οποία αναγνωρίζουμε από το id του entity και περιέχει τα levels. Εδώ καλούμε και την getDetails() η οποία θα εμφανίζει τις πληροφορίες αυτής της κλάσης, ενώ κάνουμε και @Override την String toString.



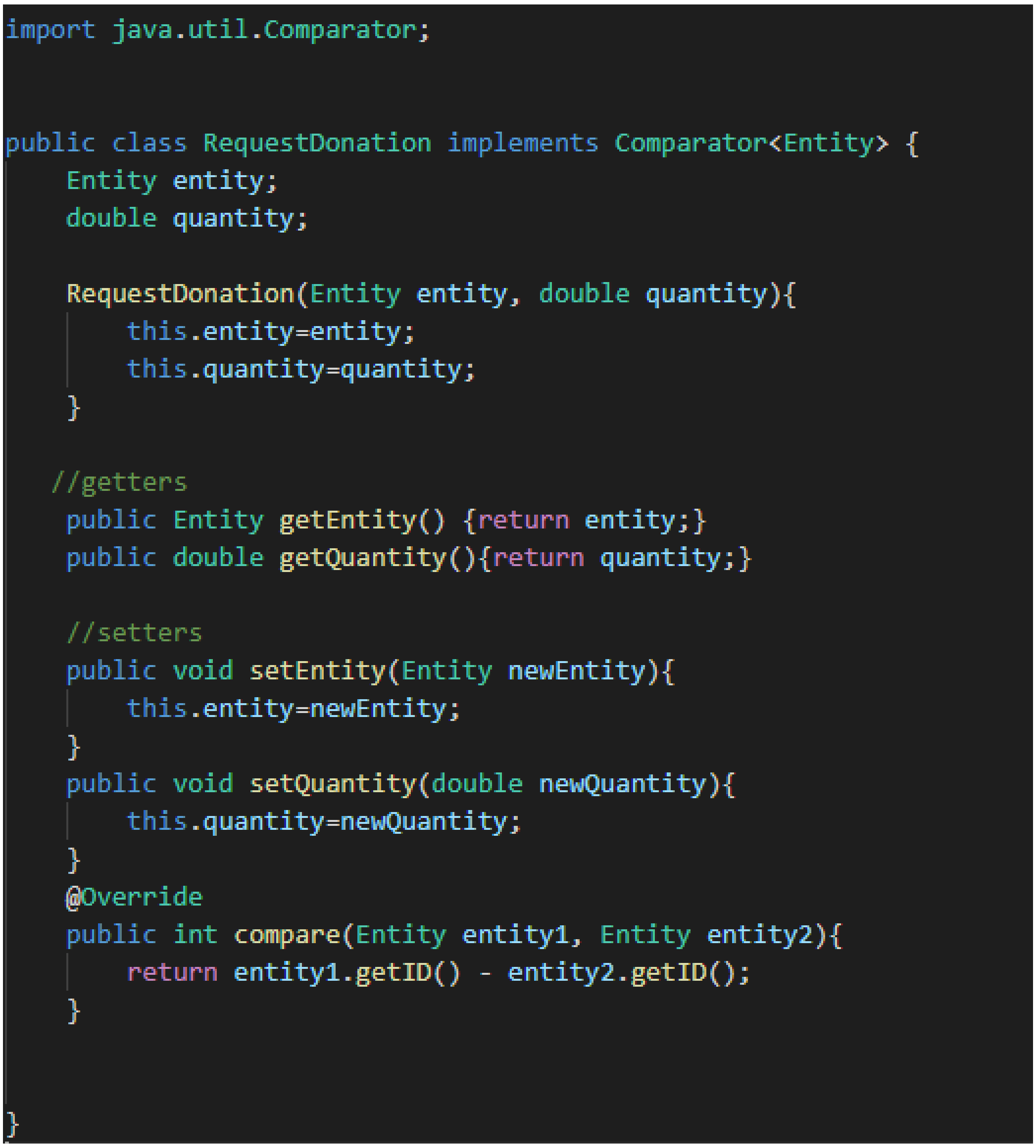
```java
public class Material extends Entity{
    double level1, level2, level3;

    public Material(int id, double level1, double level2, double level3){
        super(id);
        this.level1 = level1;
        this.level2 = level2;
        this.level3 = level3;

    }

    //getters
    public double getLvl1(){return level1;}
    public double getLvl2(){return level2;}
    public double getLvl3(){return level3;}

    //setters
    public void setLvl1(double newLvl1){
        this.level1=newLvl1;
    }
    public void setLvl2(double newLvl2){
        this.level2=newLvl2;
    }
    public void setLvl3(double newLvl3){
        this.level3=newLvl3;
    }



    public void init(){String x1 = getDetails();}
    public String getDetails(){
        return this.getClass().getName() + "\n" + " 1 person  gets quantity Level1: " +this.level1 +
        "\n 2-4 people get quantity Level 2: " + this.level2 + "\n 5 or more people get quantity Level 3: " + this.level3;


    }
    @Override
    public String toString(){
        return String.format("Entity Info: "+ getEntityInfo()+ "\nDetails: "+ getDetails()+ "\n");
    }
}
```

Το ίδιο κάνουμε και για την **Service**.



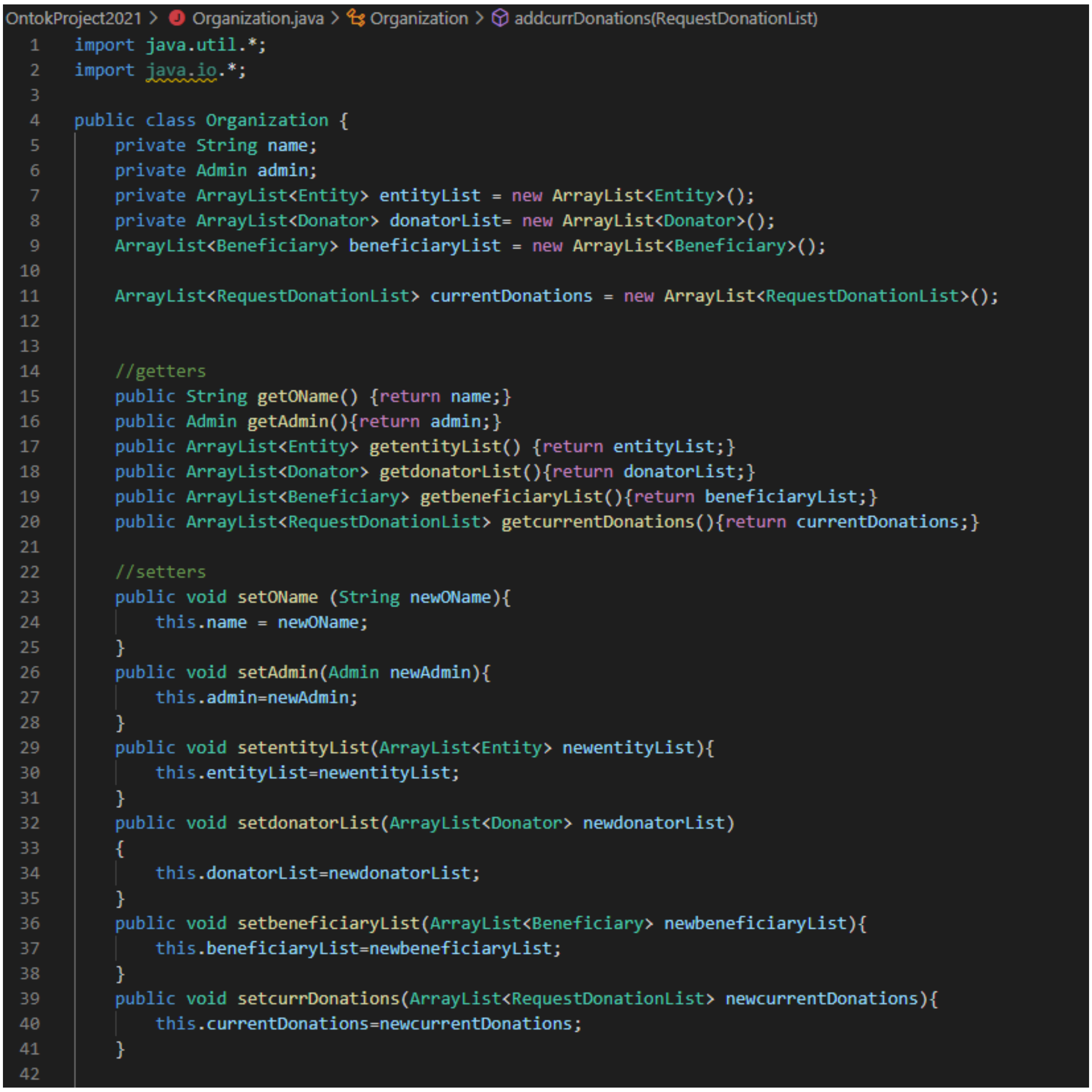
```java
public class Service extends Entity{


    public Service(int id){
        super(id);
    }

    public void init(){String y1 = getDetails();}
    public String getDetails(){return this.getClass().getName();}

    @Override
    public String toString(){ return String.format("Entity Info: "+ getEntityInfo()+ "\nDetails: "+ getDetails()+ "\n");}
}

```

Στην κλάση **RequestDonation**, για να δείξουμε ότι δύο αντικείμενα που αφορούν το ίδιο entity είναι ταυτόσημα κάναμε implement Comparator, ο οποίος συγκρίνει δύο διαφορετικά ID (πχ. από entity1, entity2).

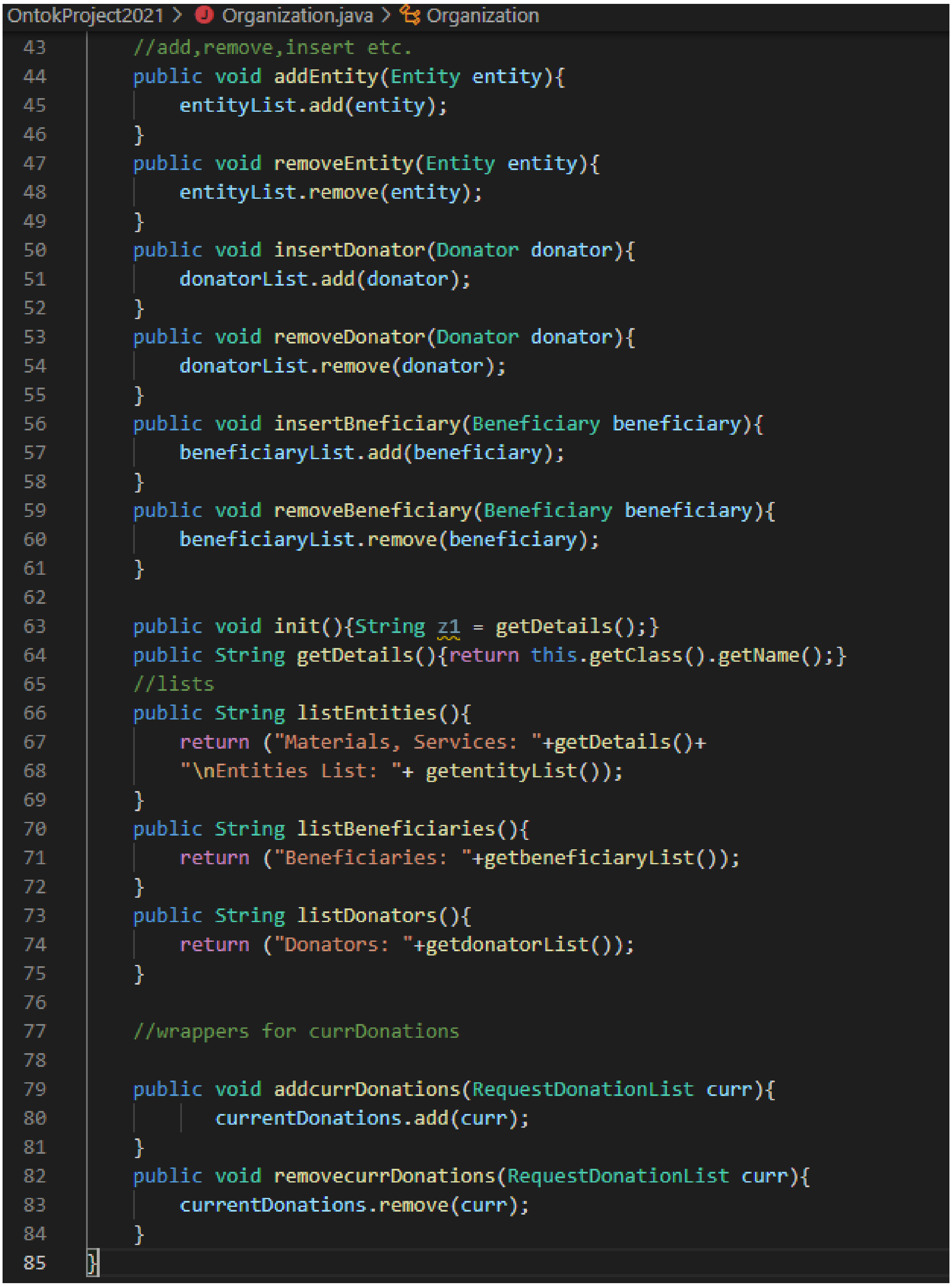


```java
import java.util.Comparator;


public class RequestDonation implements Comparator<Entity> {
    Entity entity;
    double quantity;

    RequestDonation(Entity entity, double quantity){
        this.entity=entity;
        this.quantity=quantity;
    }

   //getters
    public Entity getEntity() {return entity;}
    public double getQuantity(){return quantity;}

    //setters
    public void setEntity(Entity newEntity){
        this.entity=newEntity;
    }
    public void setQuantity(double newQuantity){
        this.quantity=newQuantity;
    }
    @Override
    public int compare(Entity entity1, Entity entity2){
        return entity1.getID() - entity2.getID();
    }


}
```

Στην κλάση **Organization**, υλοποιήσαμε τα ζητούμενα έχοντας μεθόδους με τις οποίες μπορούμε να προσθέσουμε/αφαιρέσουμε όλα τα αντικείμενα/χρήστες του οργανισμού. Υλοποιήσαμε και εδώ την printDetails() η οποία θα επιστρέφει όλα τα δεδομένα (υλικά, υπηρεσίες, beneficiaries, donators) του οργανισμού.

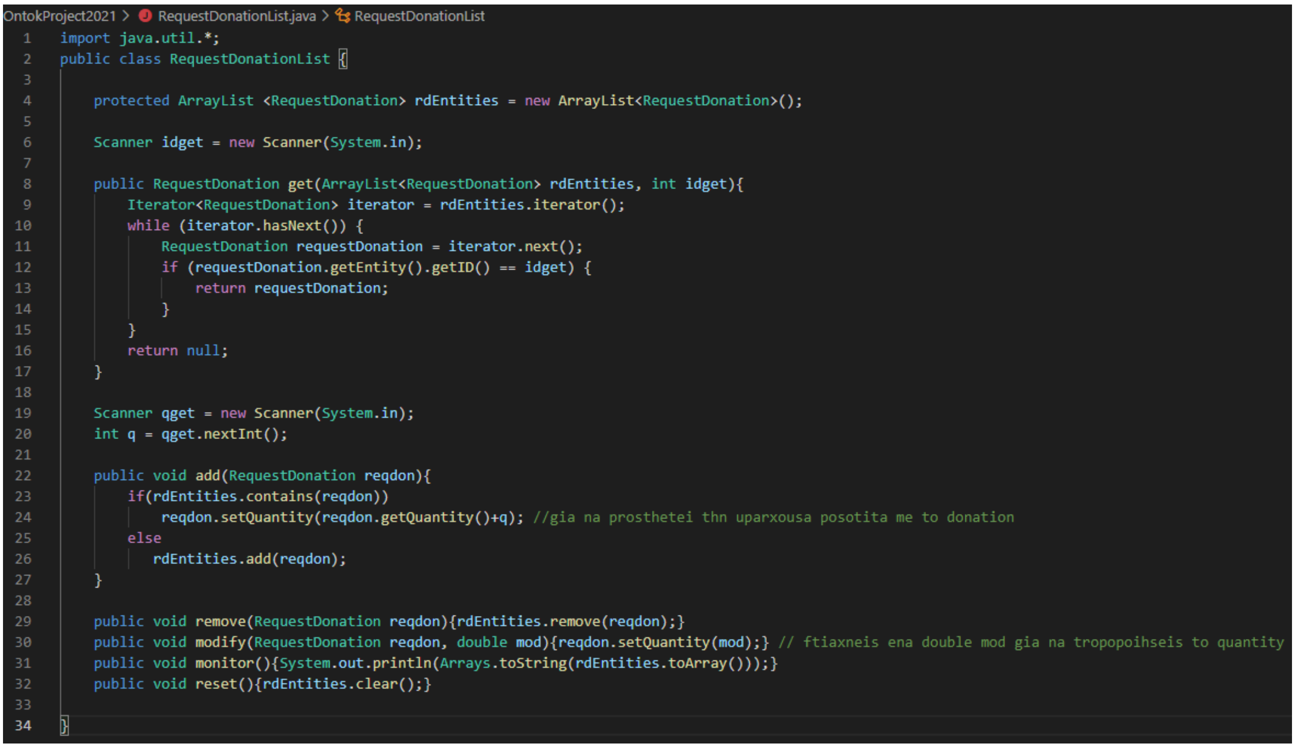

OntokProject2021 > Organization.java > Organization > addcurrDonations(RequestDonationList)

```java
import java.util.*;
import java.io.*;

public class Organization {
    private String name;
    private Admin admin;
    private ArrayList<Entity> entityList = new ArrayList<Entity>();
    private ArrayList<Donator> donatorList= new ArrayList<Donator>();
    ArrayList<Beneficiary> beneficiaryList = new ArrayList<Beneficiary>();

    ArrayList<RequestDonationList> currentDonations = new ArrayList<RequestDonationList>();


    //getters
    public String getOName() {return name;}
    public Admin getAdmin(){return admin;}
    public ArrayList<Entity> getentityList() {return entityList;}
    public ArrayList<Donator> getdonatorList(){return donatorList;}
    public ArrayList<Beneficiary> getbeneficiaryList(){return beneficiaryList;}
    public ArrayList<RequestDonationList> getcurrentDonations(){return currentDonations;}

    //setters
    public void setOName (String newOName){
        this.name = newOName;
    }
    public void setAdmin(Admin newAdmin){
        this.admin=newAdmin;
    }
    public void setentityList(ArrayList<Entity> newentityList){
        this.entityList=newentityList;
    }
    public void setdonatorList(ArrayList<Donator> newdonatorList)
    {
        this.donatorList=newdonatorList;
    }
    public void setbeneficiaryList(ArrayList<Beneficiary> newbeneficiaryList){
        this.beneficiaryList=newbeneficiaryList;
    }
    public void setcurrDonations(ArrayList<RequestDonationList> newcurrentDonations){
        this.currentDonations=newcurrentDonations;
    }

```



```java
    //add,remove,insert etc.
    public void addEntity(Entity entity){
        entityList.add(entity);
    }
    public void removeEntity(Entity entity){
        entityList.remove(entity);
    }
    public void insertDonator(Donator donator){
        donatorList.add(donator);
    }
    public void removeDonator(Donator donator){
        donatorList.remove(donator);
    }
    public void insertBneficiary(Beneficiary beneficiary){
        beneficiaryList.add(beneficiary);
    }
    public void removeBeneficiary(Beneficiary beneficiary){
        beneficiaryList.remove(beneficiary);
    }

    public void init(){String z1 = getDetails();}
    public String getDetails(){return this.getClass().getName();}
    //lists
    public String listEntities(){
        return ("Materials, Services: "+getDetails()+
        "\nEntities List: "+ getentityList());
    }
    public String listBeneficiaries(){
        return ("Beneficiaries: "+getbeneficiaryList());
    }
    public String listDonators(){
        return ("Donators: "+getdonatorList());
    }

    //wrappers for currDonations

    public void addcurrDonations(RequestDonationList curr){
            currentDonations.add(curr);
    }
    public void removecurrDonations(RequestDonationList curr){
        currentDonations.remove(curr);
    }
}
```

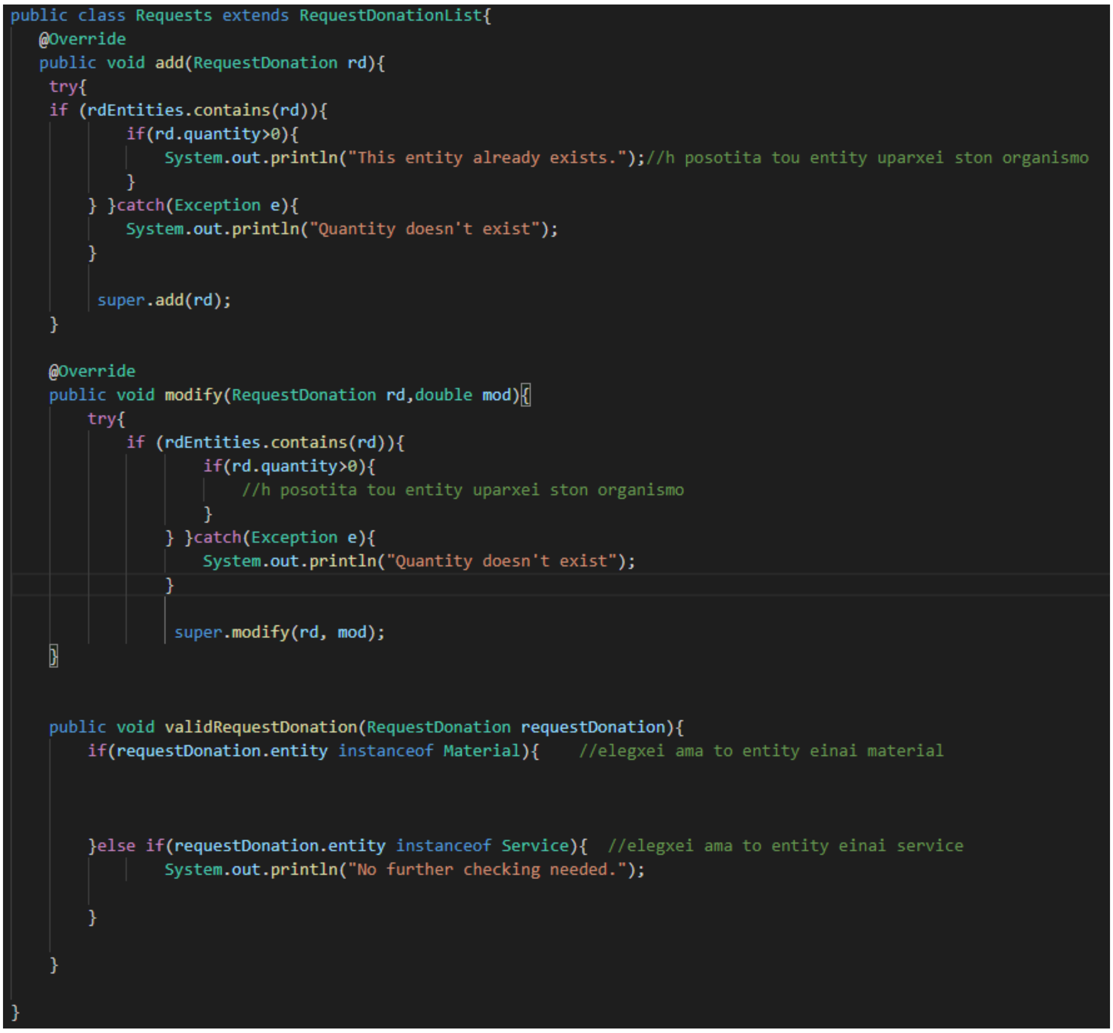
Έπειτα, στην κλάση **RequestDonationList** δημιουργούμε την protected λίστα RequestDonation των rdEntities (για να είναι προσβάσιμες μόνο στις υποκλάσεις της). Για να μπορούμε να επιστρέψουμε ένα αντικείμενο από την λίστα μέσω του id του, υλοποιούμε την μέθοδο get για την RequestDonation έτσι ώστε να πάρουμε τη λίστα των rdEntities και το id. Αυτό το καταφέρνουμε μέσω της χρήσης ενός Iterator και μιας while ενώ στη συνέχεια μέσω του Scanner δίνουμε μία τιμή για ένα quantity το οποίο θα θέλουμε να προστεθεί στις υπόλοιπες ποσότητες (αν υπάρχουν).


OntokProject2021 > RequestDonationList.java > RequestDonationList

```java
import java.util.*;
public class RequestDonationList {

    protected ArrayList <RequestDonation> rdEntities = new ArrayList<RequestDonation>();

    Scanner idget = new Scanner(System.in);

    public RequestDonation get(ArrayList<RequestDonation> rdEntities, int idget){
        Iterator<RequestDonation> iterator = rdEntities.iterator();
        while (iterator.hasNext()) {
            RequestDonation requestDonation = iterator.next();
            if (requestDonation.getEntity().getID() == idget) {
                return requestDonation;
            }
        }
        return null;
    }

    Scanner qget = new Scanner(System.in);
    int q = qget.nextInt();

    public void add(RequestDonation reqdon){
        if(rdEntities.contains(reqdon))
            reqdon.setQuantity(reqdon.getQuantity()+q); //gia na prosthetei thn uparxousa posotita me to donation
        else
          rdEntities.add(reqdon);
    }

    public void remove(RequestDonation reqdon){rdEntities.remove(reqdon);}
    public void modify(RequestDonation reqdon, double mod){reqdon.setQuantity(mod);} // ftiaxneis ena double mod gia na tropopoihseis to quantity
    public void monitor(){System.out.println(Arrays.toString(rdEntities.toArray()));}
    public void reset(){rdEntities.clear();}

}
```

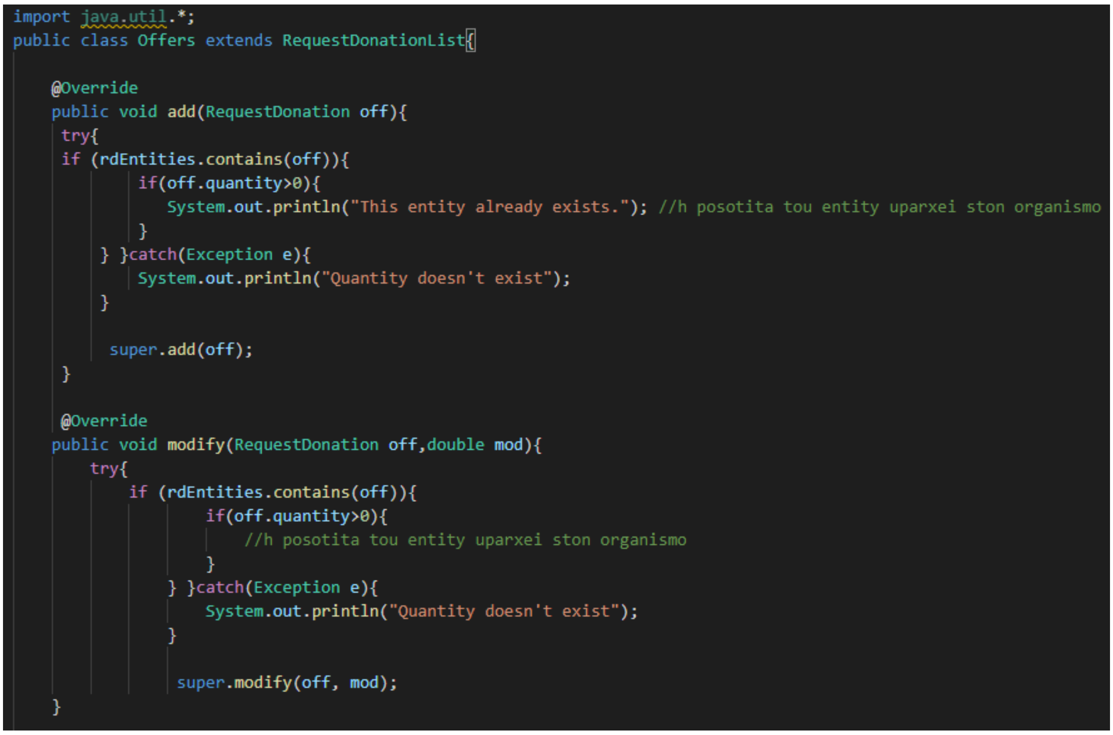
Για να προστεθεί σε ήδη υπάρχουσα ποσότητα βάζουμε το reqdon.setQuantity(reqdon.getQuantity()**+q**);, αλλιώς εάν δεν υπάρχει ήδη κάποιο quantity απλά προσθέτουμε αυτό που βάζουμε τώρα μέσω της add. Στη συνέχεια φτιάχνουμε και τις υπόλοιπες μεθόδους remove,modify,monitor,reset όπου και στη modify προσθέτουμε ένα double mod για να τροποποιήσουμε την ποσότητα (αν δεν βάζαμε το mod, θα κάναμε set την αρχική ποσότητα).

Δημιουργούμε και την υποκλάση της RequestDonationList, την **Requests**, στην οποία υπερκαλύπτουμε την μέθοδο add και modify έτσι ώστε αν υπάρχει το επιλεγμένο entity στον οργανισμό να δείξουμε ότι υπάρχει και να προσθέσουμε και τα υπόλοιπα αλλά και να βγάζει κατάλληλο μήνυμα εξαίρεσης όταν δεν υπάρχει. Το ίδιο κάνουμε και στην modify. Αυτό, όπως και στην επόμενη υποκλάση Offers, συμβαίνει έτσι ώστε να μπορούμε να προσθέσουμε ένα νέο αντικείμενο τύπου RequestDonation στην π.χ. στην addOffers του Donator (γι’αυτό και υπερκαλύπτουμε τις μεθόδους) ή και στη συγκεκριμένη στην addRequests του Beneficiary. Στη συνέχεια δημιουργήσαμε και την μέθοδο validRequestDonation μέσω της οποίας και της instanceof δείχνουμε τον τύπο του κάθε entity (Material ή Service).



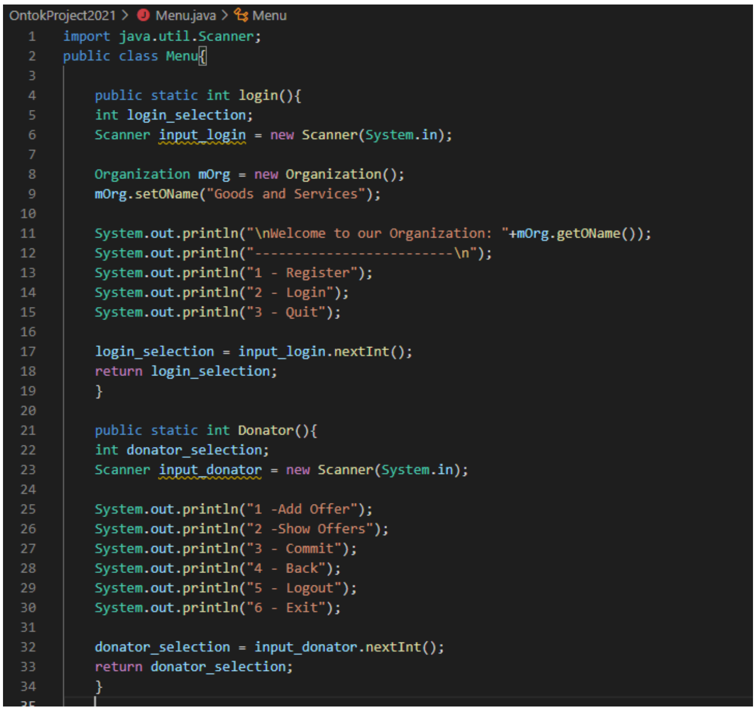
```
public class Requests extends RequestDonationList{
   @Override
   public void add(RequestDonation rd){
    try{
    if (rdEntities.contains(rd)){
            if(rd.quantity>0){
                System.out.println("This entity already exists.");//h posotita tou entity uparxei ston organismo
            }
        } }catch(Exception e){
            System.out.println("Quantity doesn't exist");
        }

         super.add(rd);
    }

    @Override
    public void modify(RequestDonation rd,double mod){
        try{
            if (rdEntities.contains(rd)){
                    if(rd.quantity>0){
                        //h posotita tou entity uparxei ston organismo
                    }
                } }catch(Exception e){
                    System.out.println("Quantity doesn't exist");
                }

                 super.modify(rd, mod);
    }


    public void validRequestDonation(RequestDonation requestDonation){
        if(requestDonation.entity instanceof Material){    //elegxei ama to entity einai material



        }else if(requestDonation.entity instanceof Service){  //elegxei ama to entity einai service
                System.out.println("No further checking needed.");

        }

    }

}
```

Στην υποκλάση **Offers** ακολουθήσαμε την ίδια μεθοδολογία.

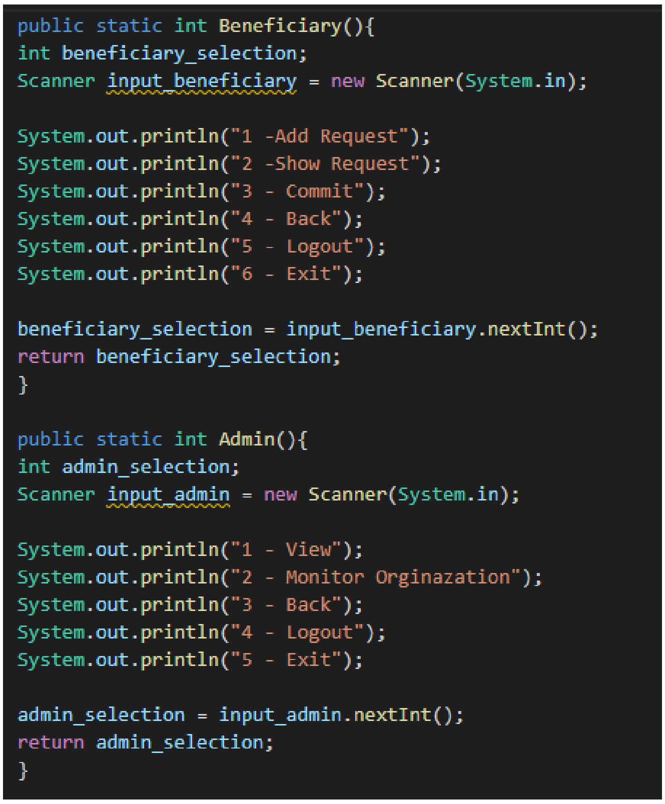


```
import java.util.*;
public class Offers extends RequestDonationList{

    @Override
    public void add(RequestDonation off){
     try{
     if (rdEntities.contains(off)){
             if(off.quantity>0){
                System.out.println("This entity already exists."); //h posotita tou entity uparxei ston organismo
             }
         } }catch(Exception e){
             System.out.println("Quantity doesn't exist");
         }

          super.add(off);
     }

     @Override
    public void modify(RequestDonation off,double mod){
        try{
            if (rdEntities.contains(off)){
                    if(off.quantity>0){
                        //h posotita tou entity uparxei ston organismo
                    }
                } }catch(Exception e){
                    System.out.println("Quantity doesn't exist");
                }

                 super.modify(off, mod);
    }
```

Στην κλάση **Menu** δημιουργούμε μεθόδους όπου εκτυπώνουν τις επιλογές για το menu του κάθε χρήστη σύμφωνα με τα ζητούμενα της εκφώνησης και επιστρέφουν έναν ακέραιο αριθμό. Δημιουργούμε ένα αντικείμενο mOrg τύπου Organization με μέσω της μεθόδου set θέτουμε το όνομα του οργανισμού το οποίο και θα εμφανίζεται στην κονσόλα με το μήνυμα καλωσορίσματος.

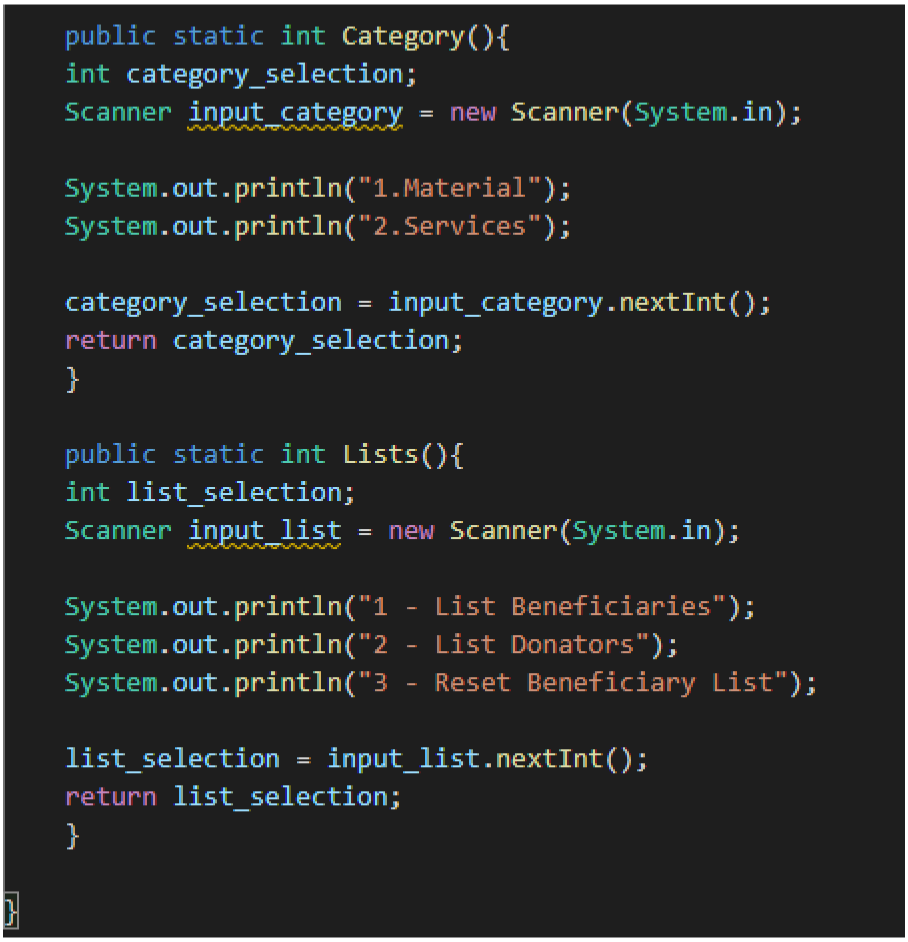


```java
import java.util.Scanner;
public class Menu{

    public static int login(){
    int login_selection;
    Scanner input_login = new Scanner(System.in);

    Organization mOrg = new Organization();
    mOrg.setOName("Goods and Services");

    System.out.println("\nWelcome to our Organization: "+mOrg.getOName());
    System.out.println("------------------------\n");
    System.out.println("1 - Register");
    System.out.println("2 - Login");
    System.out.println("3 - Quit");

    login_selection = input_login.nextInt();
    return login_selection;
    }

    public static int Donator(){
    int donator_selection;
    Scanner input_donator = new Scanner(System.in);

    System.out.println("1 -Add Offer");
    System.out.println("2 -Show Offers");
    System.out.println("3 - Commit");
    System.out.println("4 - Back");
    System.out.println("5 - Logout");
    System.out.println("6 - Exit");

    donator_selection = input_donator.nextInt();
    return donator_selection;
    }
```



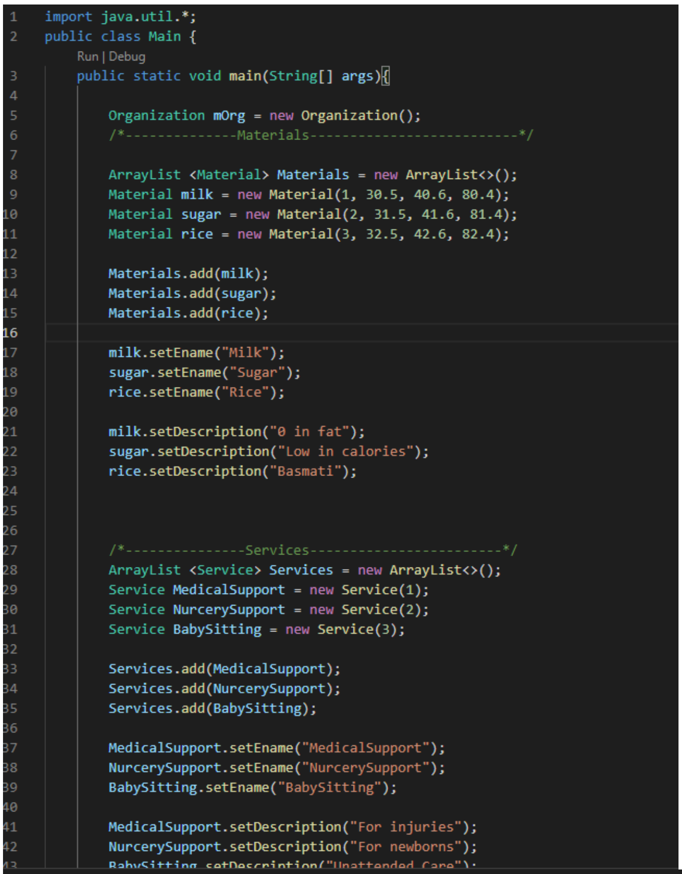
```java
public static int Beneficiary(){
int beneficiary_selection;
Scanner input_beneficiary = new Scanner(System.in);

System.out.println("1 -Add Request");
System.out.println("2 -Show Request");
System.out.println("3 - Commit");
System.out.println("4 - Back");
System.out.println("5 - Logout");
System.out.println("6 - Exit");

beneficiary_selection = input_beneficiary.nextInt();
return beneficiary_selection;
}

public static int Admin(){
int admin_selection;
Scanner input_admin = new Scanner(System.in);

System.out.println("1 - View");
System.out.println("2 - Monitor Orginazation");
System.out.println("3 - Back");
System.out.println("4 - Logout");
System.out.println("5 - Exit");

admin_selection = input_admin.nextInt();
return admin_selection;
}
```

Μέσω της μεθόδου Category επιλέγουμε τα αν θέλουμε υλικό ή υπηρεσία αν ο χρήστης που συνδέεται είναι Donator ή Beneficiary και μέσω της Lists έχουμε τις λίστες με τα στοιχεία του κάθε χρήστη καθώς κάνουμε και reset την λίστα του Beneficiary αν ο χρήστης που συνδέεται είναι ο Διαχειρηστής .

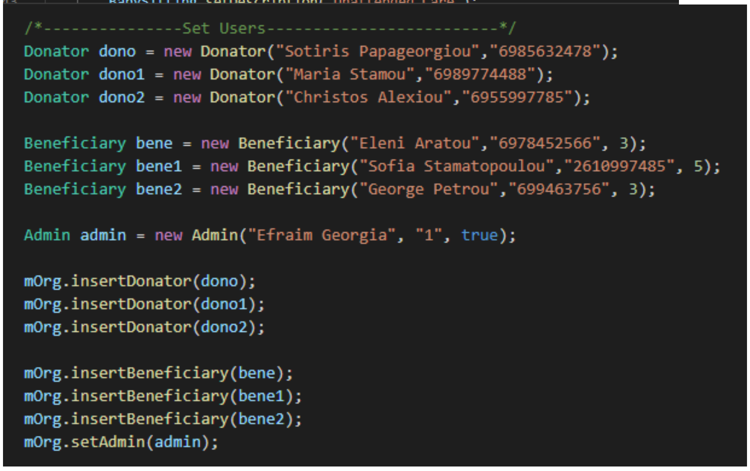


```
    public static int Category(){
    int category_selection;
    Scanner input_category = new Scanner(System.in);

    System.out.println("1.Material");
    System.out.println("2.Services");

    category_selection = input_category.nextInt();
    return category_selection;
    }

    public static int Lists(){
    int list_selection;
    Scanner input_list = new Scanner(System.in);

    System.out.println("1 - List Beneficiaries");
    System.out.println("2 - List Donators");
    System.out.println("3 - Reset Beneficiary List");

    list_selection = input_list.nextInt();
    return list_selection;
    }

}
```

Τέλος, μέσω της **Main** με τη χρήση switches καλούμε τις μεθόδους της Menu, όπου πρώτα δίνεται η επιλογή αν θέλει ο χρήστης να δημιουργήσει λογαριασμό, να συνδεθεί στο λογαριασμό του ή να τερματίσει το πρόγραμμα. Μετά εμφανίζονται επιπλέον επιλογές ανάλογα με τα δικαιώματα του χρήστη(δηλαδή αν είναι Donator, Beneficiary ή Admin). Αν όμως δωθεί αριθμός όπου δεν βρίσκεται στις επιλογές, εμφανίζεται μήνυμα ότι έχει δωθεί λάθος αριθμός. Αυτό επαναλαμβάνεται μέχρι ο χρήστης δώσει σωστή επιλογή. Επίσης υπάρχει η επιλογή Back, όπου με τη χρήση ένως while ξανακαλείται το αντίστοιχο Menu.

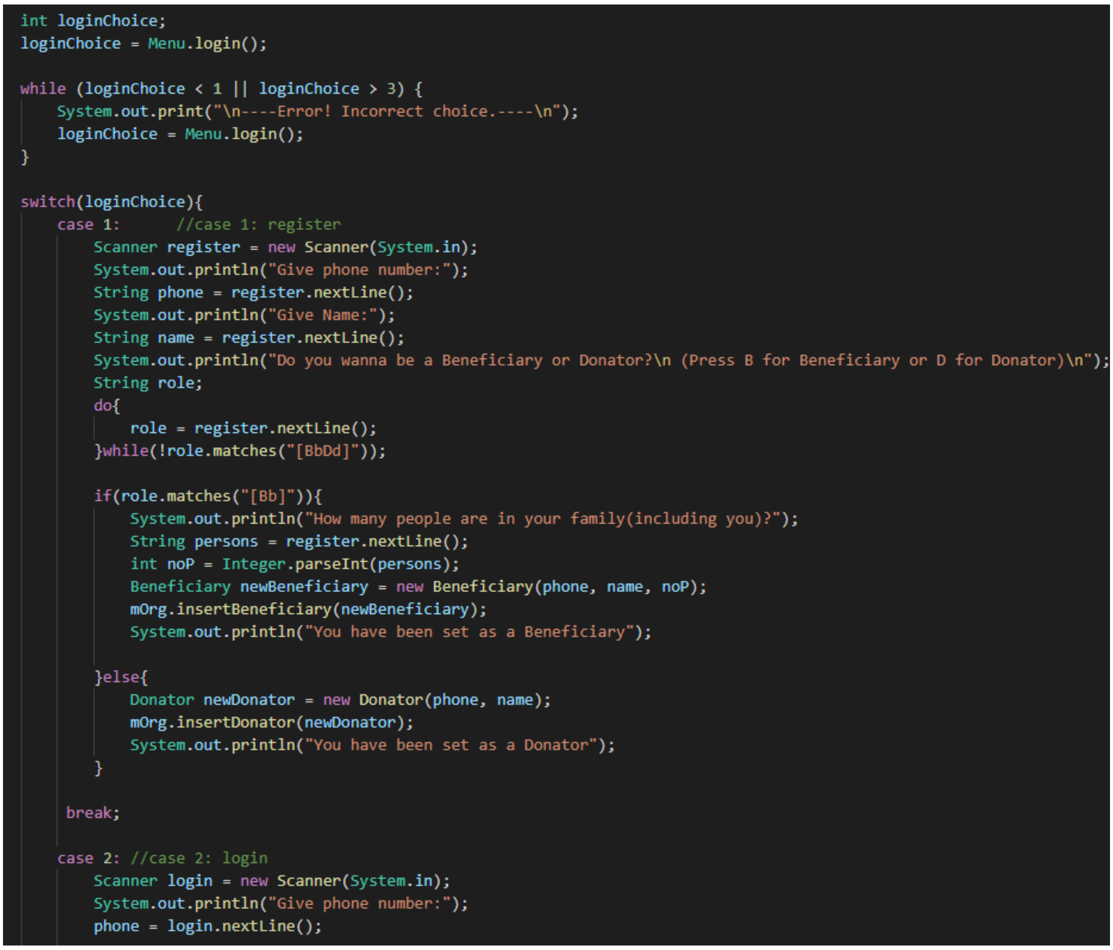


```java
import java.util.*;
public class Main {
    public static void main(String[] args){

        Organization mOrg = new Organization();
        /*-------------Materials------------------------*/

        ArrayList <Material> Materials = new ArrayList<>();
        Material milk = new Material(1, 30.5, 40.6, 80.4);
        Material sugar = new Material(2, 31.5, 41.6, 81.4);
        Material rice = new Material(3, 32.5, 42.6, 82.4);

        Materials.add(milk);
        Materials.add(sugar);
        Materials.add(rice);

        milk.setEname("Milk");
        sugar.setEname("Sugar");
        rice.setEname("Rice");

        milk.setDescription("0 in fat");
        sugar.setDescription("Low in calories");
        rice.setDescription("Basmati");



        /*--------------Services----------------------*/
        ArrayList <Service> Services = new ArrayList<>();
        Service MedicalSupport = new Service(1);
        Service NurcerySupport = new Service(2);
        Service BabySitting = new Service(3);

        Services.add(MedicalSupport);
        Services.add(NurcerySupport);
        Services.add(BabySitting);

        MedicalSupport.setEname("MedicalSupport");
        NurcerySupport.setEname("NurcerySupport");
        BabySitting.setEname("BabySitting");

        MedicalSupport.setDescription("For injuries");
        NurcerySupport.setDescription("For newborns");
        BabySitting.setDescription("Unattended Care");
```



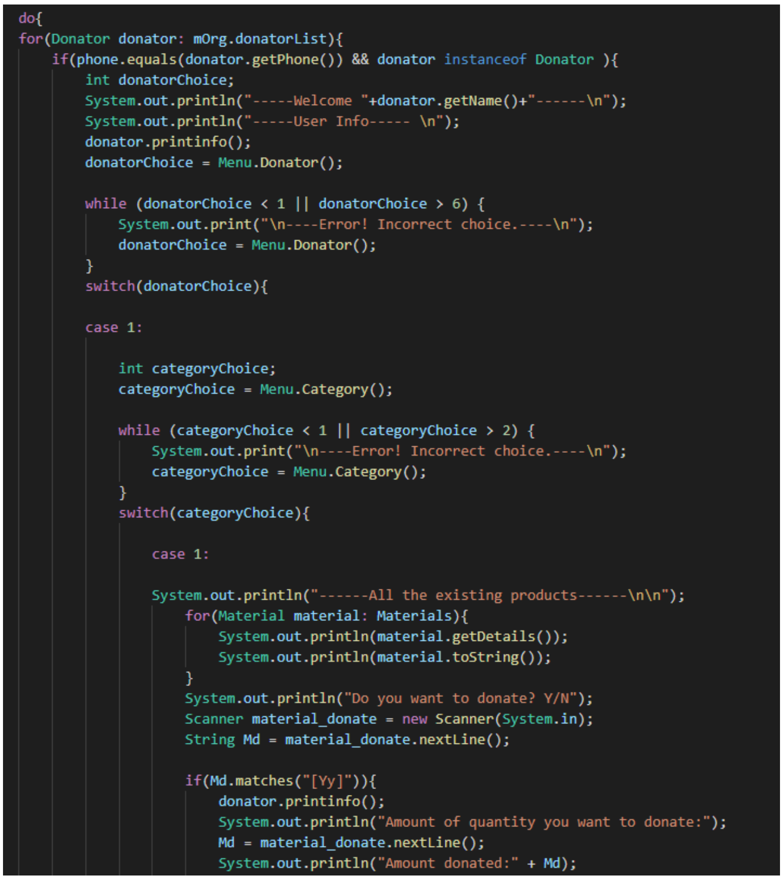
```java
/*--------------Set Users-----------------------*/
Donator dono = new Donator("Sotiris Papageorgiou","6985632478");
Donator dono1 = new Donator("Maria Stamou","6989774488");
Donator dono2 = new Donator("Christos Alexiou","6955997785");

Beneficiary bene = new Beneficiary("Eleni Aratou","6978452566", 3);
Beneficiary bene1 = new Beneficiary("Sofia Stamatopoulou","2610997485", 5);
Beneficiary bene2 = new Beneficiary("George Petrou","699463756", 3);

Admin admin = new Admin("Efraim Georgia", "1", true);

mOrg.insertDonator(dono);
mOrg.insertDonator(dono1);
mOrg.insertDonator(dono2);

mOrg.insertBeneficiary(bene);
mOrg.insertBeneficiary(bene1);
mOrg.insertBeneficiary(bene2);
mOrg.setAdmin(admin);
```

Στην αρχή δηλώσαμε τα objects που θα χρησιμοποιήσουμε με τα ονόματα, περιγραφές και στοιχεία κάθε οντότητας και χρήστη.

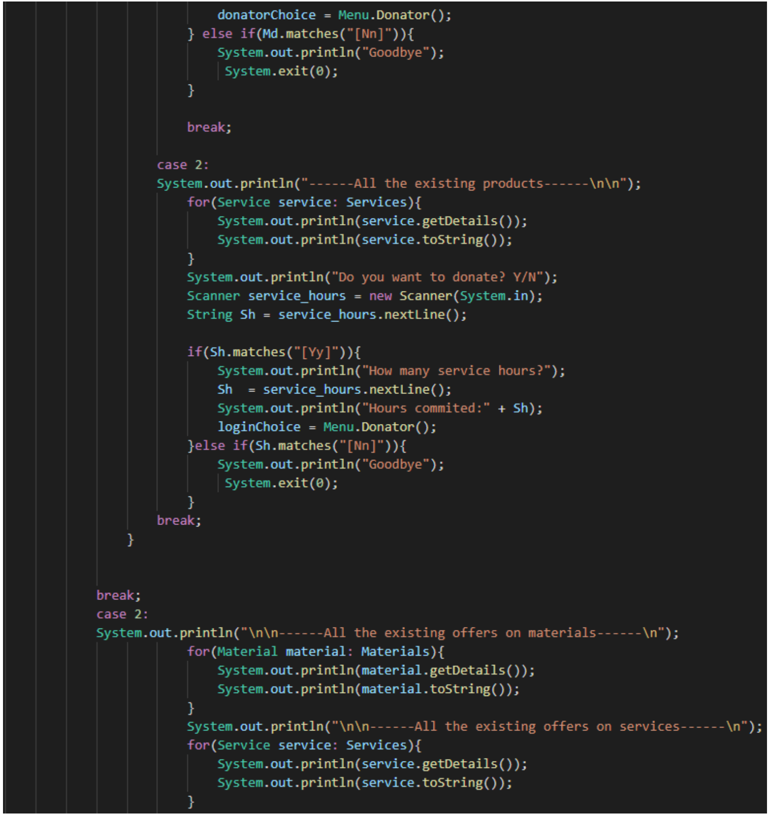


```
int loginChoice;
loginChoice = Menu.login();

while (loginChoice < 1 || loginChoice > 3) {
    System.out.print("\n----Error! Incorrect choice.----\n");
    loginChoice = Menu.login();
}

switch(loginChoice){
    case 1:      //case 1: register
        Scanner register = new Scanner(System.in);
        System.out.println("Give phone number:");
        String phone = register.nextLine();
        System.out.println("Give Name:");
        String name = register.nextLine();
        System.out.println("Do you wanna be a Beneficiary or Donator?\n (Press B for Beneficiary or D for Donator)\n");
        String role;
        do{
            role = register.nextLine();
        }while(!role.matches("[BbDd]"));

        if(role.matches("[Bb]")){
            System.out.println("How many people are in your family(including you)?");
            String persons = register.nextLine();
            int noP = Integer.parseInt(persons);
            Beneficiary newBeneficiary = new Beneficiary(phone, name, noP);
            mOrg.insertBeneficiary(newBeneficiary);
            System.out.println("You have been set as a Beneficiary");

        }else{
            Donator newDonator = new Donator(phone, name);
            mOrg.insertDonator(newDonator);
            System.out.println("You have been set as a Donator");
        }

     break;

    case 2: //case 2: login
        Scanner login = new Scanner(System.in);
        System.out.println("Give phone number:");
        phone = login.nextLine();
```

Συνεχίζοντας, μέσω τον cases, αρχικά φτιάξαμε case στην περίπτωση που ένας χρήστης θέλει να εγγραφεί στο σύστημα και αν θέλει να γίνει Donator ή Beneficiary. Στο 2o case είναι η επιλογή login όπου ο χρήστης δίνει τον αριθμό τηλεφώνου του και ανάλογα σε ποιον ταιριάζει από αυτούς που έχουμε ορίσει στο σύστημα στην αρχή της main τυπώνει τα στοιχεία του.



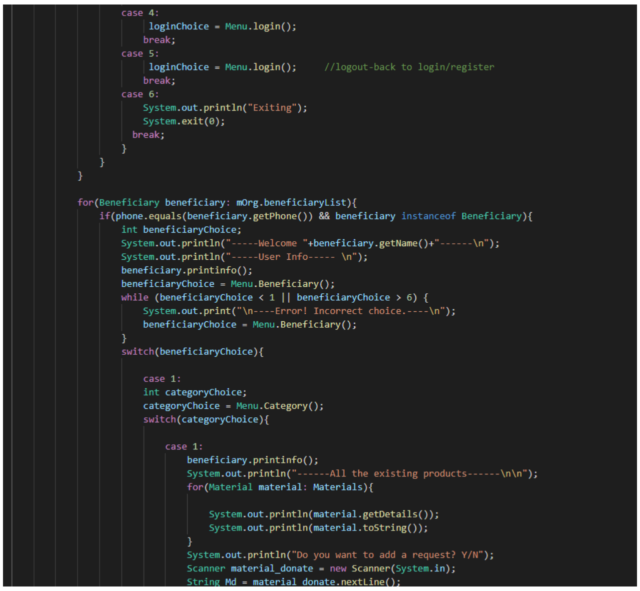
```java
do{
for(Donator donator: mOrg.donatorList){
    if(phone.equals(donator.getPhone()) && donator instanceof Donator ){
        int donatorChoice;
        System.out.println("-----Welcome "+donator.getName()+"------\n");
        System.out.println("-----User Info----- \n");
        donator.printinfo();
        donatorChoice = Menu.Donator();

        while (donatorChoice < 1 || donatorChoice > 6) {
            System.out.print("\n----Error! Incorrect choice.----\n");
            donatorChoice = Menu.Donator();
        }
        switch(donatorChoice){

        case 1:

            int categoryChoice;
            categoryChoice = Menu.Category();

            while (categoryChoice < 1 || categoryChoice > 2) {
                System.out.print("\n----Error! Incorrect choice.----\n");
                categoryChoice = Menu.Category();
            }
            switch(categoryChoice){

                case 1:

                System.out.println("------All the existing products------\n\n");
                    for(Material material: Materials){
                        System.out.println(material.getDetails());
                        System.out.println(material.toString());
                    }
                    System.out.println("Do you want to donate? Y/N");
                    Scanner material_donate = new Scanner(System.in);
                    String Md = material_donate.nextLine();

                    if(Md.matches("[Yy]")){
                        donator.printinfo();
                        System.out.println("Amount of quantity you want to donate:");
                        Md = material_donate.nextLine();
                        System.out.println("Amount donated:" + Md);
```



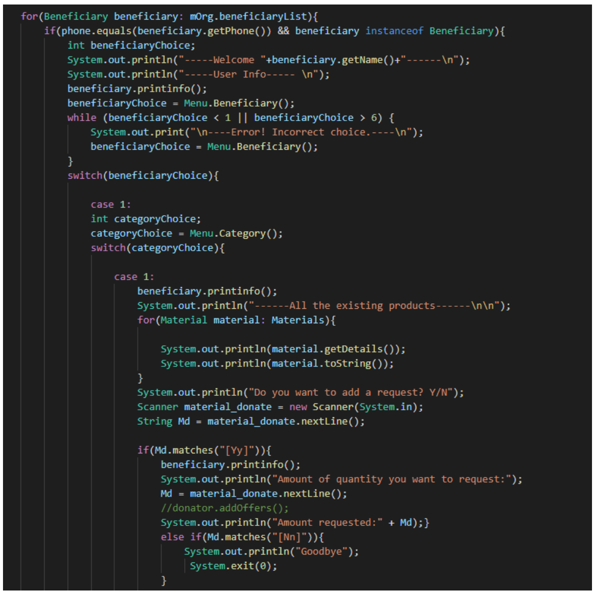
```java
                        donatorChoice = Menu.Donator();
                    } else if(Md.matches("[Nn]")){
                        System.out.println("Goodbye");
                         System.exit(0);
                    }

                    break;

                case 2:
                System.out.println("------All the existing products------\n\n");
                    for(Service service: Services){
                        System.out.println(service.getDetails());
                        System.out.println(service.toString());
                    }
                    System.out.println("Do you want to donate? Y/N");
                    Scanner service_hours = new Scanner(System.in);
                    String Sh = service_hours.nextLine();

                    if(Sh.matches("[Yy]")){
                        System.out.println("How many service hours?");
                        Sh  = service_hours.nextLine();
                        System.out.println("Hours commited:" + Sh);
                        loginChoice = Menu.Donator();
                    }else if(Sh.matches("[Nn]")){
                        System.out.println("Goodbye");
                         System.exit(0);
                    }
                break;
            }


        break;
        case 2:
        System.out.println("\n\n------All the existing offers on materials------\n");
                    for(Material material: Materials){
                        System.out.println(material.getDetails());
                        System.out.println(material.toString());
                    }
                    System.out.println("\n\n------All the existing offers on services------\n");
                    for(Service service: Services){
                        System.out.println(service.getDetails());
                        System.out.println(service.toString());
                    }
```

Για παράδειγμα, αν είναι Donator μέσω της printinfo() τυπώνεται κατάλληλο μήνυμα με τις πληροφορίες του. Έπειτα εμφανίζεται μήνυμα αν θέλει να δωρήσει υλικά/υπηρεσίες καθώς και την ποσότητα.

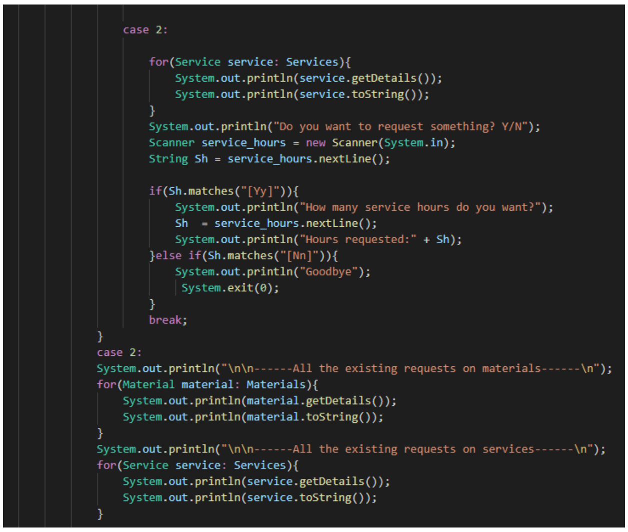
Το ίδιο κάνουμε και στην περίπτωση που ο χρήστης είναι Beneficiary:



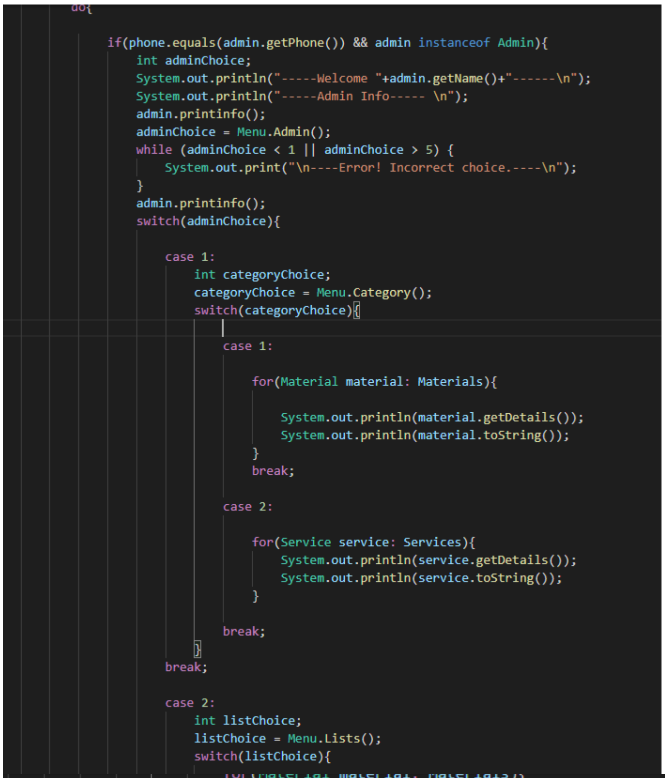
```
                case 4:
                    loginChoice = Menu.login();
                    break;
                case 5:
                    loginChoice = Menu.login();     //logout-back to login/register
                    break;
                case 6:
                    System.out.println("Exiting");
                    System.exit(0);
                  break;
                }
            }
        }

        for(Beneficiary beneficiary: mOrg.beneficiaryList){
            if(phone.equals(beneficiary.getPhone()) && beneficiary instanceof Beneficiary){
                int beneficiaryChoice;
                System.out.println("-----Welcome "+beneficiary.getName()+"------\n");
                System.out.println("-----User Info----- \n");
                beneficiary.printinfo();
                beneficiaryChoice = Menu.Beneficiary();
                while (beneficiaryChoice < 1 || beneficiaryChoice > 6) {
                    System.out.print("\n----Error! Incorrect choice.----\n");
                    beneficiaryChoice = Menu.Beneficiary();
                }
                switch(beneficiaryChoice){

                    case 1:
                    int categoryChoice;
                    categoryChoice = Menu.Category();
                    switch(categoryChoice){

                        case 1:
                            beneficiary.printinfo();
                            System.out.println("------All the existing products------\n\n");
                            for(Material material: Materials){

                                System.out.println(material.getDetails());
                                System.out.println(material.toString());
                            }
                            System.out.println("Do you want to add a request? Y/N");
                            Scanner material_donate = new Scanner(System.in);
                            String Md = material_donate.nextLine();
```



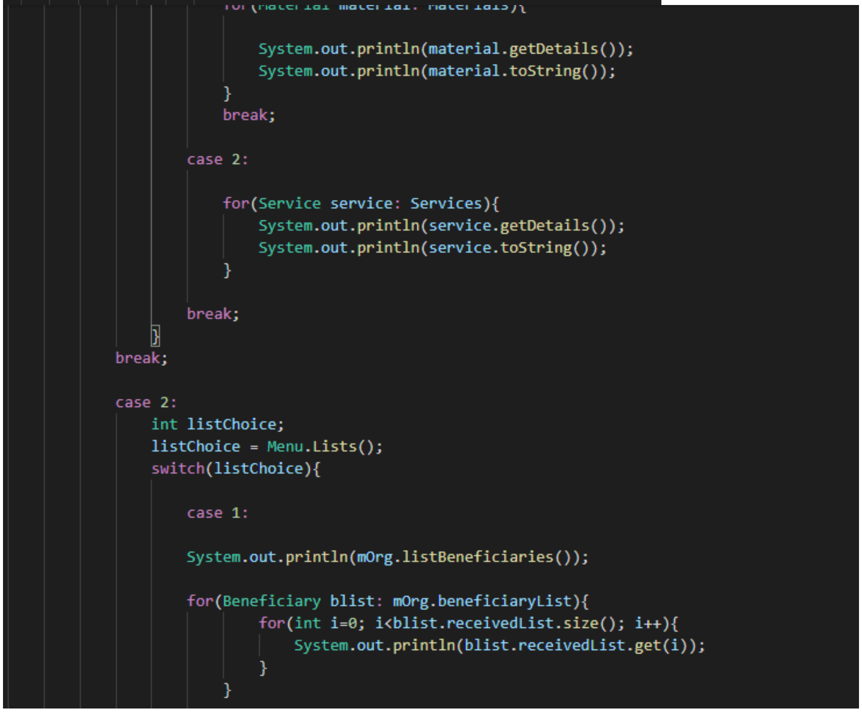
```
for(Beneficiary beneficiary: mOrg.beneficiaryList){
    if(phone.equals(beneficiary.getPhone()) && beneficiary instanceof Beneficiary){
        int beneficiaryChoice;
        System.out.println("-----Welcome "+beneficiary.getName()+"------\n");
        System.out.println("-----User Info----- \n");
        beneficiary.printinfo();
        beneficiaryChoice = Menu.Beneficiary();
        while (beneficiaryChoice < 1 || beneficiaryChoice > 6) {
            System.out.print("\n----Error! Incorrect choice.----\n");
            beneficiaryChoice = Menu.Beneficiary();
        }
        switch(beneficiaryChoice){

            case 1:
            int categoryChoice;
            categoryChoice = Menu.Category();
            switch(categoryChoice){

                case 1:
                    beneficiary.printinfo();
                    System.out.println("------All the existing products------\n\n");
                    for(Material material: Materials){

                        System.out.println(material.getDetails());
                        System.out.println(material.toString());
                    }
                    System.out.println("Do you want to add a request? Y/N");
                    Scanner material_donate = new Scanner(System.in);
                    String Md = material_donate.nextLine();

                    if(Md.matches("[Yy]")){
                        beneficiary.printinfo();
                        System.out.println("Amount of quantity you want to request:");
                        Md = material_donate.nextLine();
                        //donator.addOffers();
                        System.out.println("Amount requested:" + Md);}
                        else if(Md.matches("[Nn]")){
                            System.out.println("Goodbye");
                             System.exit(0);
                        }
```



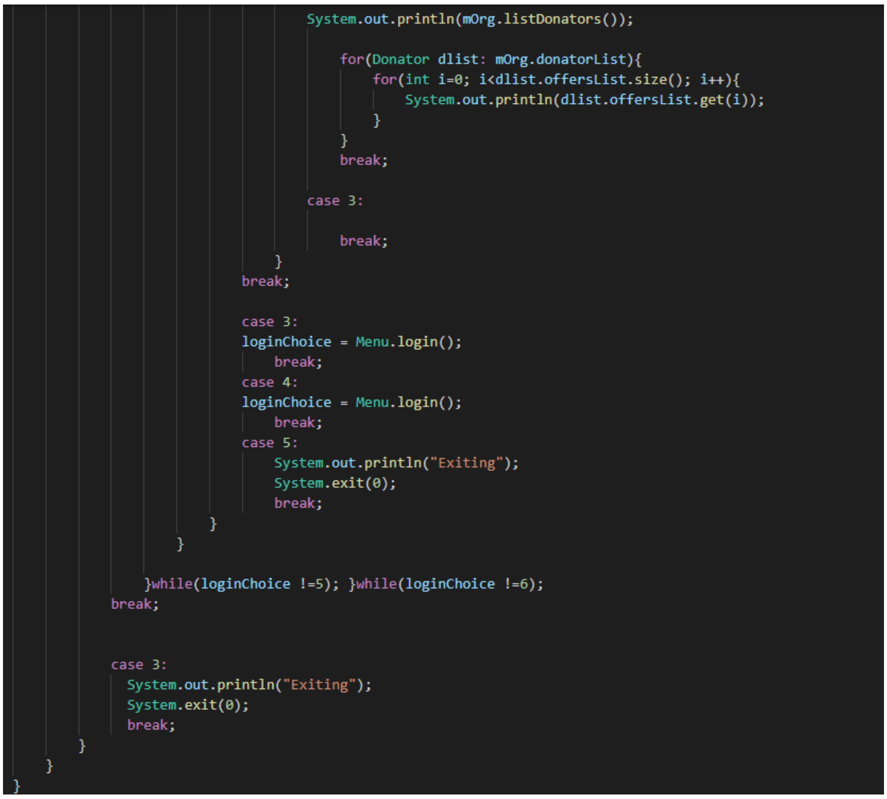
```
                case 2:

                    for(Service service: Services){
                        System.out.println(service.getDetails());
                        System.out.println(service.toString());
                    }
                    System.out.println("Do you want to request something? Y/N");
                    Scanner service_hours = new Scanner(System.in);
                    String Sh = service_hours.nextLine();

                    if(Sh.matches("[Yy]")){
                        System.out.println("How many service hours do you want?");
                        Sh  = service_hours.nextLine();
                        System.out.println("Hours requested:" + Sh);
                    }else if(Sh.matches("[Nn]")){
                        System.out.println("Goodbye");
                         System.exit(0);
                    }
                    break;
            }
            case 2:
            System.out.println("\n\n------All the existing requests on materials------\n");
            for(Material material: Materials){
                System.out.println(material.getDetails());
                System.out.println(material.toString());
            }
            System.out.println("\n\n------All the existing requests on services------\n");
            for(Service service: Services){
                System.out.println(service.getDetails());
                System.out.println(service.toString());
            }
```



```
        do{

            if(phone.equals(admin.getPhone()) && admin instanceof Admin){
                int adminChoice;
                System.out.println("-----Welcome "+admin.getName()+"------\n");
                System.out.println("-----Admin Info----- \n");
                admin.printinfo();
                adminChoice = Menu.Admin();
                while (adminChoice < 1 || adminChoice > 5) {
                    System.out.print("\n----Error! Incorrect choice.----\n");
                }
                admin.printinfo();
                switch(adminChoice){

                    case 1:
                        int categoryChoice;
                        categoryChoice = Menu.Category();
                        switch(categoryChoice){

                            case 1:

                                for(Material material: Materials){

                                    System.out.println(material.getDetails());
                                    System.out.println(material.toString());
                                }
                                break;

                            case 2:

                                for(Service service: Services){
                                    System.out.println(service.getDetails());
                                    System.out.println(service.toString());
                                }

                            break;
                        }
                    break;

                    case 2:
                        int listChoice;
                        listChoice = Menu.Lists();
                        switch(listChoice){
```



```
                                for(Material material: Materials){

                                    System.out.println(material.getDetails());
                                    System.out.println(material.toString());
                                }
                                break;

                            case 2:

                                for(Service service: Services){
                                    System.out.println(service.getDetails());
                                    System.out.println(service.toString());
                                }

                            break;
                        }
                    break;

                    case 2:
                        int listChoice;
                        listChoice = Menu.Lists();
                        switch(listChoice){

                            case 1:

                            System.out.println(mOrg.listBeneficiaries());

                            for(Beneficiary blist: mOrg.beneficiaryList){
                                    for(int i=0; i<blist.receivedList.size(); i++){
                                        System.out.println(blist.receivedList.get(i));
                                    }
                                }
```

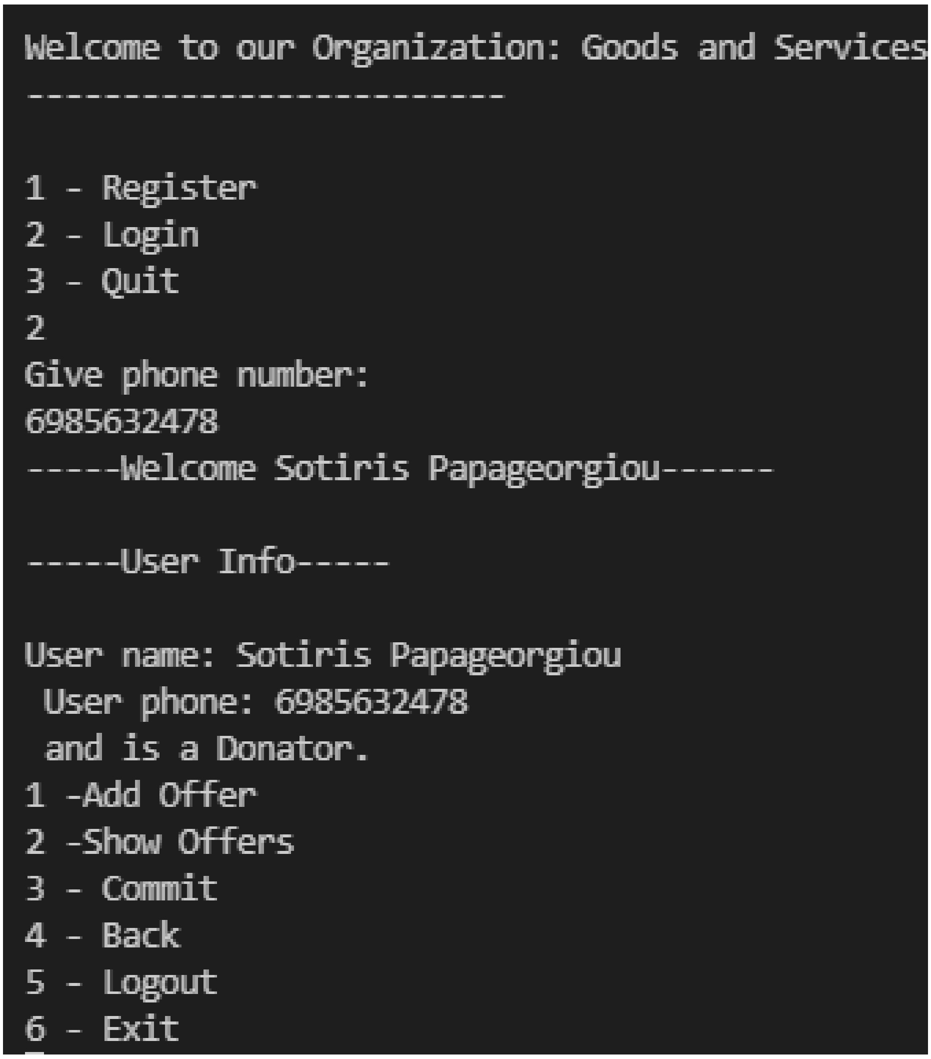


```
                                    System.out.println(mOrg.listDonators());

                                        for(Donator dlist: mOrg.donatorList){
                                            for(int i=0; i<dlist.offersList.size(); i++){
                                                System.out.println(dlist.offersList.get(i));
                                            }
                                        }
                                        break;

                                    case 3:

                                        break;
                                }
                            break;

                            case 3:
                            loginChoice = Menu.login();
                                break;
                            case 4:
                            loginChoice = Menu.login();
                                break;
                            case 5:
                                System.out.println("Exiting");
                                System.exit(0);
                                break;
                        }
                    }

                }while(loginChoice !=5); }while(loginChoice !=6);
            break;


            case 3:
              System.out.println("Exiting");
              System.exit(0);
              break;
        }
    }
}
```

Παρόμοια, και στην περίπτωση που κάνει login ο Admin.

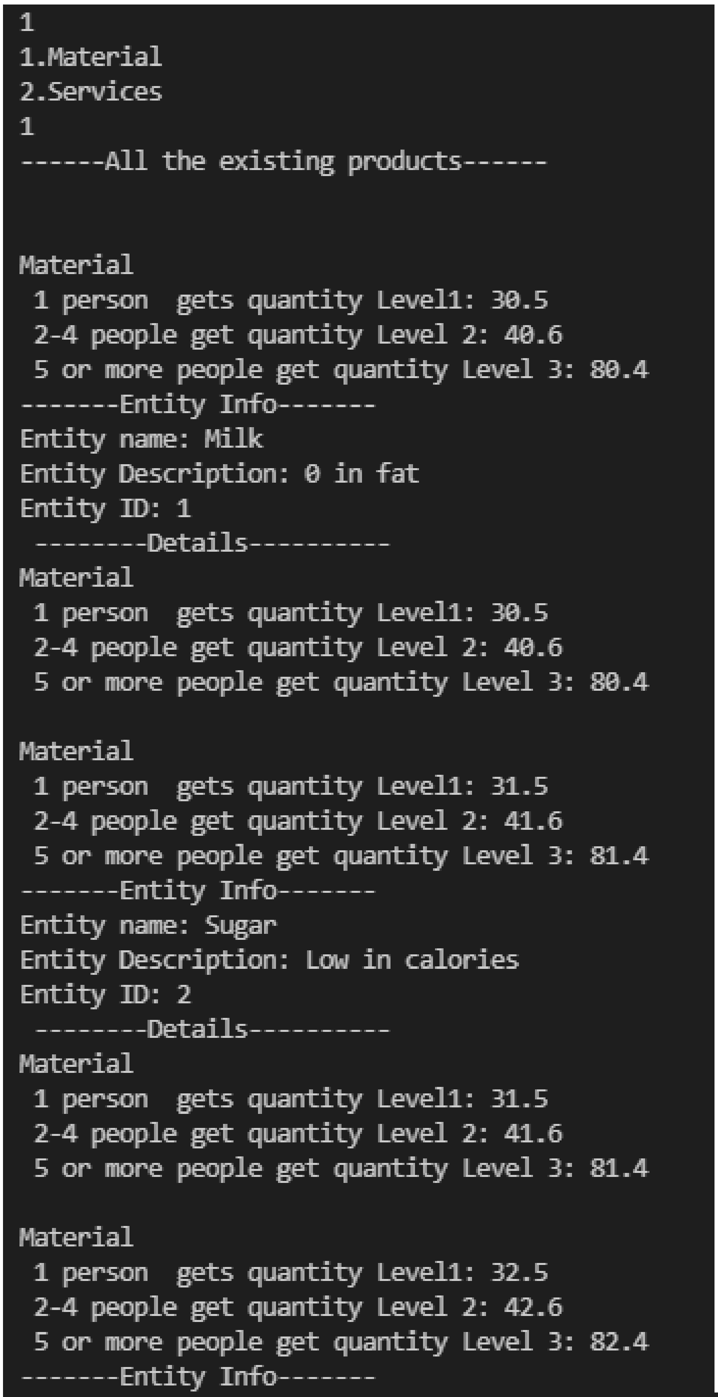
Γενικά, χρησιμοποιούμε και συνθήκες while στην περίπτωση που ως input ο χρήστης βάλει αριθμό που δεν αντιστοιχεί κάποια επιλογή και εμφανίζεται κατάλληλο μήνυμα.

Αποτελέσματα:

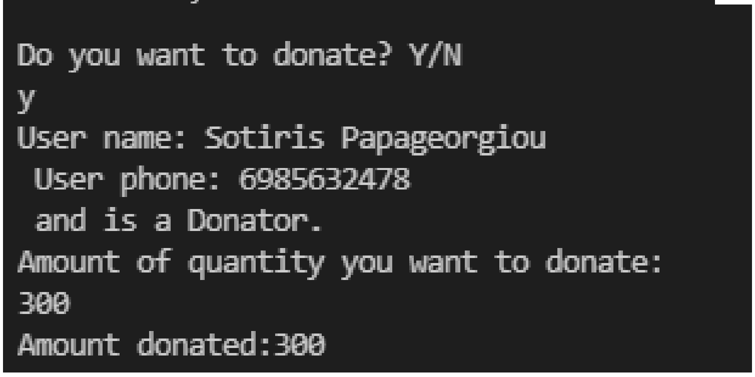


```
Welcome to our Organization: Goods and Services
-------------------------


1 - Register
2 - Login
3 - Quit
2
Give phone number:
6985632478
-----Welcome Sotiris Papageorgiou------

-----User Info-----

User name: Sotiris Papageorgiou
 User phone: 6985632478
 and is a Donator.
1 -Add Offer
2 -Show Offers
3 - Commit
4 - Back
5 - Logout
6 - Exit
```

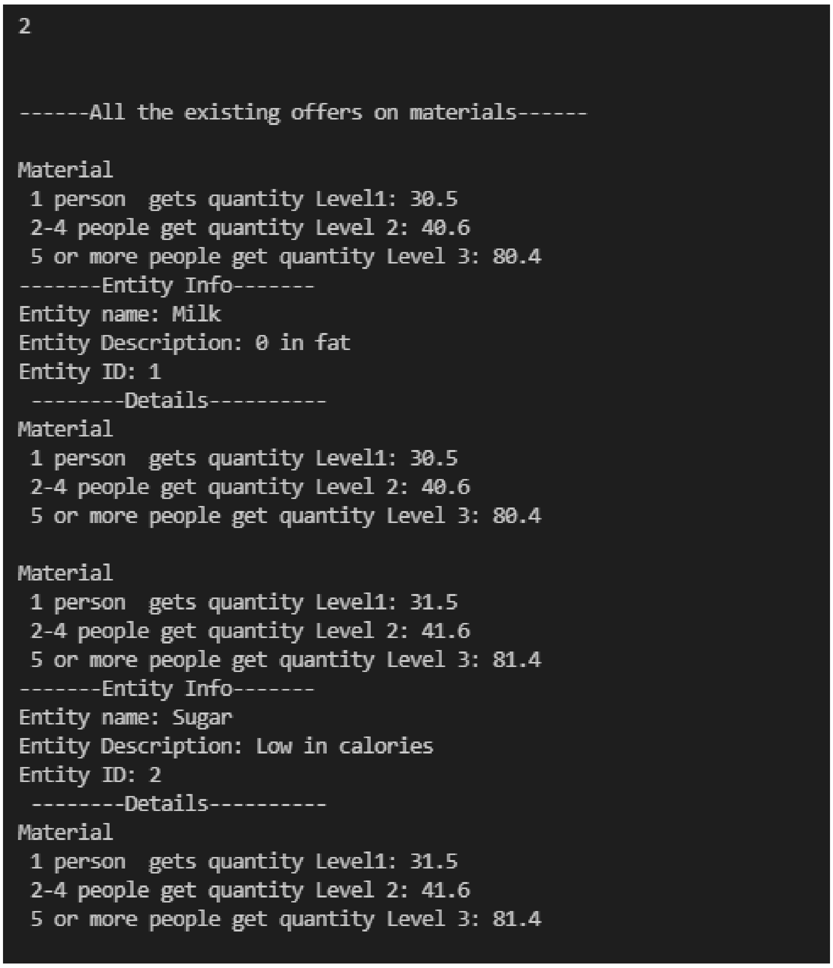
Εδώ βάλαμε το τηλέφωνο ενός ήδη εγεγγραμμένου Donator. Έπειτα, αν επιλέξουμε την επιλογή Add Offer (1) εμφανίζονται τα παρακάτω:



```
1
1.Material
2.Services
1
------All the existing products------


Material
 1 person  gets quantity Level1: 30.5
 2-4 people get quantity Level 2: 40.6
 5 or more people get quantity Level 3: 80.4
-------Entity Info-------
Entity name: Milk
Entity Description: 0 in fat
Entity ID: 1
 --------Details----------
Material
 1 person  gets quantity Level1: 30.5
 2-4 people get quantity Level 2: 40.6
 5 or more people get quantity Level 3: 80.4

Material
 1 person  gets quantity Level1: 31.5
 2-4 people get quantity Level 2: 41.6
 5 or more people get quantity Level 3: 81.4
-------Entity Info-------
Entity name: Sugar
Entity Description: Low in calories
Entity ID: 2
 --------Details----------
Material
 1 person  gets quantity Level1: 31.5
 2-4 people get quantity Level 2: 41.6
 5 or more people get quantity Level 3: 81.4

Material
 1 person  gets quantity Level1: 32.5
 2-4 people get quantity Level 2: 42.6
 5 or more people get quantity Level 3: 82.4
-------Entity Info-------
```

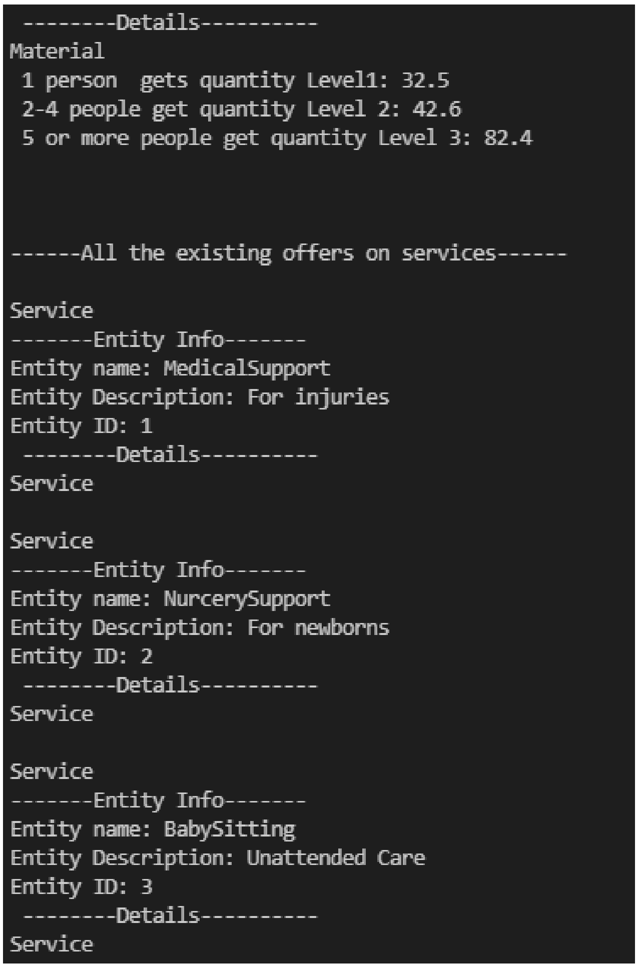


```
Do you want to donate? Y/N
y
User name: Sotiris Papageorgiou
 User phone: 6985632478
 and is a Donator.
Amount of quantity you want to donate:
300
Amount donated:300
```



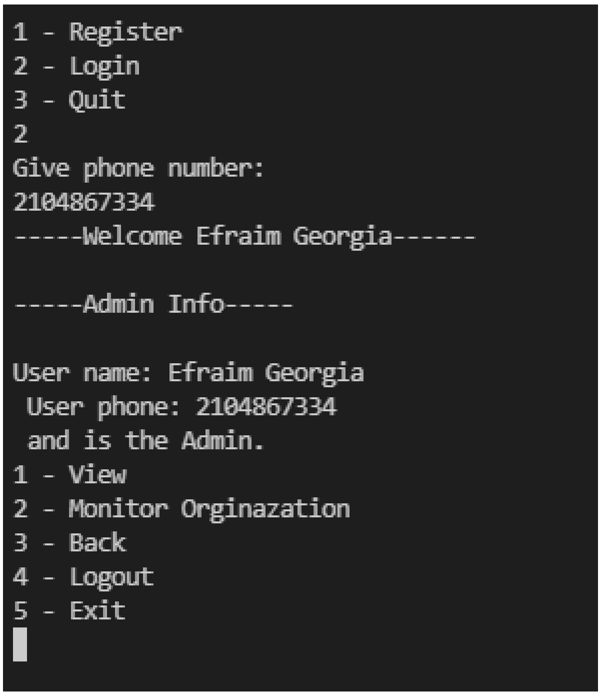
```
2


------All the existing offers on materials------

Material
 1 person  gets quantity Level1: 30.5
 2-4 people get quantity Level 2: 40.6
 5 or more people get quantity Level 3: 80.4
-------Entity Info-------
Entity name: Milk
Entity Description: 0 in fat
Entity ID: 1
 --------Details----------
Material
 1 person  gets quantity Level1: 30.5
 2-4 people get quantity Level 2: 40.6
 5 or more people get quantity Level 3: 80.4

Material
 1 person  gets quantity Level1: 31.5
 2-4 people get quantity Level 2: 41.6
 5 or more people get quantity Level 3: 81.4
-------Entity Info-------
Entity name: Sugar
Entity Description: Low in calories
Entity ID: 2
 --------Details----------
Material
 1 person  gets quantity Level1: 31.5
 2-4 people get quantity Level 2: 41.6
 5 or more people get quantity Level 3: 81.4
```

Εδώ, επιλέξαμε την (2) όπου εκτελείται η Show Offers. Αντίστοιχα αποτελέσματα και για τον Beneficiary.

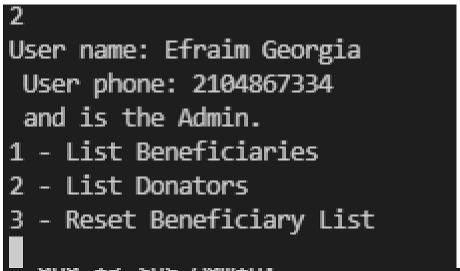


```
 --------Details----------
Material
 1 person  gets quantity Level1: 32.5
 2-4 people get quantity Level 2: 42.6
 5 or more people get quantity Level 3: 82.4



------All the existing offers on services------

Service
-------Entity Info-------
Entity name: MedicalSupport
Entity Description: For injuries
Entity ID: 1
 --------Details----------
Service

Service
-------Entity Info-------
Entity name: NurcerySupport
Entity Description: For newborns
Entity ID: 2
 --------Details----------
Service

Service
-------Entity Info-------
Entity name: BabySitting
Entity Description: Unattended Care
Entity ID: 3
 --------Details----------
Service
```

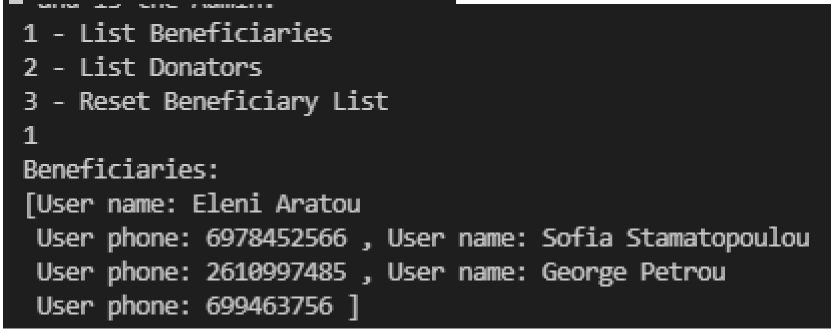
Τέλος, κάνουμε login με τον Admin και ως παράδειγμα τυπώσαμε την λίστα των ήδη υπάρχοντων Beneficiary.



```
1 - Register
2 - Login
3 - Quit
2
Give phone number:
2104867334
-----Welcome Efraim Georgia------

-----Admin Info-----

User name: Efraim Georgia
 User phone: 2104867334
 and is the Admin.
1 - View
2 - Monitor Orginazation
3 - Back
4 - Logout
5 - Exit
```



```
2
User name: Efraim Georgia
 User phone: 2104867334
 and is the Admin.
1 - List Beneficiaries
2 - List Donators
3 - Reset Beneficiary List
```

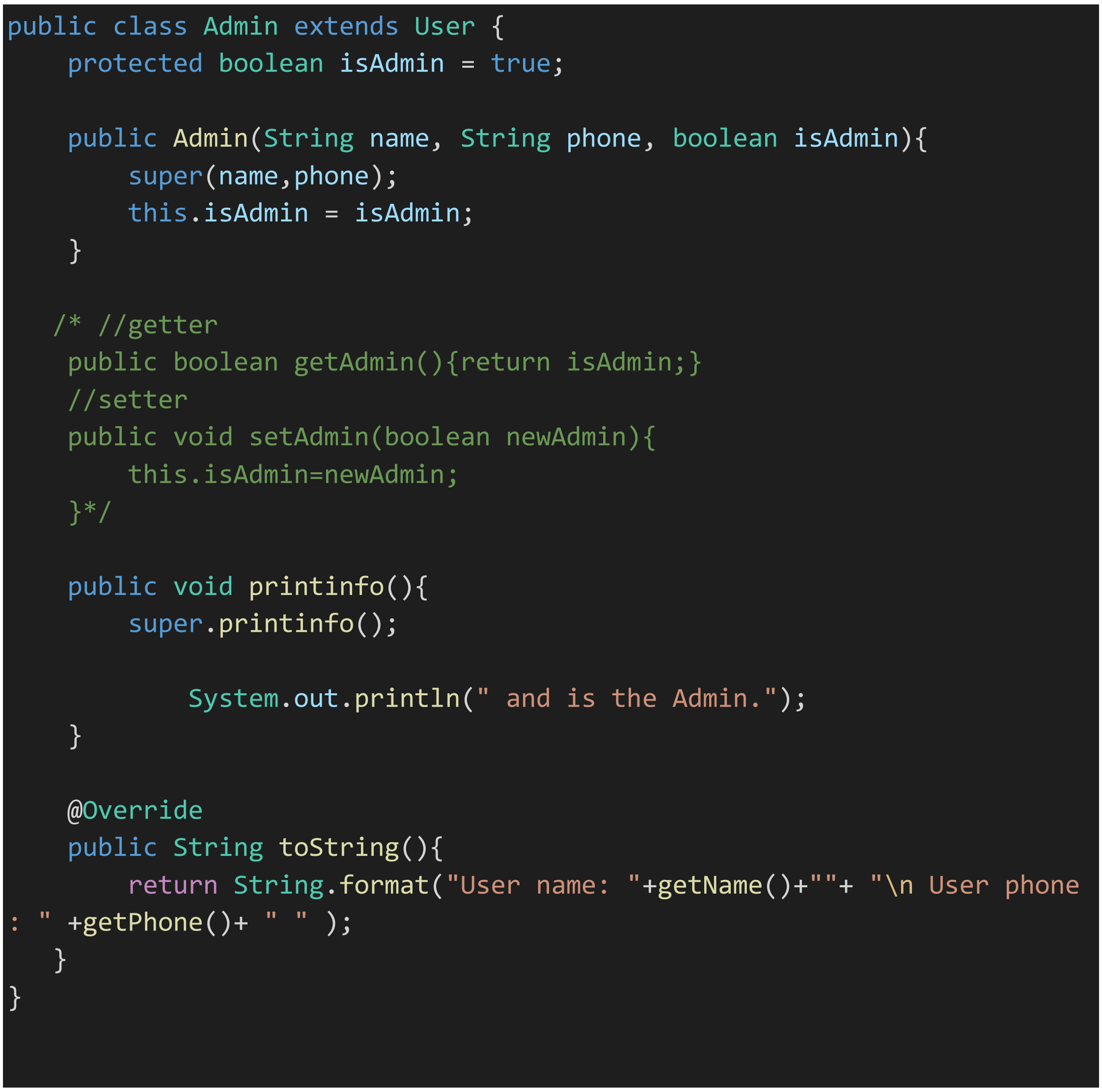


```
 and is the Admin.
 1 - List Beneficiaries
 2 - List Donators
 3 - Reset Beneficiary List
 1
 Beneficiaries:
 [User name: Eleni Aratou
  User phone: 6978452566 , User name: Sofia Stamatopoulou
  User phone: 2610997485 , User name: George Petrou
  User phone: 699463756 ]
```

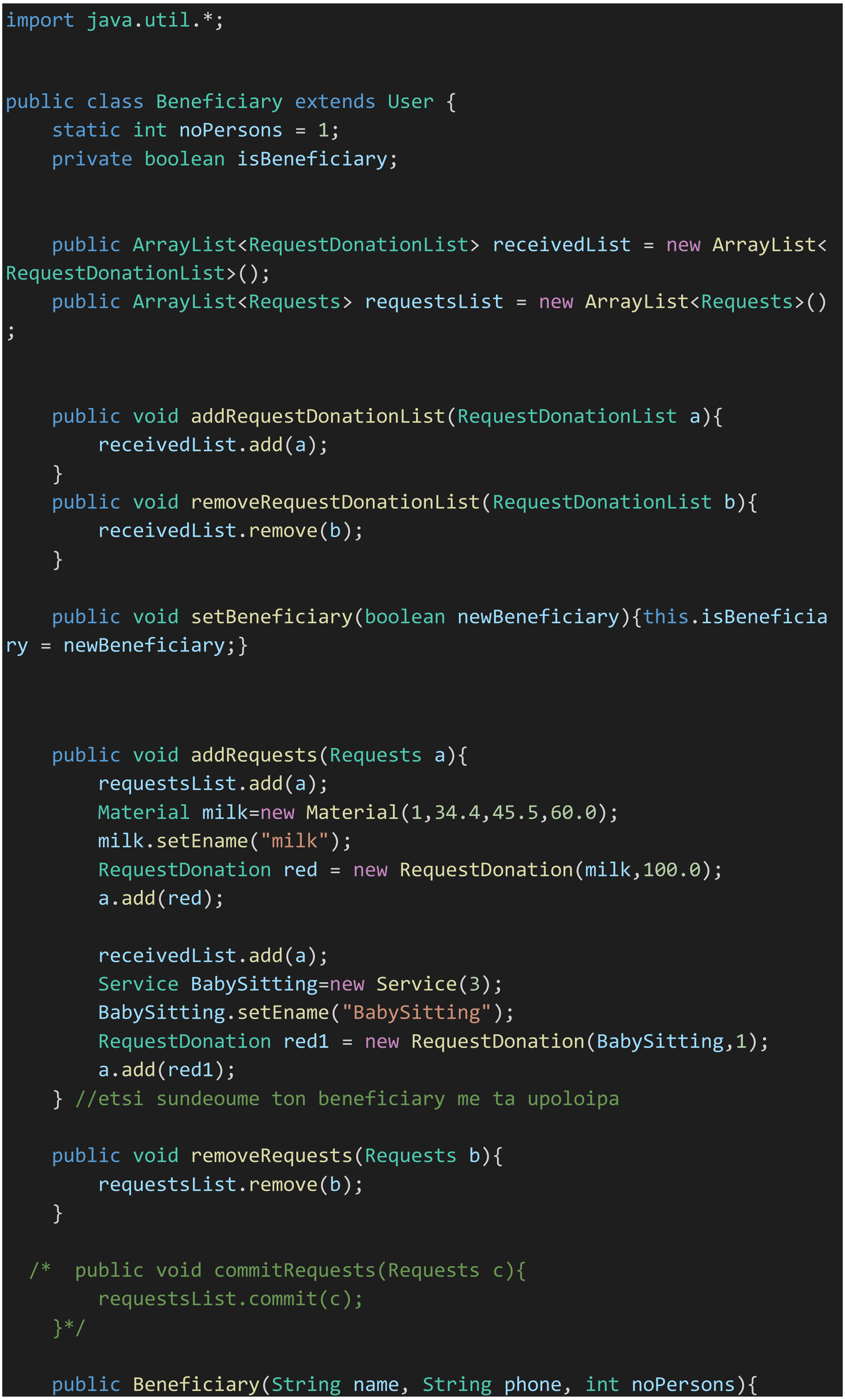
Ολοκληρωμένος κώδικας:

(Δεν είναι screenshots, εμφανίζονται έτσι μέσω του keep source formatting και μπορεί να γίνει αντιγραφή-επικόλληση, η εμφάνιση οφείλεται στο Visual Studio Code)

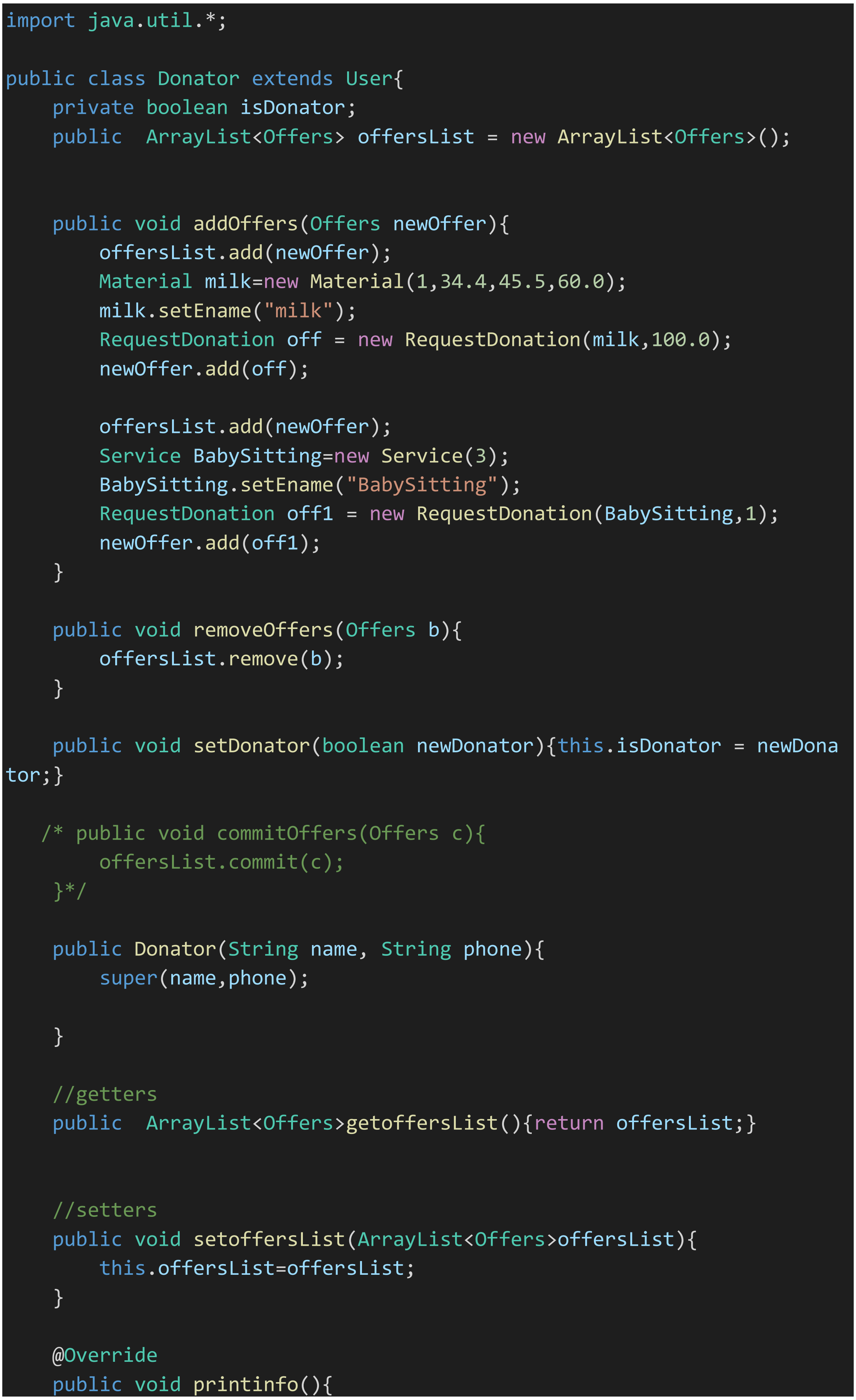
```java
public abstract class  User{
    private String name;
    private String phone;

    public User(String name, String phone){
        this.name=name;
        this.phone=phone;
    }

    //getters
    public String getName(){return name;}
    public String getPhone(){return phone;}

    //setters
    public void setName(String newName){
        this.name=newName;
    }
    public void setPhone(String newPhone){
        this.phone=newPhone;
    }

    @Override
    public String toString(){
        return String.format("User name: "+getName()+""+ "\n User phone
: " +getPhone()+ " " );
   }
    public void printinfo(){
        System.out.println("User name: "+getName()+""+ "\n User phone:
" +getPhone()+ " " );
    }
}
```



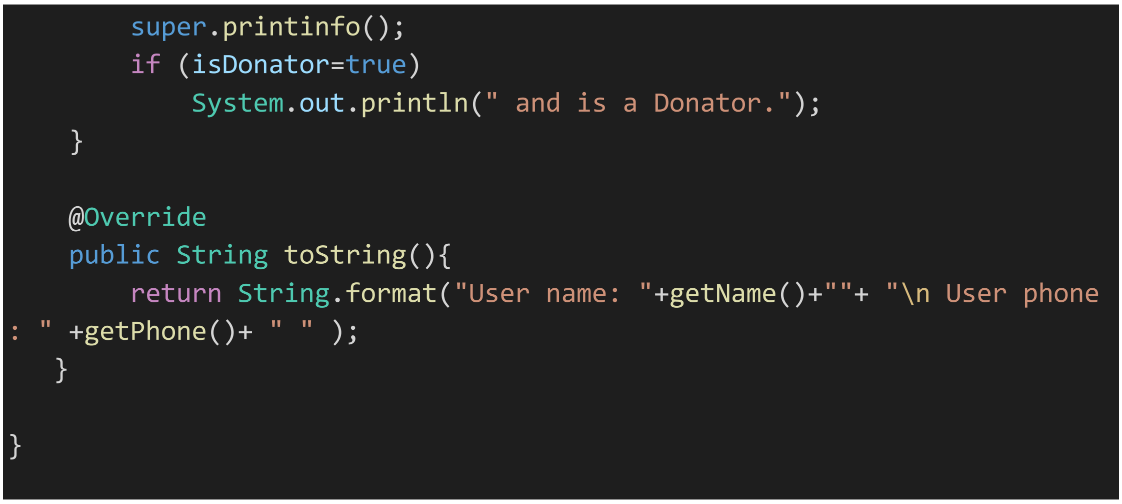
```
public class Admin extends User {
    protected boolean isAdmin = true;

    public Admin(String name, String phone, boolean isAdmin){
        super(name,phone);
        this.isAdmin = isAdmin;
    }

   /* //getter
    public boolean getAdmin(){return isAdmin;}
    //setter
    public void setAdmin(boolean newAdmin){
        this.isAdmin=newAdmin;
    }*/

    public void printinfo(){
        super.printinfo();

            System.out.println(" and is the Admin.");
    }

    @Override
    public String toString(){
        return String.format("User name: "+getName()+""+ "\n User phone
: " +getPhone()+ " " );
   }
}
```



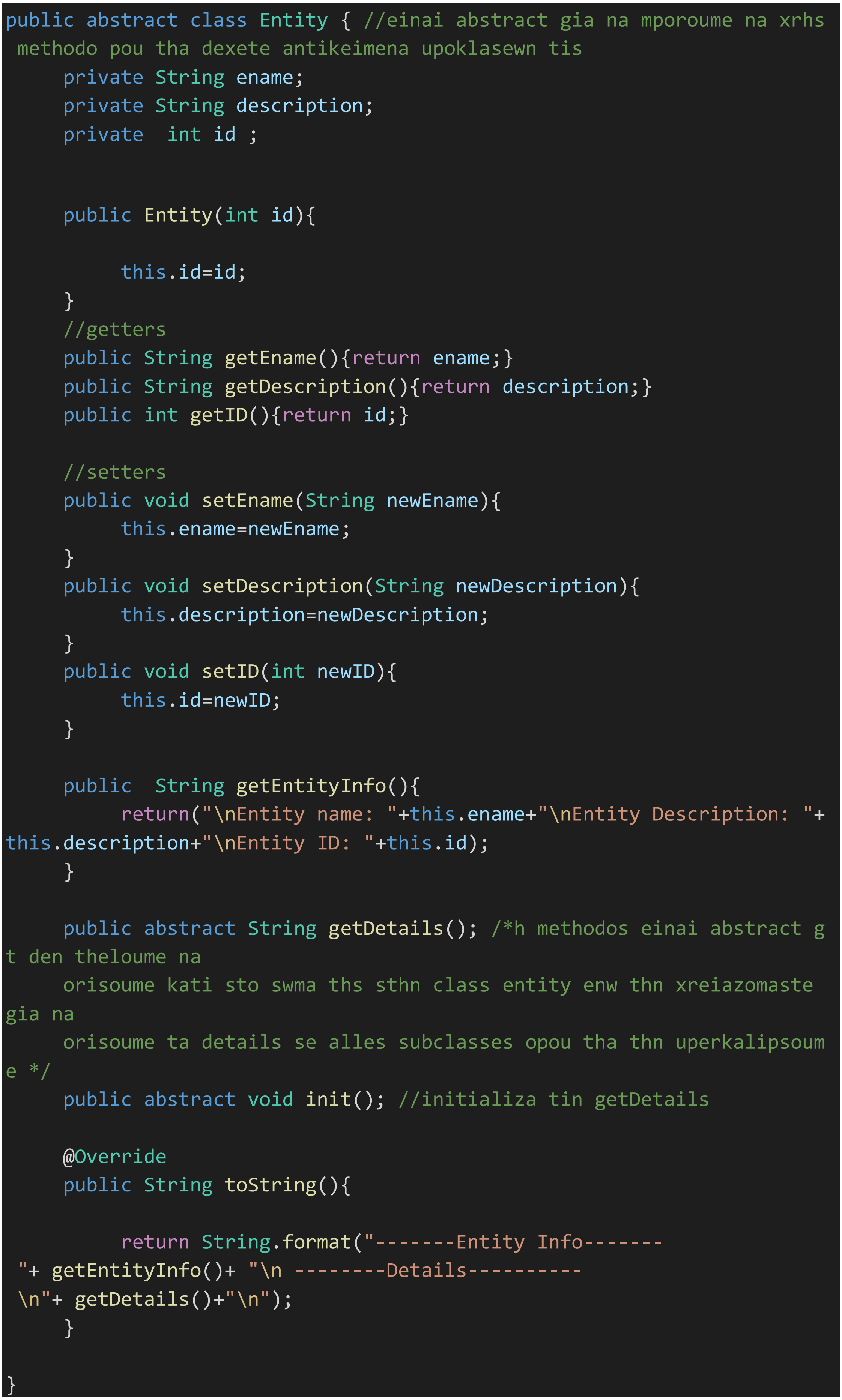
```
import java.util.*;


public class Beneficiary extends User {
    static int noPersons = 1;
    private boolean isBeneficiary;


    public ArrayList<RequestDonationList> receivedList = new ArrayList<
RequestDonationList>();
    public ArrayList<Requests> requestsList = new ArrayList<Requests>()
;


    public void addRequestDonationList(RequestDonationList a){
        receivedList.add(a);
    }
    public void removeRequestDonationList(RequestDonationList b){
        receivedList.remove(b);
    }

    public void setBeneficiary(boolean newBeneficiary){this.isBeneficia
ry = newBeneficiary;}



    public void addRequests(Requests a){
        requestsList.add(a);
        Material milk=new Material(1,34.4,45.5,60.0);
        milk.setEname("milk");
        RequestDonation red = new RequestDonation(milk,100.0);
        a.add(red);

        receivedList.add(a);
        Service BabySitting=new Service(3);
        BabySitting.setEname("BabySitting");
        RequestDonation red1 = new RequestDonation(BabySitting,1);
        a.add(red1);
    } //etsi sundeoume ton beneficiary me ta upoloipa

    public void removeRequests(Requests b){
        requestsList.remove(b);
    }

  /*  public void commitRequests(Requests c){
        requestsList.commit(c);
    }*/

    public Beneficiary(String name, String phone, int noPersons){
        super(name,phone);
        this.noPersons=noPersons;
    }

    //getters
    public ArrayList<RequestDonationList>getreceivedList(){return recei
vedList;}
    public ArrayList<Requests> getrequestsList(){return requestsList;}
    public static int getNoPersons(){return noPersons;}

    //setters
    public void setreceivedList(ArrayList<RequestDonationList>receivedL
ist){
        this.receivedList=receivedList;
    }
    public void setrequestsList(ArrayList<Requests>requestsList){
        this.requestsList=requestsList;
    }
    public void setNoPersons(int newNoPersons){
        this.noPersons=newNoPersons;
    }

    @Override
    public void printinfo(){
        super.printinfo();
        if (isBeneficiary=true)
            System.out.println(" and is a Beneficiary.");
    }

    @Override
    public String toString(){
        return String.format("User name: "+getName()+""+ "\n User phone
: " +getPhone()+ " " );
   }

}
```



```java
import java.util.*;

public class Donator extends User{
    private boolean isDonator;
    public  ArrayList<Offers> offersList = new ArrayList<Offers>();


    public void addOffers(Offers newOffer){
        offersList.add(newOffer);
        Material milk=new Material(1,34.4,45.5,60.0);
        milk.setEname("milk");
        RequestDonation off = new RequestDonation(milk,100.0);
        newOffer.add(off);

        offersList.add(newOffer);
        Service BabySitting=new Service(3);
        BabySitting.setEname("BabySitting");
        RequestDonation off1 = new RequestDonation(BabySitting,1);
        newOffer.add(off1);
    }

    public void removeOffers(Offers b){
        offersList.remove(b);
    }

    public void setDonator(boolean newDonator){this.isDonator = newDonator;}

   /* public void commitOffers(Offers c){
        offersList.commit(c);
    }*/

    public Donator(String name, String phone){
        super(name,phone);

    }

    //getters
    public  ArrayList<Offers>getoffersList(){return offersList;}


    //setters
    public void setoffersList(ArrayList<Offers>offersList){
        this.offersList=offersList;
    }

    @Override
    public void printinfo(){
```

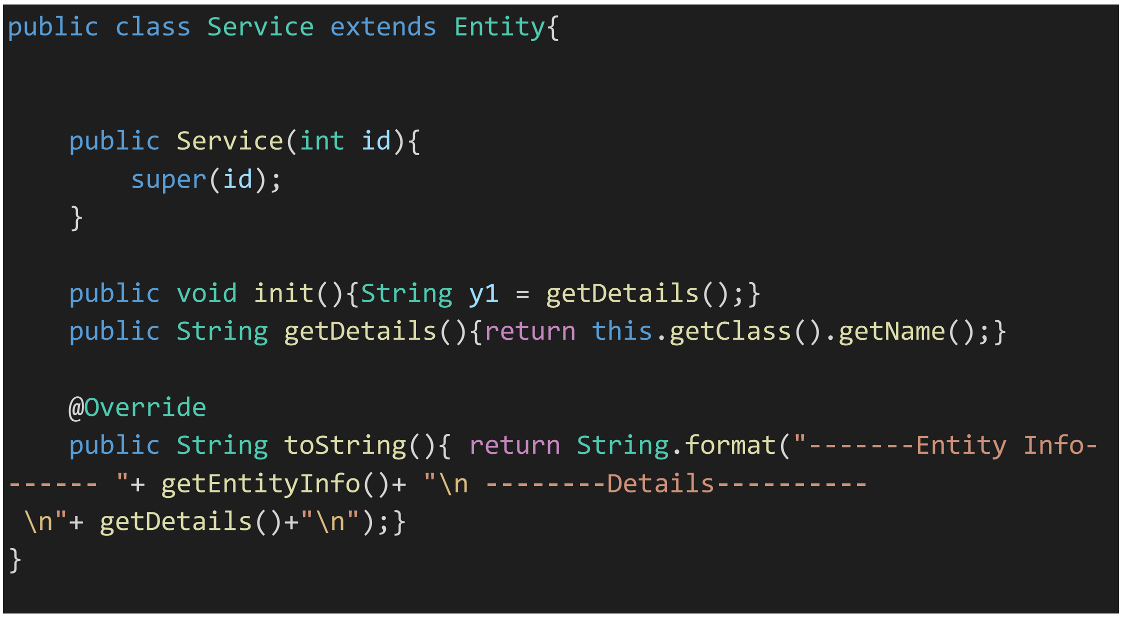


```
        super.printinfo();
        if (isDonator=true)
            System.out.println(" and is a Donator.");
    }

    @Override
    public String toString(){
        return String.format("User name: "+getName()+""+ "\n User phone
: " +getPhone()+ " " );
   }

}
```

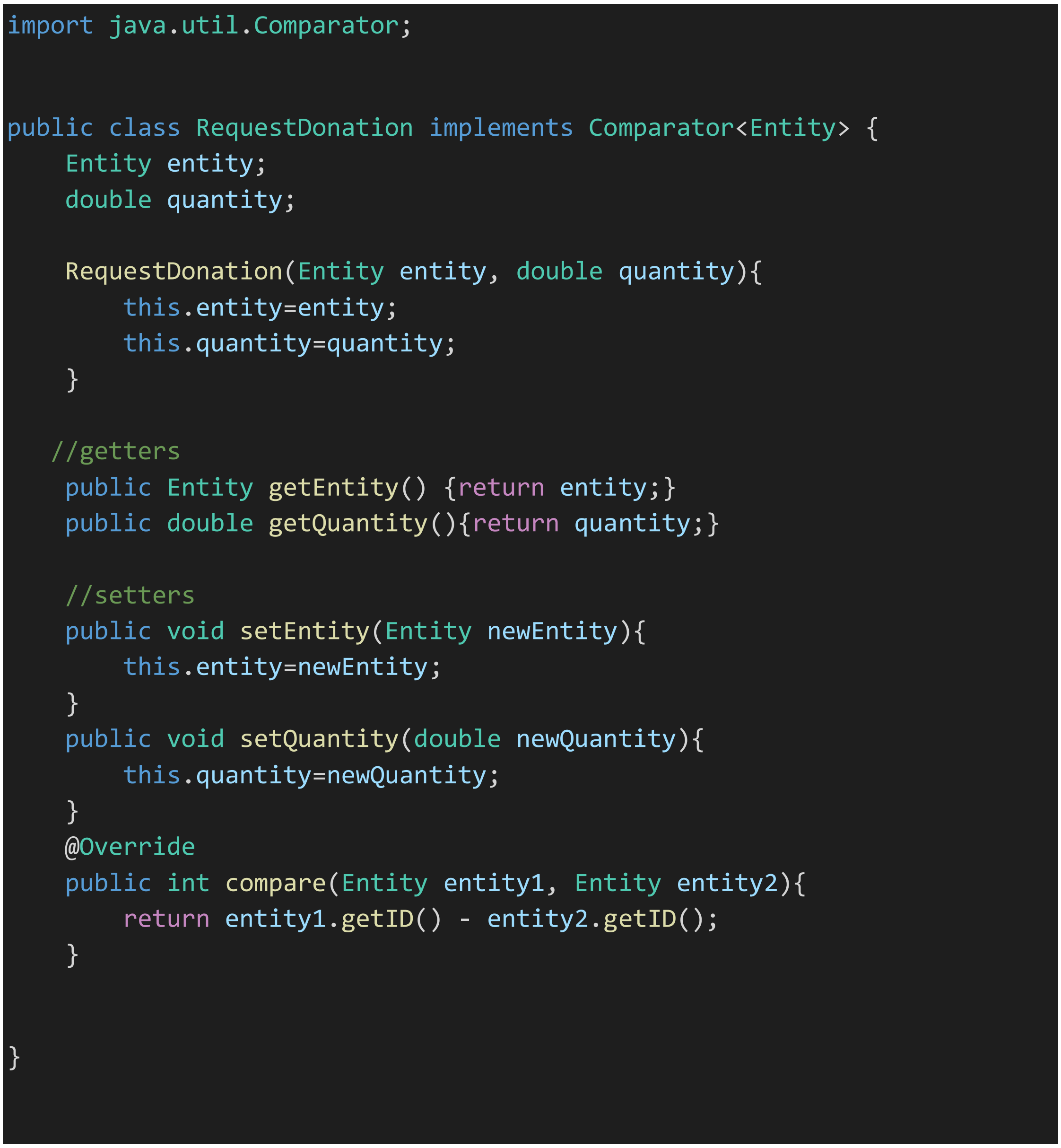


```
public abstract class Entity { //einai abstract gia na mporoume na xrhs
 methodo pou tha dexete antikeimena upoklasewn tis
     private String ename;
     private String description;
     private  int id ;


     public Entity(int id){

          this.id=id;
     }
     //getters
     public String getEname(){return ename;}
     public String getDescription(){return description;}
     public int getID(){return id;}

     //setters
     public void setEname(String newEname){
          this.ename=newEname;
     }
     public void setDescription(String newDescription){
          this.description=newDescription;
     }
     public void setID(int newID){
          this.id=newID;
     }

     public  String getEntityInfo(){
          return("\nEntity name: "+this.ename+"\nEntity Description: "+
this.description+"\nEntity ID: "+this.id);
     }

     public abstract String getDetails(); /*h methodos einai abstract g
t den theloume na
     orisoume kati sto swma ths sthn class entity enw thn xreiazomaste
gia na
     orisoume ta details se alles subclasses opou tha thn uperkalipsoum
e */
     public abstract void init(); //initializa tin getDetails

     @Override
     public String toString(){

          return String.format("-------Entity Info-------
 "+ getEntityInfo()+ "\n --------Details----------
 \n"+ getDetails()+"\n");
     }

}
```

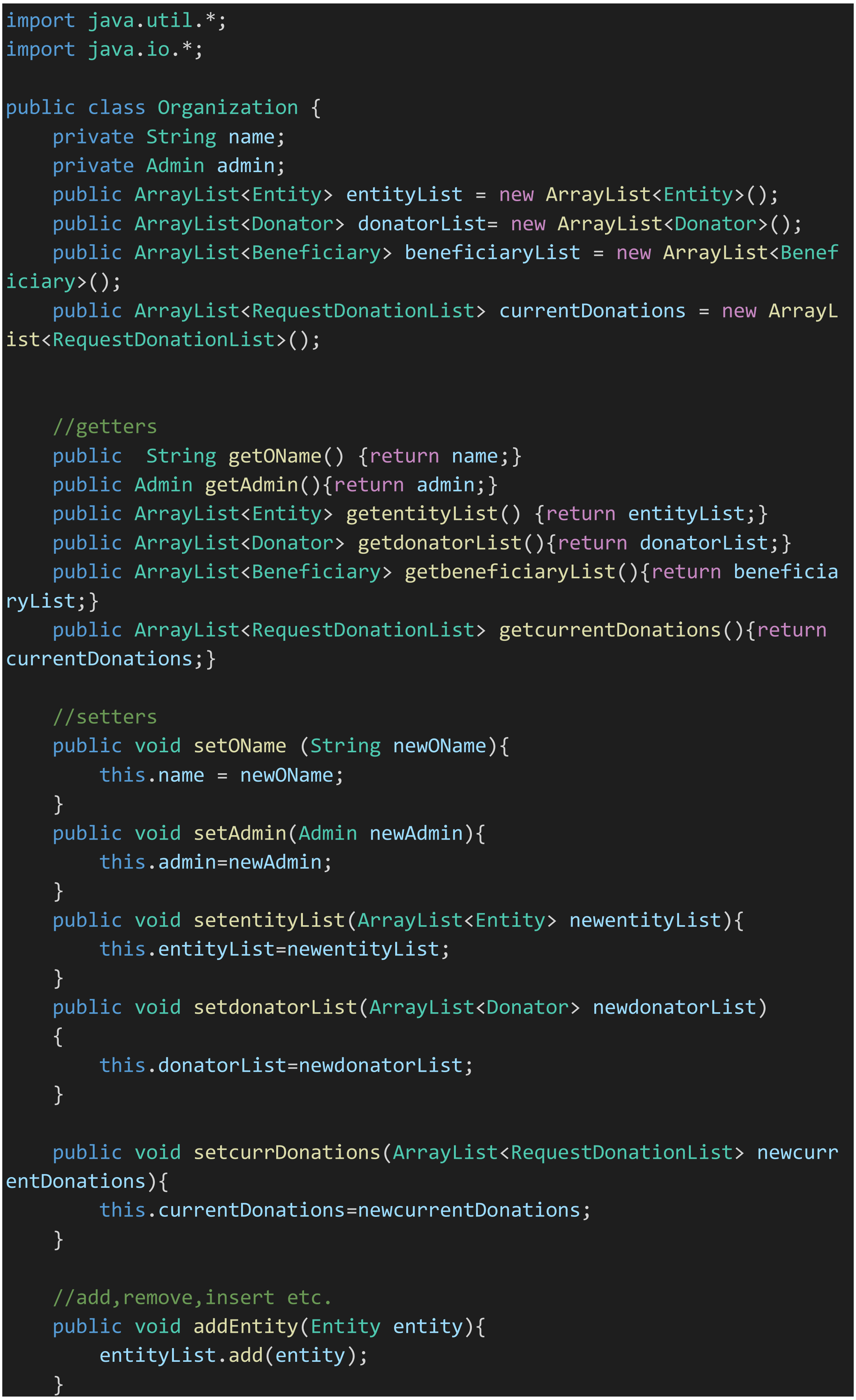
```java
public class Material extends Entity{
    double level1, level2, level3;

    public Material(int id, double level1, double level2, double level3
){
        super(id);
        this.level1 = level1;
        this.level2 = level2;
        this.level3 = level3;

    }

    //getters
    public double getLvl1(){return level1;}
    public double getLvl2(){return level2;}
    public double getLvl3(){return level3;}

    //setters
    public void setLvl1(double newLvl1){
        this.level1=newLvl1;
    }
    public void setLvl2(double newLvl2){
        this.level2=newLvl2;
    }
    public void setLvl3(double newLvl3){
        this.level3=newLvl3;
    }



    public void init(){String x1 = getDetails();}
    public String getDetails(){
        return this.getClass().getName() + "\n" + " 1 person  gets quan
tity Level1: " +this.level1 +
        "\n 2-
4 people get quantity Level 2: " + this.level2 + "\n 5 or more people g
et quantity Level 3: " + this.level3;


    }
    @Override
    public String toString(){
        return String.format("-------Entity Info-------
 "+ getEntityInfo()+ "\n --------Details----------
 \n"+ getDetails()+"\n");
    }
}
```



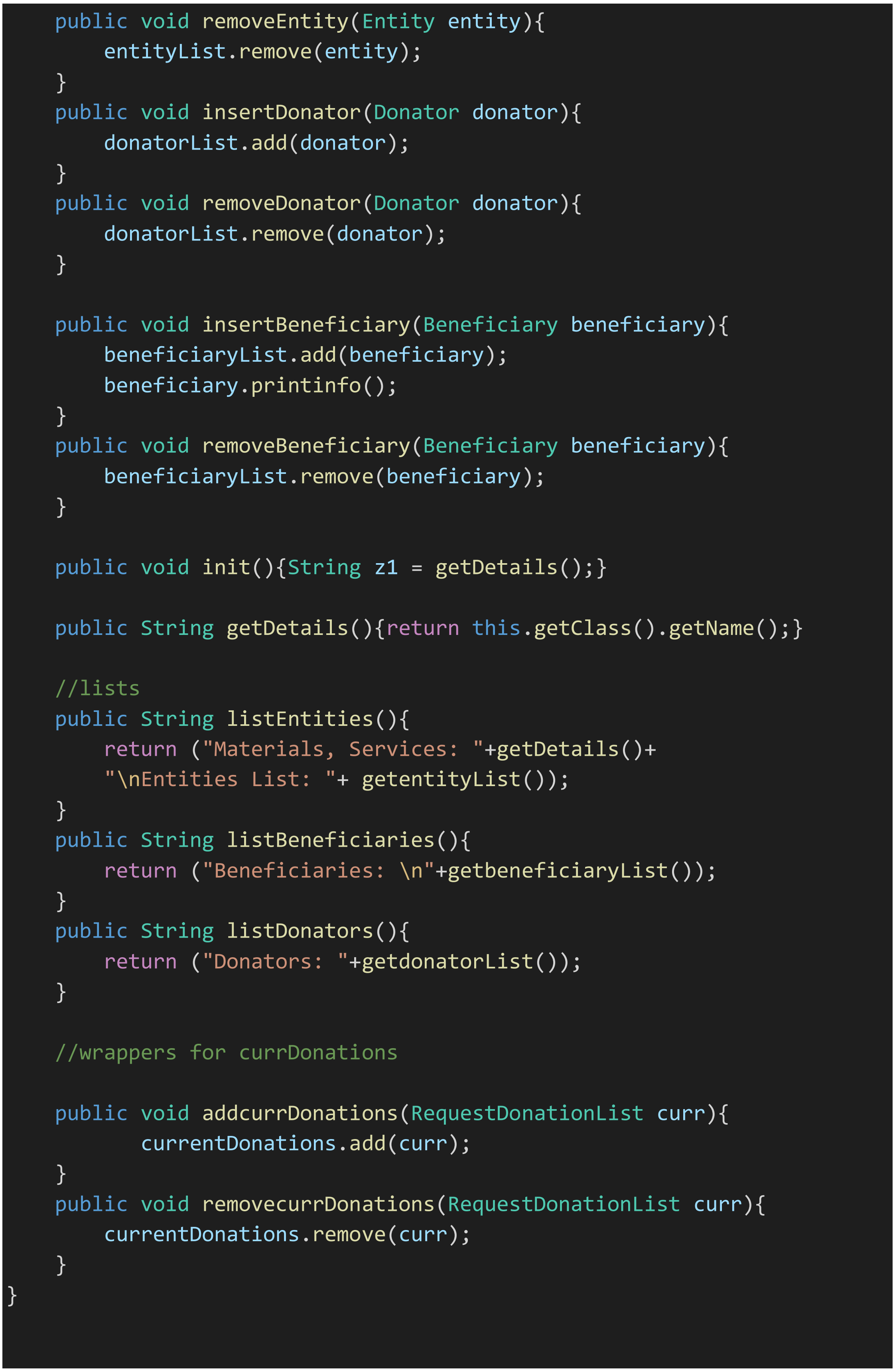
```
public class Service extends Entity{


    public Service(int id){
        super(id);
    }

    public void init(){String y1 = getDetails();}
    public String getDetails(){return this.getClass().getName();}

    @Override
    public String toString(){ return String.format("-------Entity Info-
------ "+ getEntityInfo()+ "\n --------Details----------
 \n"+ getDetails()+"\n");}
}
```



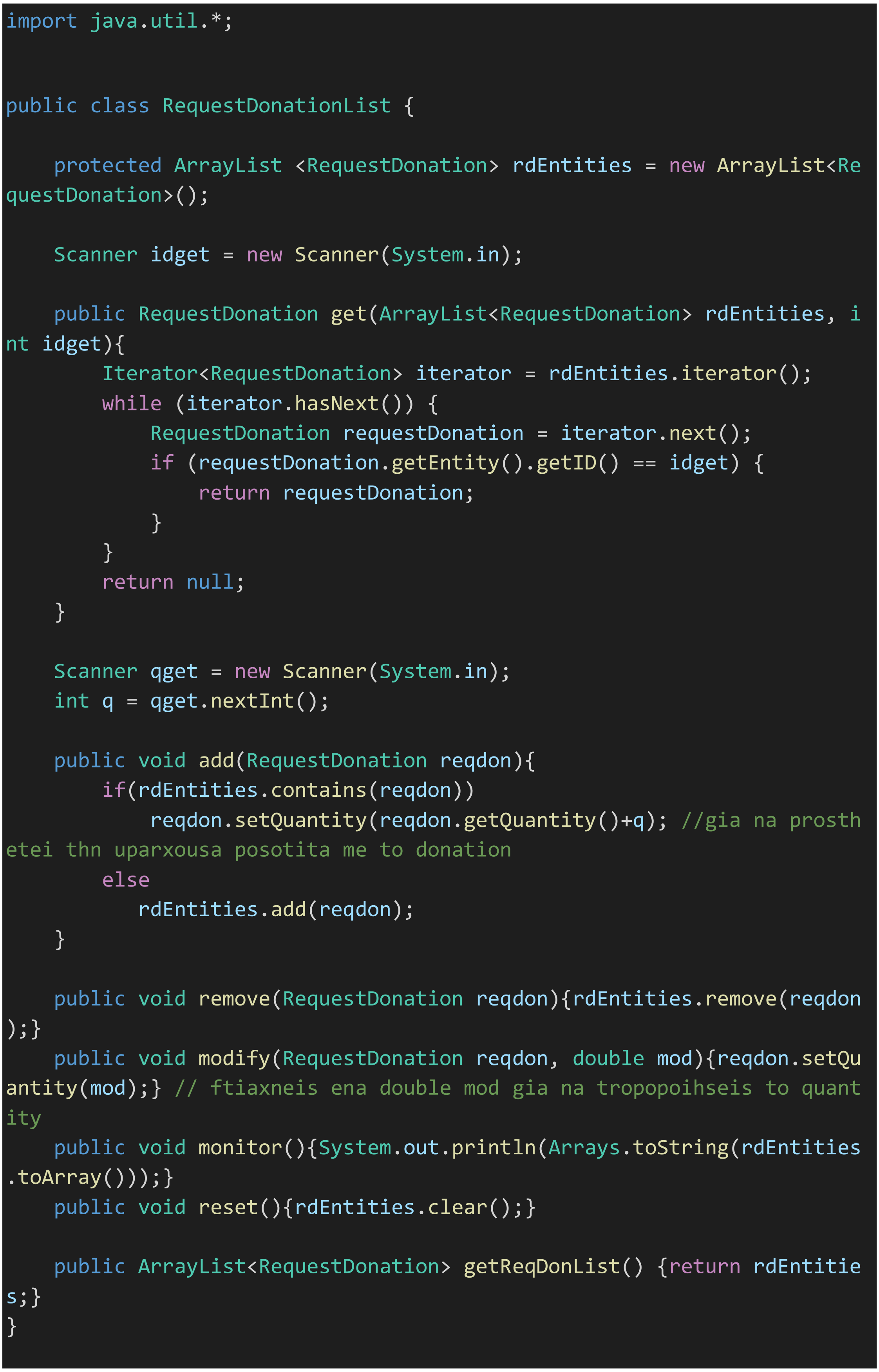
```java
import java.util.Comparator;


public class RequestDonation implements Comparator<Entity> {
    Entity entity;
    double quantity;

    RequestDonation(Entity entity, double quantity){
        this.entity=entity;
        this.quantity=quantity;
    }

   //getters
    public Entity getEntity() {return entity;}
    public double getQuantity(){return quantity;}

    //setters
    public void setEntity(Entity newEntity){
        this.entity=newEntity;
    }
    public void setQuantity(double newQuantity){
        this.quantity=newQuantity;
    }
    @Override
    public int compare(Entity entity1, Entity entity2){
        return entity1.getID() - entity2.getID();
    }


}
```



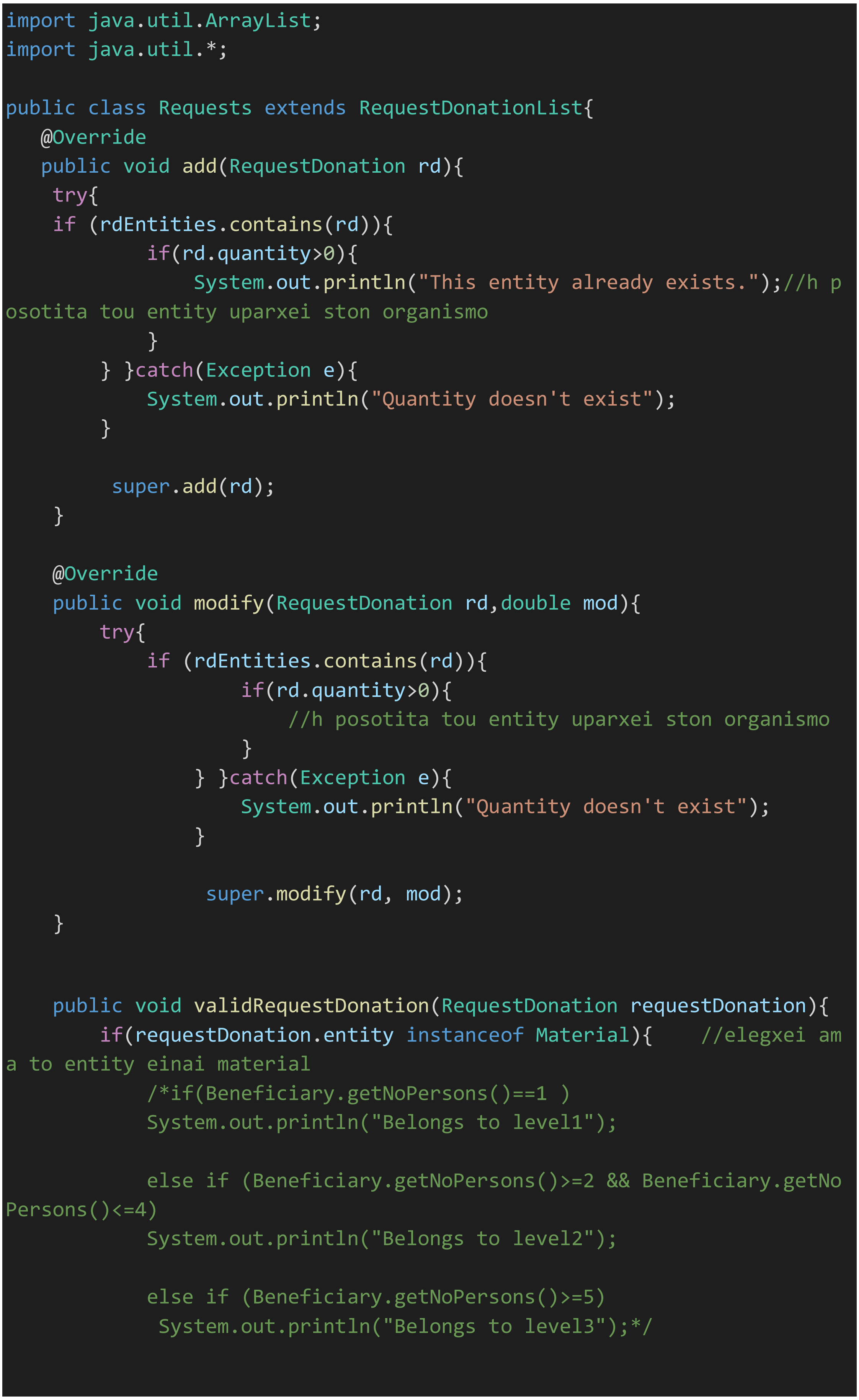
```java
import java.util.*;
import java.io.*;

public class Organization {
    private String name;
    private Admin admin;
    public ArrayList<Entity> entityList = new ArrayList<Entity>();
    public ArrayList<Donator> donatorList= new ArrayList<Donator>();
    public ArrayList<Beneficiary> beneficiaryList = new ArrayList<Beneficiary>();
    public ArrayList<RequestDonationList> currentDonations = new ArrayList<RequestDonationList>();


    //getters
    public  String getOName() {return name;}
    public Admin getAdmin(){return admin;}
    public ArrayList<Entity> getentityList() {return entityList;}
    public ArrayList<Donator> getdonatorList(){return donatorList;}
    public ArrayList<Beneficiary> getbeneficiaryList(){return beneficiaryList;}
    public ArrayList<RequestDonationList> getcurrentDonations(){return currentDonations;}

    //setters
    public void setOName (String newOName){
        this.name = newOName;
    }
    public void setAdmin(Admin newAdmin){
        this.admin=newAdmin;
    }
    public void setentityList(ArrayList<Entity> newentityList){
        this.entityList=newentityList;
    }
    public void setdonatorList(ArrayList<Donator> newdonatorList)
    {
        this.donatorList=newdonatorList;
    }

    public void setcurrDonations(ArrayList<RequestDonationList> newcurrentDonations){
        this.currentDonations=newcurrentDonations;
    }

    //add,remove,insert etc.
    public void addEntity(Entity entity){
        entityList.add(entity);
    }
```



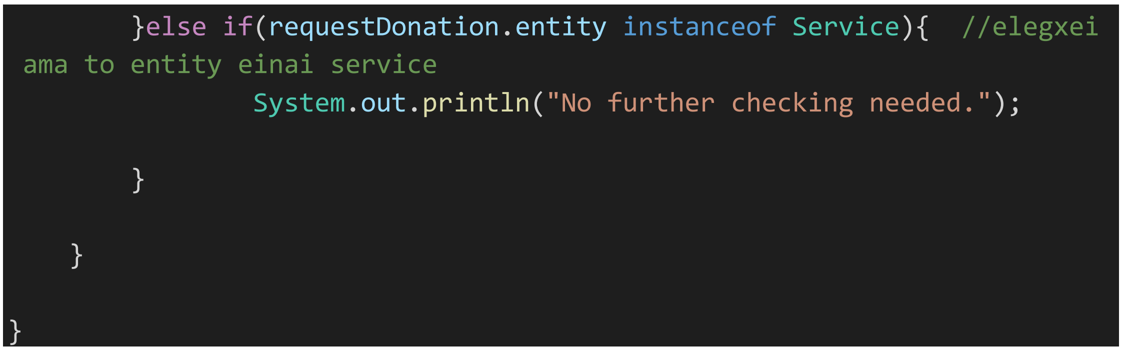
```
    public void removeEntity(Entity entity){
        entityList.remove(entity);
    }
    public void insertDonator(Donator donator){
        donatorList.add(donator);
    }
    public void removeDonator(Donator donator){
        donatorList.remove(donator);
    }

    public void insertBeneficiary(Beneficiary beneficiary){
        beneficiaryList.add(beneficiary);
        beneficiary.printinfo();
    }
    public void removeBeneficiary(Beneficiary beneficiary){
        beneficiaryList.remove(beneficiary);
    }

    public void init(){String z1 = getDetails();}

    public String getDetails(){return this.getClass().getName();}

    //lists
    public String listEntities(){
        return ("Materials, Services: "+getDetails()+
        "\nEntities List: "+ getentityList());
    }
    public String listBeneficiaries(){
        return ("Beneficiaries: \n"+getbeneficiaryList());
    }
    public String listDonators(){
        return ("Donators: "+getdonatorList());
    }

    //wrappers for currDonations

    public void addcurrDonations(RequestDonationList curr){
           currentDonations.add(curr);
    }
    public void removecurrDonations(RequestDonationList curr){
        currentDonations.remove(curr);
    }
}
```



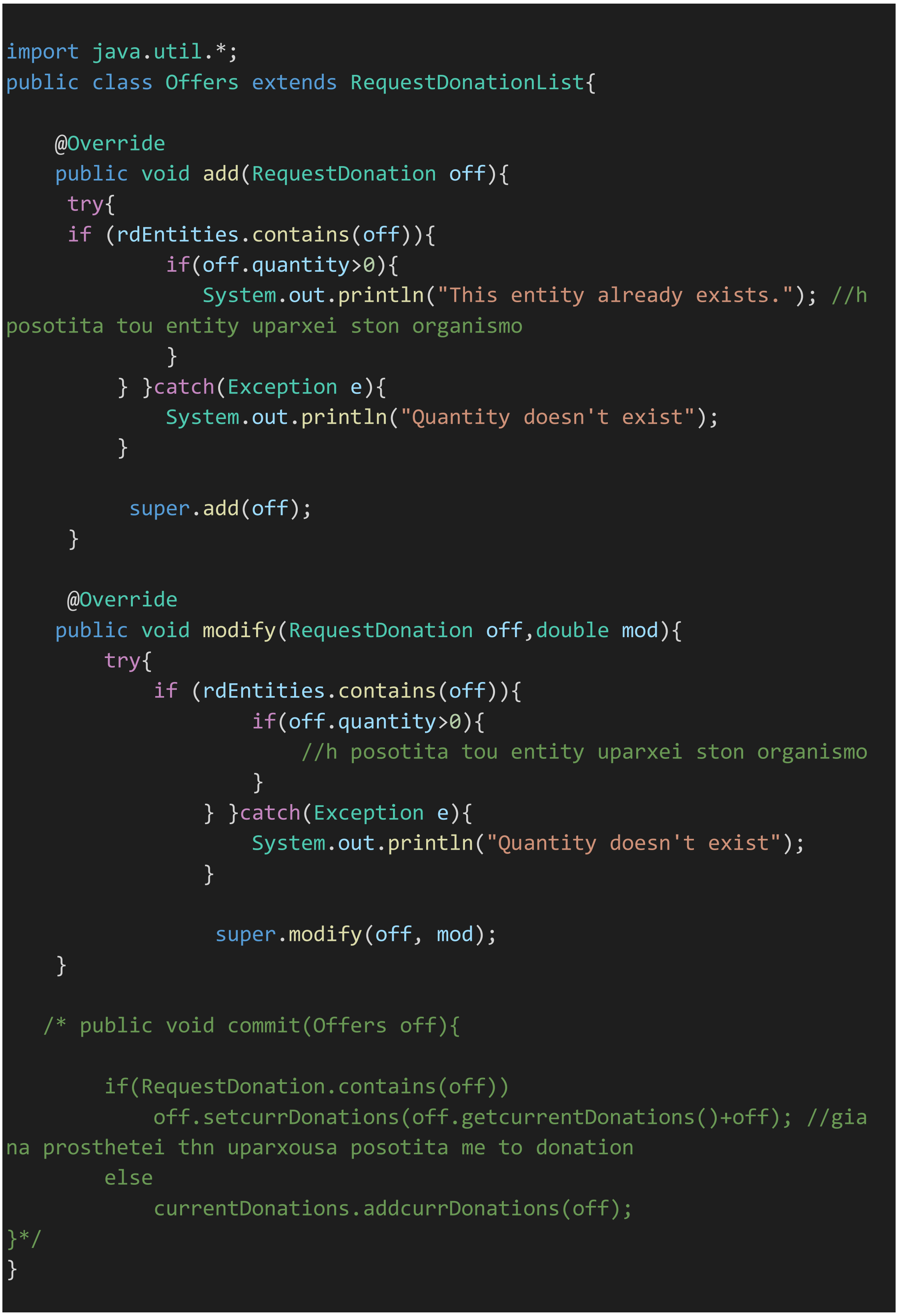
```
import java.util.*;


public class RequestDonationList {

    protected ArrayList <RequestDonation> rdEntities = new ArrayList<RequestDonation>();

    Scanner idget = new Scanner(System.in);

    public RequestDonation get(ArrayList<RequestDonation> rdEntities, int idget){
        Iterator<RequestDonation> iterator = rdEntities.iterator();
        while (iterator.hasNext()) {
            RequestDonation requestDonation = iterator.next();
            if (requestDonation.getEntity().getID() == idget) {
                return requestDonation;
            }
        }
        return null;
    }

    Scanner qget = new Scanner(System.in);
    int q = qget.nextInt();

    public void add(RequestDonation reqdon){
        if(rdEntities.contains(reqdon))
            reqdon.setQuantity(reqdon.getQuantity()+q); //gia na prosthetei thn uparxousa posotita me to donation
        else
           rdEntities.add(reqdon);
    }

    public void remove(RequestDonation reqdon){rdEntities.remove(reqdon);}
    public void modify(RequestDonation reqdon, double mod){reqdon.setQuantity(mod);} // ftiaxneis ena double mod gia na tropopoihseis to quantity
    public void monitor(){System.out.println(Arrays.toString(rdEntities.toArray()));}
    public void reset(){rdEntities.clear();}

    public ArrayList<RequestDonation> getReqDonList() {return rdEntities;}
}
```



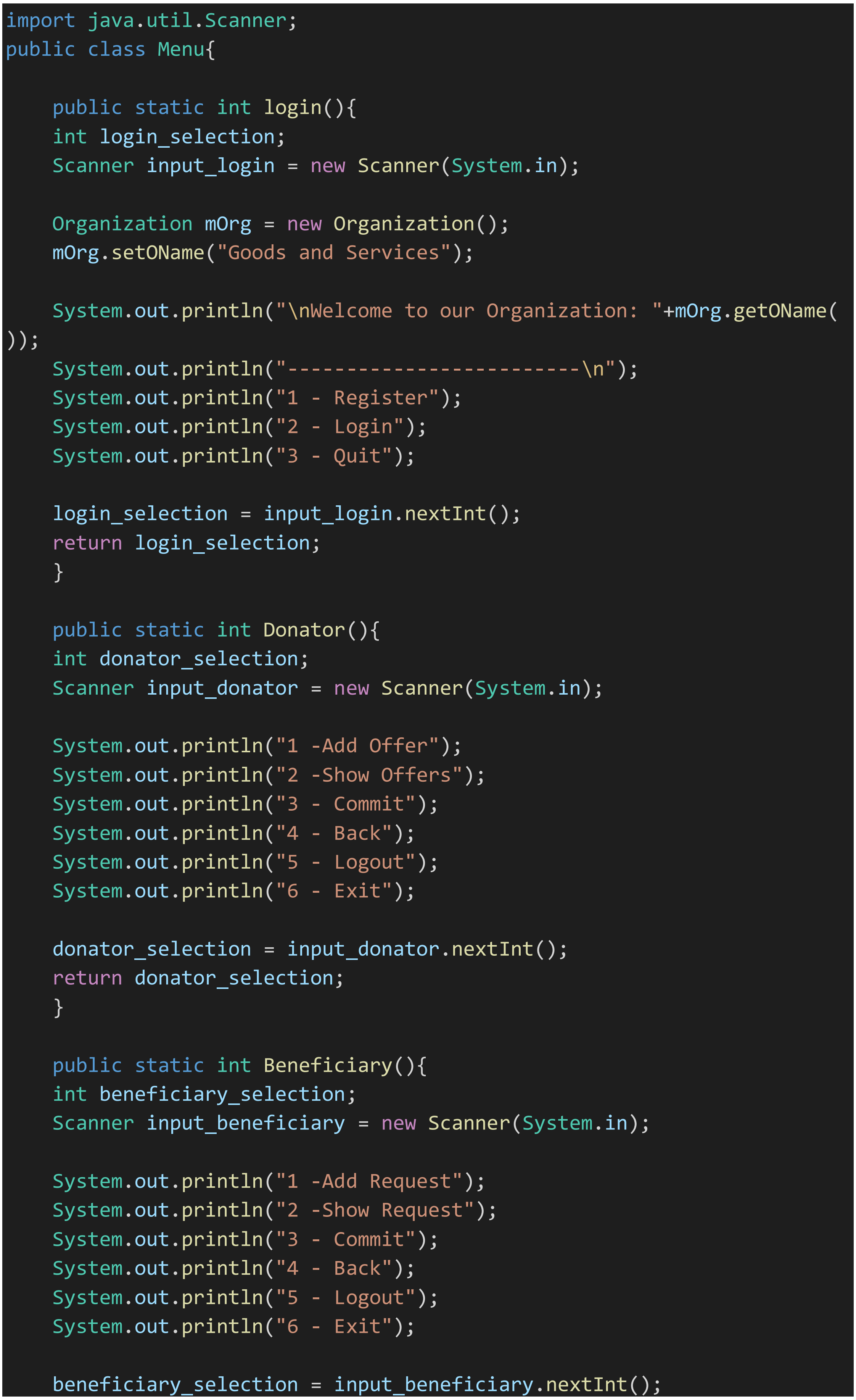
```java
import java.util.ArrayList;
import java.util.*;

public class Requests extends RequestDonationList{
   @Override
   public void add(RequestDonation rd){
    try{
    if (rdEntities.contains(rd)){
            if(rd.quantity>0){
                System.out.println("This entity already exists.");//h p
osotita tou entity uparxei ston organismo
            }
        } }catch(Exception e){
            System.out.println("Quantity doesn't exist");
        }

         super.add(rd);
    }

    @Override
    public void modify(RequestDonation rd,double mod){
        try{
            if (rdEntities.contains(rd)){
                    if(rd.quantity>0){
                        //h posotita tou entity uparxei ston organismo
                    }
                } }catch(Exception e){
                    System.out.println("Quantity doesn't exist");
                }

                 super.modify(rd, mod);
    }


    public void validRequestDonation(RequestDonation requestDonation){
        if(requestDonation.entity instanceof Material){    //elegxei am
a to entity einai material
            /*if(Beneficiary.getNoPersons()==1 )
            System.out.println("Belongs to level1");

            else if (Beneficiary.getNoPersons()>=2 && Beneficiary.getNo
Persons()<=4)
            System.out.println("Belongs to level2");

            else if (Beneficiary.getNoPersons()>=5)
             System.out.println("Belongs to level3");*/
```



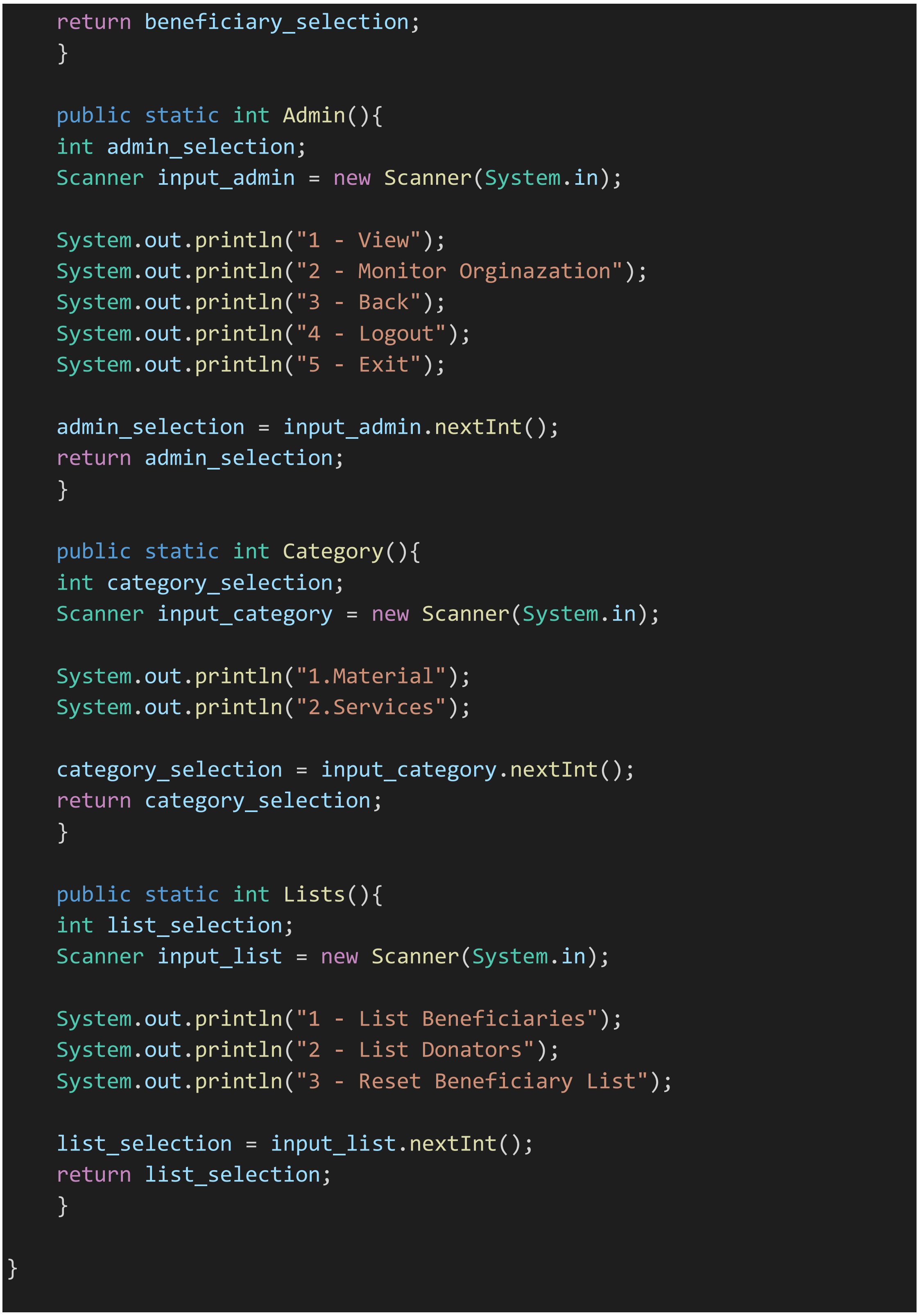
```
        }else if(requestDonation.entity instanceof Service){  //elegxei
 ama to entity einai service
                System.out.println("No further checking needed.");

        }

    }

}
```



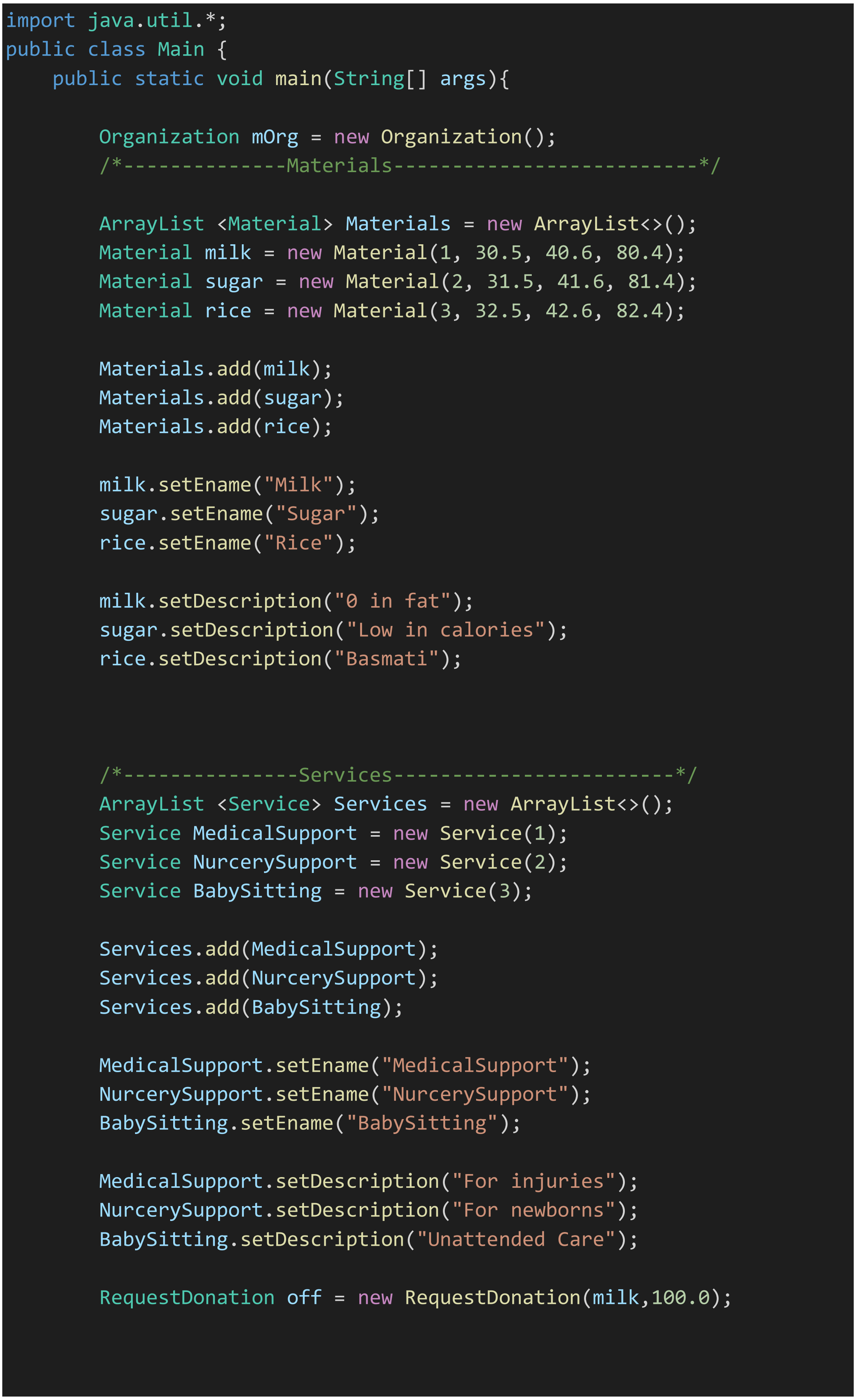
```
import java.util.*;
public class Offers extends RequestDonationList{

    @Override
    public void add(RequestDonation off){
     try{
     if (rdEntities.contains(off)){
             if(off.quantity>0){
                System.out.println("This entity already exists."); //h
posotita tou entity uparxei ston organismo
             }
         } }catch(Exception e){
             System.out.println("Quantity doesn't exist");
         }

          super.add(off);
     }

     @Override
    public void modify(RequestDonation off,double mod){
        try{
            if (rdEntities.contains(off)){
                    if(off.quantity>0){
                        //h posotita tou entity uparxei ston organismo
                    }
                } }catch(Exception e){
                    System.out.println("Quantity doesn't exist");
                }

                 super.modify(off, mod);
    }

   /* public void commit(Offers off){

        if(RequestDonation.contains(off))
            off.setcurrDonations(off.getcurrentDonations()+off); //gia
na prosthetei thn uparxousa posotita me to donation
        else
            currentDonations.addcurrDonations(off);
}*/
}
```



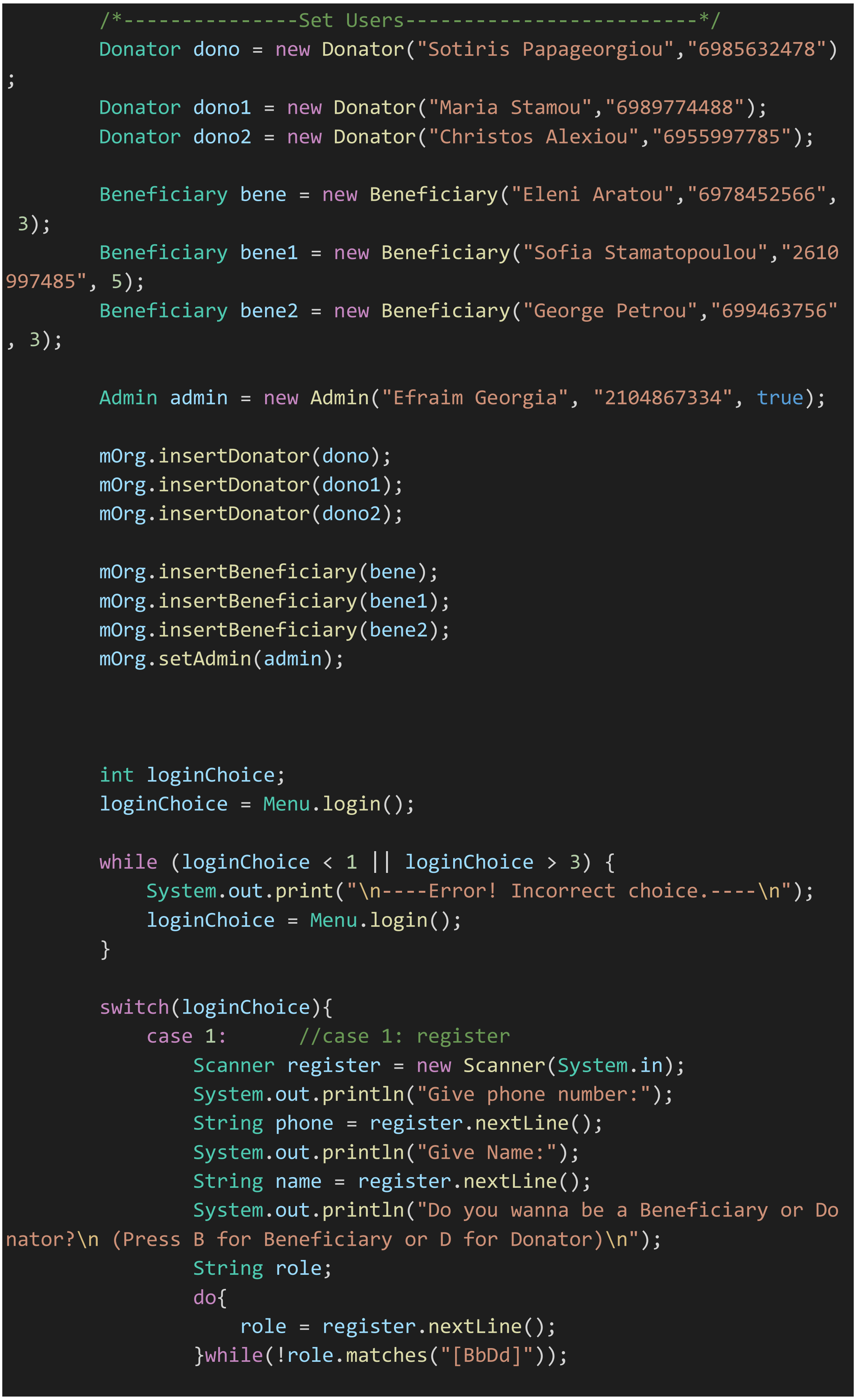
```java
import java.util.Scanner;
public class Menu{

    public static int login(){
    int login_selection;
    Scanner input_login = new Scanner(System.in);

    Organization mOrg = new Organization();
    mOrg.setOName("Goods and Services");

    System.out.println("\nWelcome to our Organization: "+mOrg.getOName(
));
    System.out.println("-------------------------\n");
    System.out.println("1 - Register");
    System.out.println("2 - Login");
    System.out.println("3 - Quit");

    login_selection = input_login.nextInt();
    return login_selection;
    }

    public static int Donator(){
    int donator_selection;
    Scanner input_donator = new Scanner(System.in);

    System.out.println("1 -Add Offer");
    System.out.println("2 -Show Offers");
    System.out.println("3 - Commit");
    System.out.println("4 - Back");
    System.out.println("5 - Logout");
    System.out.println("6 - Exit");

    donator_selection = input_donator.nextInt();
    return donator_selection;
    }

    public static int Beneficiary(){
    int beneficiary_selection;
    Scanner input_beneficiary = new Scanner(System.in);

    System.out.println("1 -Add Request");
    System.out.println("2 -Show Request");
    System.out.println("3 - Commit");
    System.out.println("4 - Back");
    System.out.println("5 - Logout");
    System.out.println("6 - Exit");

    beneficiary_selection = input_beneficiary.nextInt();
```



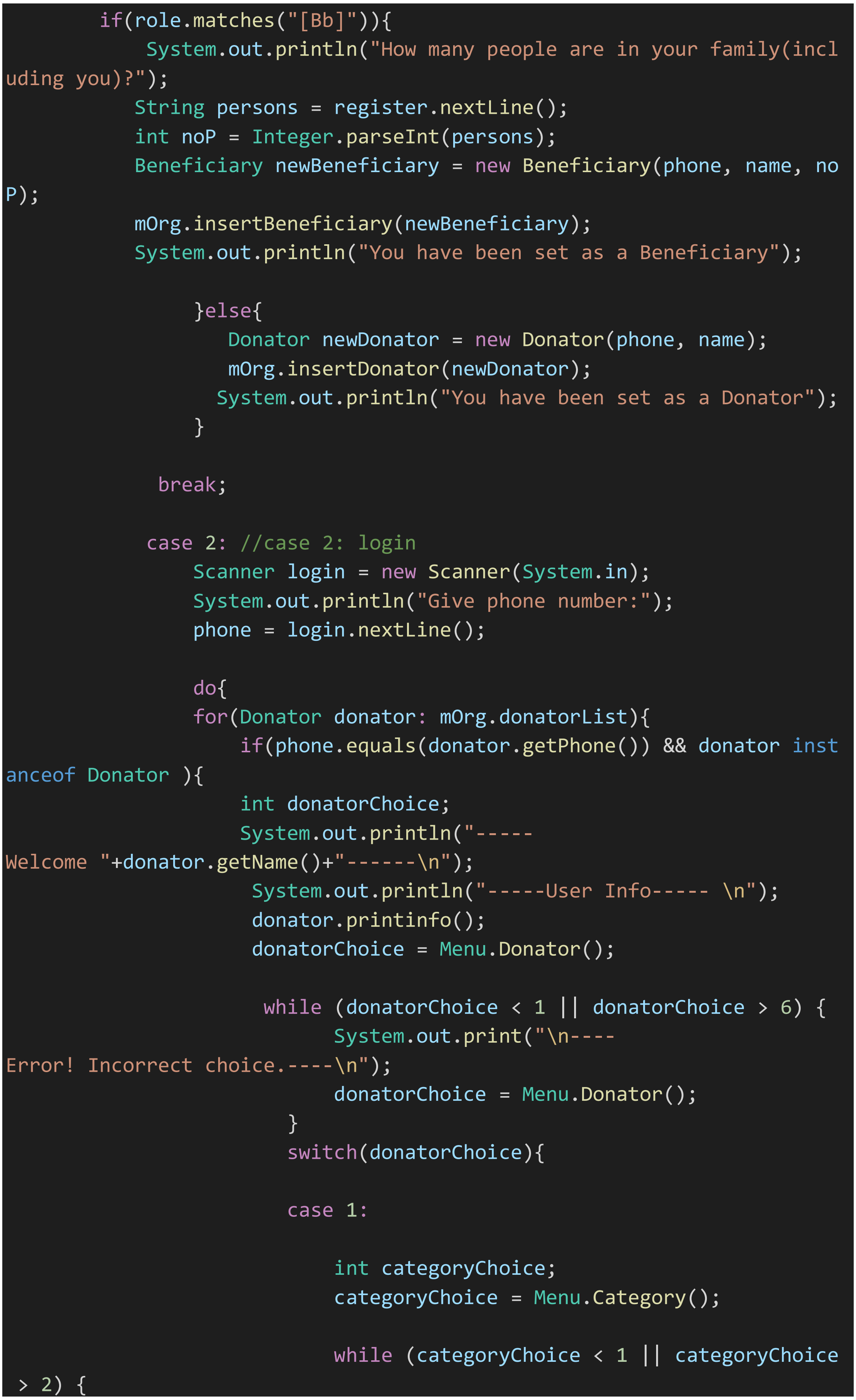
```
    return beneficiary_selection;
    }

    public static int Admin(){
    int admin_selection;
    Scanner input_admin = new Scanner(System.in);

    System.out.println("1 - View");
    System.out.println("2 - Monitor Orginazation");
    System.out.println("3 - Back");
    System.out.println("4 - Logout");
    System.out.println("5 - Exit");

    admin_selection = input_admin.nextInt();
    return admin_selection;
    }

    public static int Category(){
    int category_selection;
    Scanner input_category = new Scanner(System.in);

    System.out.println("1.Material");
    System.out.println("2.Services");

    category_selection = input_category.nextInt();
    return category_selection;
    }

    public static int Lists(){
    int list_selection;
    Scanner input_list = new Scanner(System.in);

    System.out.println("1 - List Beneficiaries");
    System.out.println("2 - List Donators");
    System.out.println("3 - Reset Beneficiary List");

    list_selection = input_list.nextInt();
    return list_selection;
    }

}
```



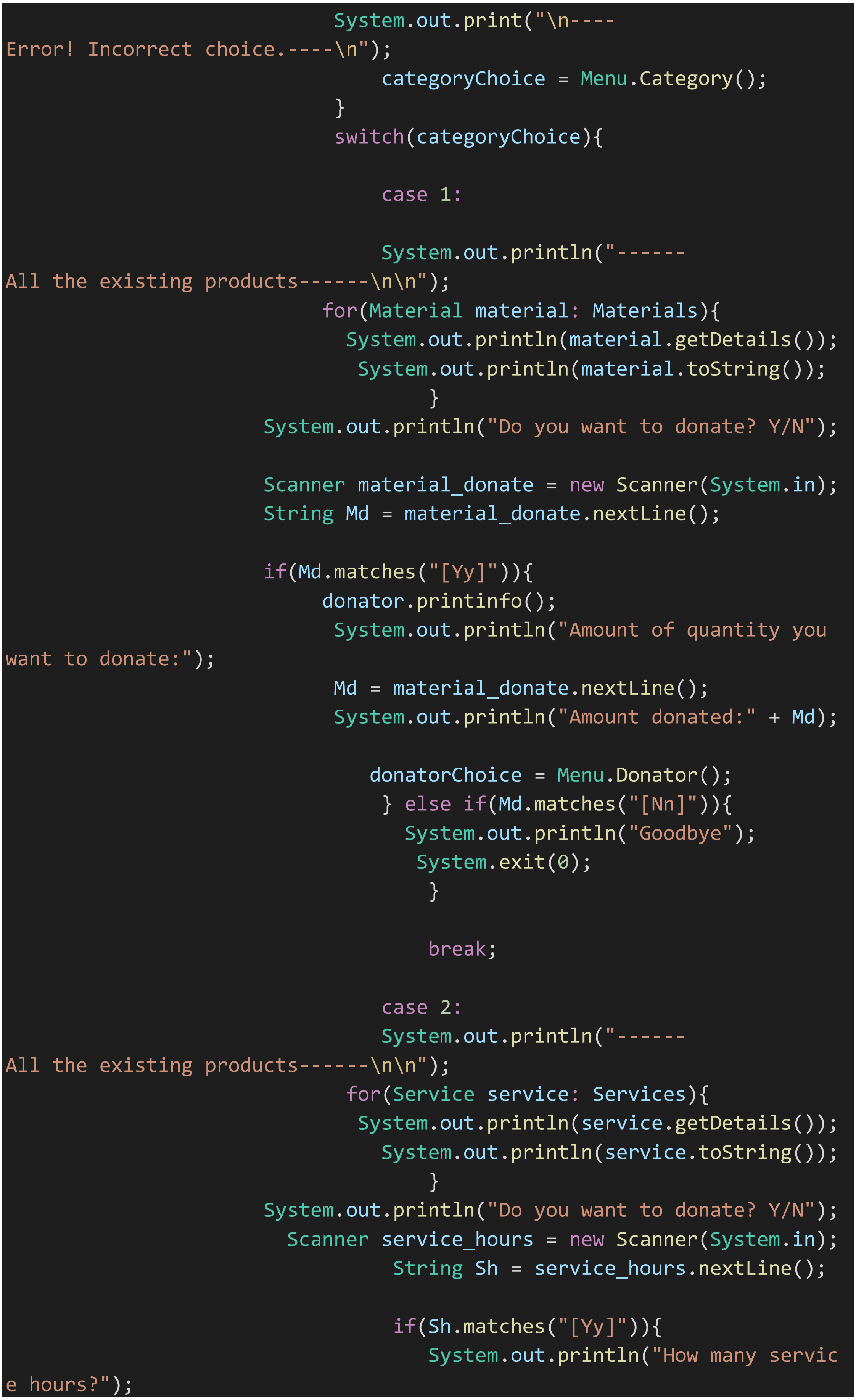
```
import java.util.*;
public class Main {
    public static void main(String[] args){

        Organization mOrg = new Organization();
        /*--------------Materials--------------------------*/

        ArrayList <Material> Materials = new ArrayList<>();
        Material milk = new Material(1, 30.5, 40.6, 80.4);
        Material sugar = new Material(2, 31.5, 41.6, 81.4);
        Material rice = new Material(3, 32.5, 42.6, 82.4);

        Materials.add(milk);
        Materials.add(sugar);
        Materials.add(rice);

        milk.setEname("Milk");
        sugar.setEname("Sugar");
        rice.setEname("Rice");

        milk.setDescription("0 in fat");
        sugar.setDescription("Low in calories");
        rice.setDescription("Basmati");



        /*---------------Services------------------------*/
        ArrayList <Service> Services = new ArrayList<>();
        Service MedicalSupport = new Service(1);
        Service NurcerySupport = new Service(2);
        Service BabySitting = new Service(3);

        Services.add(MedicalSupport);
        Services.add(NurcerySupport);
        Services.add(BabySitting);

        MedicalSupport.setEname("MedicalSupport");
        NurcerySupport.setEname("NurcerySupport");
        BabySitting.setEname("BabySitting");

        MedicalSupport.setDescription("For injuries");
        NurcerySupport.setDescription("For newborns");
        BabySitting.setDescription("Unattended Care");

        RequestDonation off = new RequestDonation(milk,100.0);
```



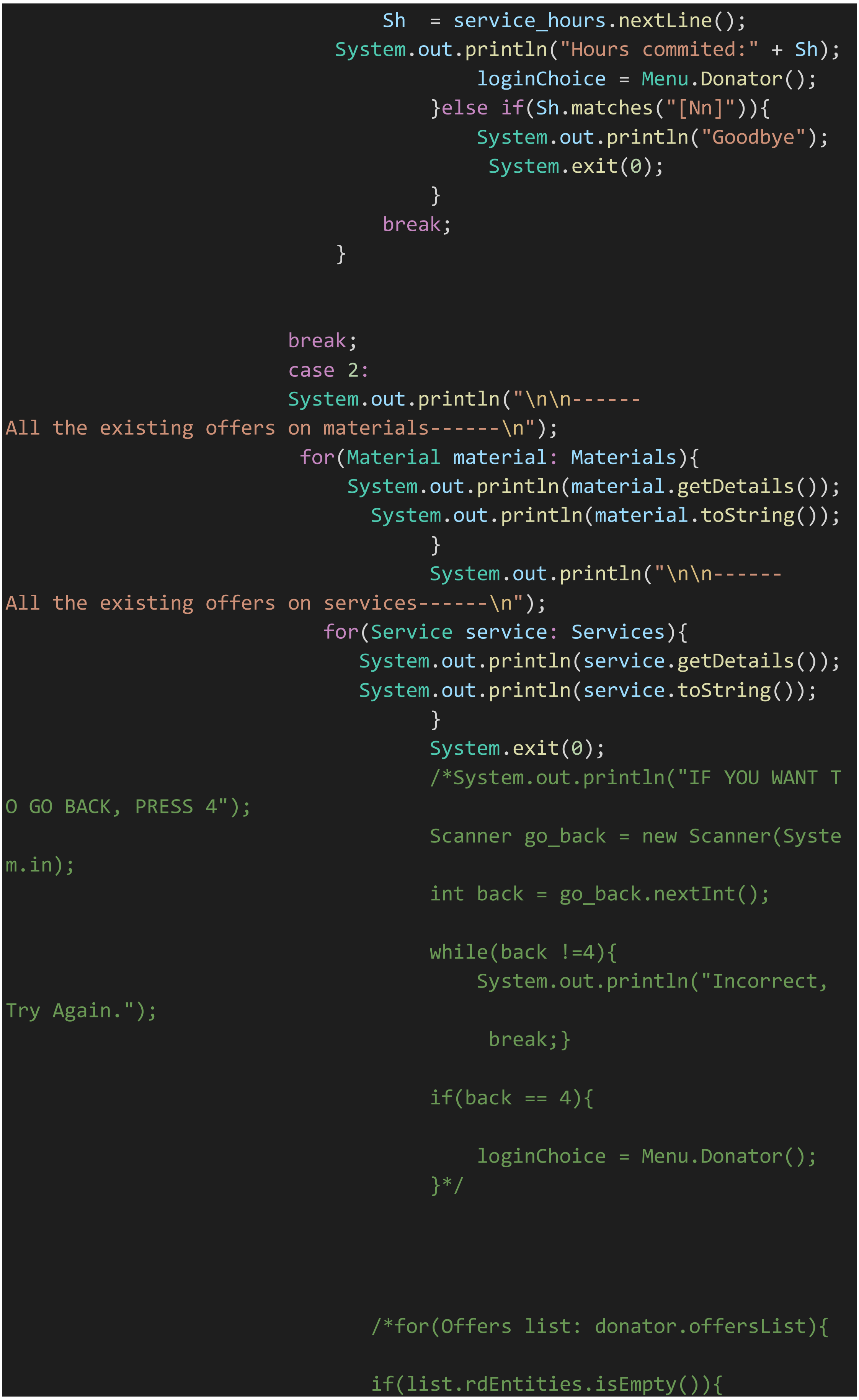
```
        /*---------------Set Users------------------------*/
        Donator dono = new Donator("Sotiris Papageorgiou","6985632478")
;
        Donator dono1 = new Donator("Maria Stamou","6989774488");
        Donator dono2 = new Donator("Christos Alexiou","6955997785");

        Beneficiary bene = new Beneficiary("Eleni Aratou","6978452566",
 3);
        Beneficiary bene1 = new Beneficiary("Sofia Stamatopoulou","2610
997485", 5);
        Beneficiary bene2 = new Beneficiary("George Petrou","699463756"
, 3);

        Admin admin = new Admin("Efraim Georgia", "2104867334", true);

        mOrg.insertDonator(dono);
        mOrg.insertDonator(dono1);
        mOrg.insertDonator(dono2);

        mOrg.insertBeneficiary(bene);
        mOrg.insertBeneficiary(bene1);
        mOrg.insertBeneficiary(bene2);
        mOrg.setAdmin(admin);



        int loginChoice;
        loginChoice = Menu.login();

        while (loginChoice < 1 || loginChoice > 3) {
            System.out.print("\n----Error! Incorrect choice.----\n");
            loginChoice = Menu.login();
        }

        switch(loginChoice){
            case 1:      //case 1: register
                Scanner register = new Scanner(System.in);
                System.out.println("Give phone number:");
                String phone = register.nextLine();
                System.out.println("Give Name:");
                String name = register.nextLine();
                System.out.println("Do you wanna be a Beneficiary or Do
nator?\n (Press B for Beneficiary or D for Donator)\n");
                String role;
                do{
                    role = register.nextLine();
                }while(!role.matches("[BbDd]"));
```



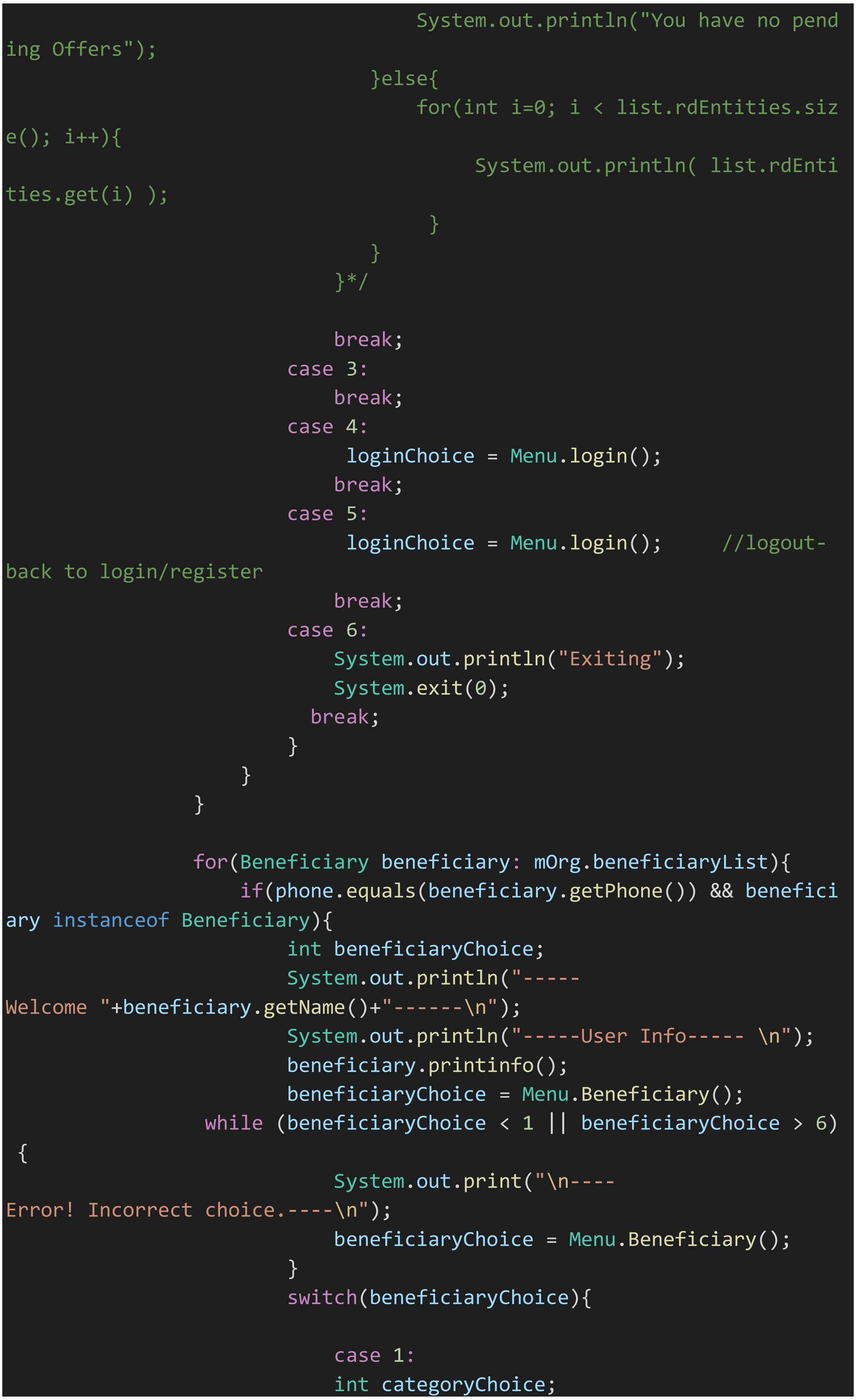
```
    if(role.matches("[Bb]")){
      System.out.println("How many people are in your family(incl
uding you)?");
     String persons = register.nextLine();
     int noP = Integer.parseInt(persons);
     Beneficiary newBeneficiary = new Beneficiary(phone, name, no
P);
     mOrg.insertBeneficiary(newBeneficiary);
     System.out.println("You have been set as a Beneficiary");

        }else{
          Donator newDonator = new Donator(phone, name);
          mOrg.insertDonator(newDonator);
         System.out.println("You have been set as a Donator");
        }

       break;

      case 2: //case 2: login
        Scanner login = new Scanner(System.in);
        System.out.println("Give phone number:");
        phone = login.nextLine();

        do{
        for(Donator donator: mOrg.donatorList){
          if(phone.equals(donator.getPhone()) && donator inst
anceof Donator ){
          int donatorChoice;
          System.out.println("-----
Welcome "+donator.getName()+"------\n");
           System.out.println("-----User Info----- \n");
           donator.printinfo();
           donatorChoice = Menu.Donator();

           while (donatorChoice < 1 || donatorChoice > 6) {
              System.out.print("\n----
Error! Incorrect choice.----\n");
              donatorChoice = Menu.Donator();
            }
            switch(donatorChoice){

            case 1:

              int categoryChoice;
              categoryChoice = Menu.Category();

              while (categoryChoice < 1 || categoryChoice
 > 2) {
```



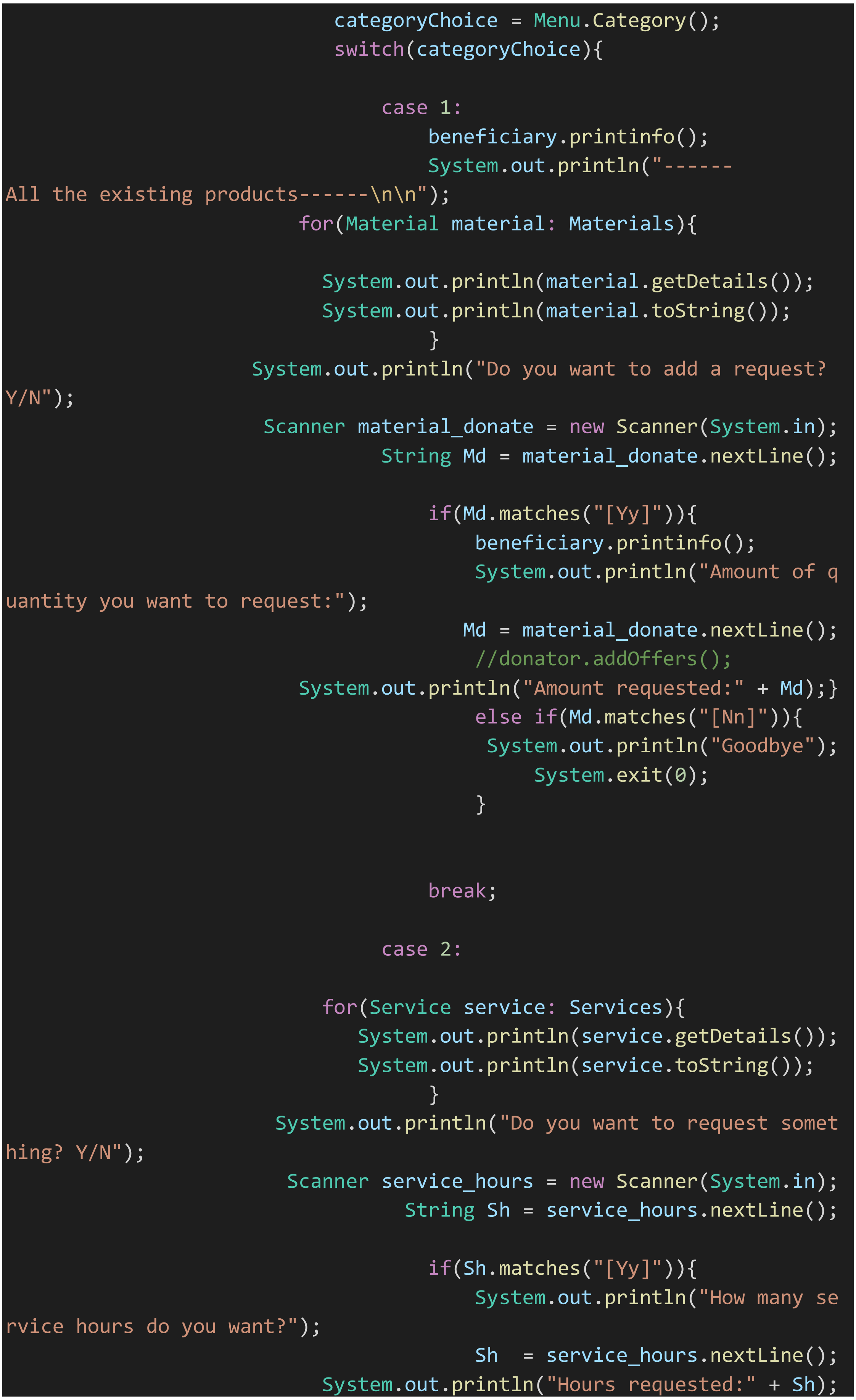
```
                            System.out.print("\n----
Error! Incorrect choice.----\n");
                                categoryChoice = Menu.Category();
                            }
                            switch(categoryChoice){

                                case 1:

                                System.out.println("------
All the existing products------\n\n");
                           for(Material material: Materials){
                             System.out.println(material.getDetails());
                              System.out.println(material.toString());
                                    }
                      System.out.println("Do you want to donate? Y/N");

                      Scanner material_donate = new Scanner(System.in);
                      String Md = material_donate.nextLine();

                      if(Md.matches("[Yy]")){
                           donator.printinfo();
                            System.out.println("Amount of quantity you
want to donate:");
                            Md = material_donate.nextLine();
                            System.out.println("Amount donated:" + Md);

                               donatorChoice = Menu.Donator();
                                } else if(Md.matches("[Nn]")){
                                  System.out.println("Goodbye");
                                   System.exit(0);
                                    }

                                    break;

                                case 2:
                                System.out.println("------
All the existing products------\n\n");
                             for(Service service: Services){
                              System.out.println(service.getDetails());
                                System.out.println(service.toString());
                                    }
                      System.out.println("Do you want to donate? Y/N");
                        Scanner service_hours = new Scanner(System.in);
                                 String Sh = service_hours.nextLine();

                                 if(Sh.matches("[Yy]")){
                                    System.out.println("How many servic
e hours?");
```



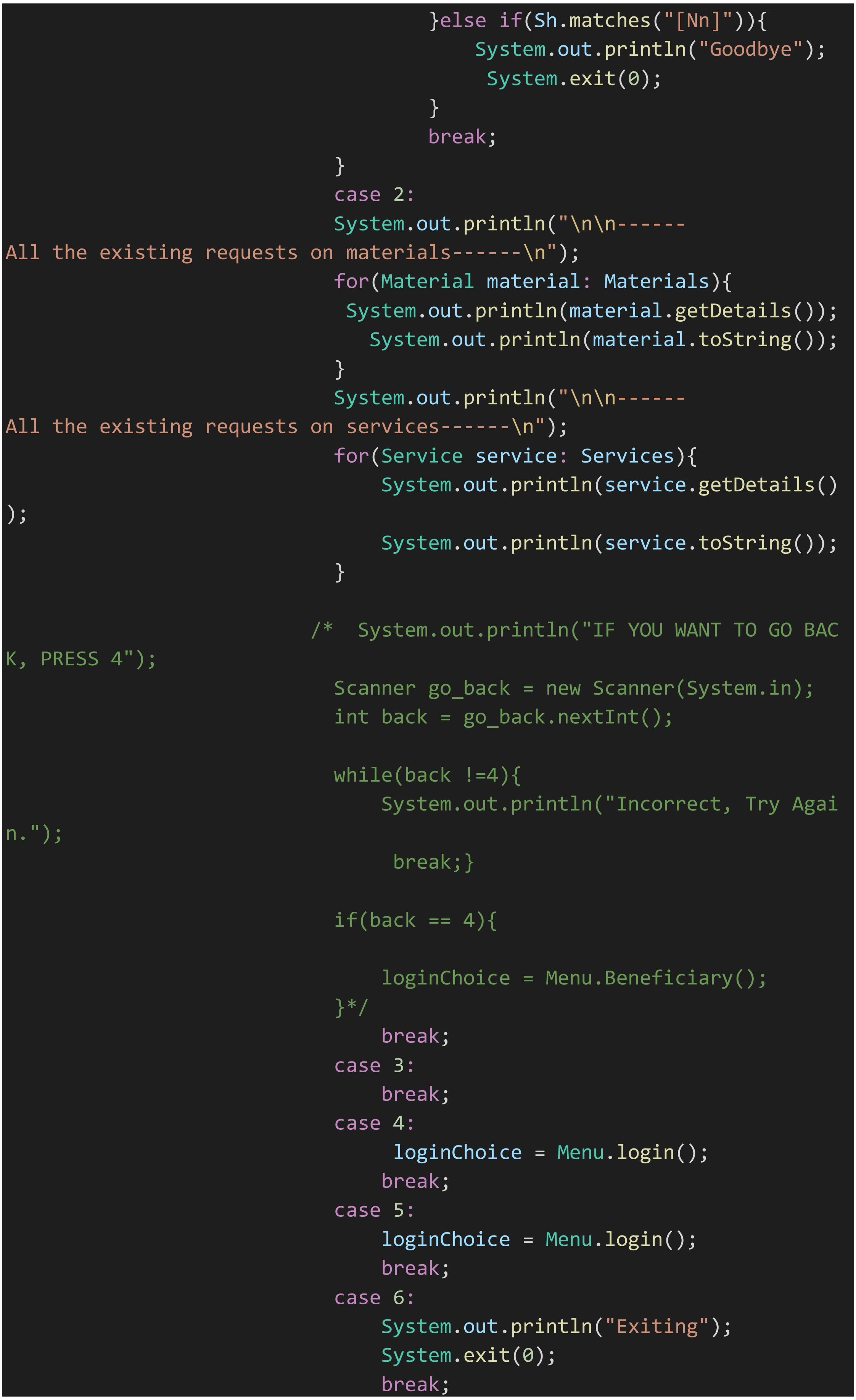
```
                                Sh  = service_hours.nextLine();
                            System.out.println("Hours commited:" + Sh);
                                        loginChoice = Menu.Donator();
                                    }else if(Sh.matches("[Nn]")){
                                        System.out.println("Goodbye");
                                         System.exit(0);
                                    }
                                break;
                            }


                        break;
                        case 2:
                        System.out.println("\n\n------
All the existing offers on materials------\n");
                         for(Material material: Materials){
                             System.out.println(material.getDetails());
                               System.out.println(material.toString());
                                    }
                                    System.out.println("\n\n------
All the existing offers on services------\n");
                           for(Service service: Services){
                              System.out.println(service.getDetails());
                              System.out.println(service.toString());
                                    }
                                    System.exit(0);
                                    /*System.out.println("IF YOU WANT T
O GO BACK, PRESS 4");

                                    Scanner go_back = new Scanner(Syste
m.in);

                                    int back = go_back.nextInt();

                                    while(back !=4){
                                        System.out.println("Incorrect,
Try Again.");

                                         break;}

                                    if(back == 4){

                                        loginChoice = Menu.Donator();
                                    }*/





                               /*for(Offers list: donator.offersList){

                               if(list.rdEntities.isEmpty()){
```



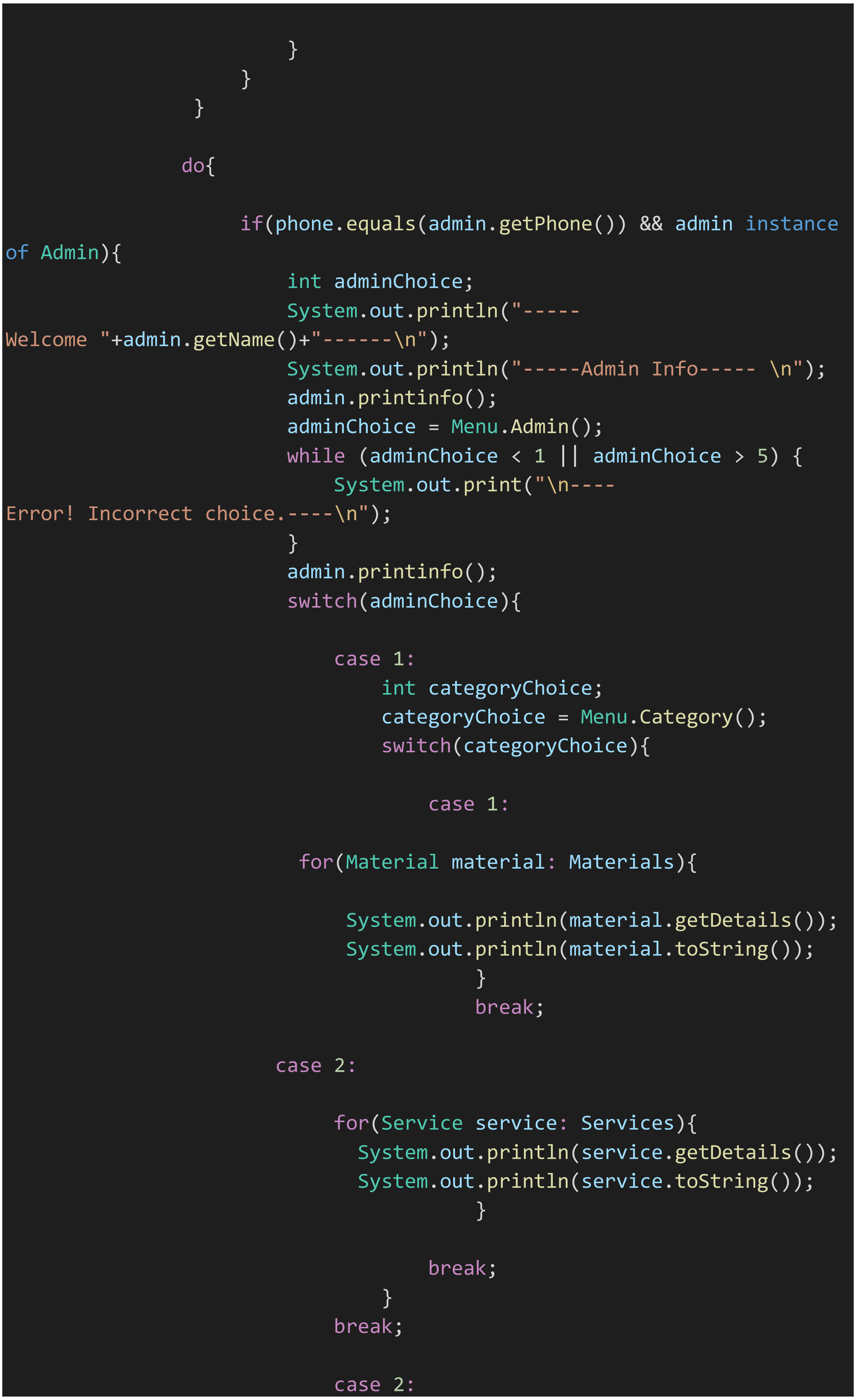
```
                                   System.out.println("You have no pend
ing Offers");
                               }else{
                                   for(int i=0; i < list.rdEntities.siz
e(); i++){
                                        System.out.println( list.rdEnti
ties.get(i) );
                                    }
                               }
                            }*/

                            break;
                        case 3:
                            break;
                        case 4:
                             loginChoice = Menu.login();
                            break;
                        case 5:
                             loginChoice = Menu.login();     //logout-
back to login/register
                            break;
                        case 6:
                            System.out.println("Exiting");
                            System.exit(0);
                          break;
                        }
                    }
                }

                for(Beneficiary beneficiary: mOrg.beneficiaryList){
                    if(phone.equals(beneficiary.getPhone()) && benefici
ary instanceof Beneficiary){
                        int beneficiaryChoice;
                        System.out.println("-----
Welcome "+beneficiary.getName()+"------\n");
                        System.out.println("-----User Info----- \n");
                        beneficiary.printinfo();
                        beneficiaryChoice = Menu.Beneficiary();
                 while (beneficiaryChoice < 1 || beneficiaryChoice > 6)
 {
                            System.out.print("\n----
Error! Incorrect choice.----\n");
                            beneficiaryChoice = Menu.Beneficiary();
                        }
                        switch(beneficiaryChoice){

                            case 1:
                            int categoryChoice;
```



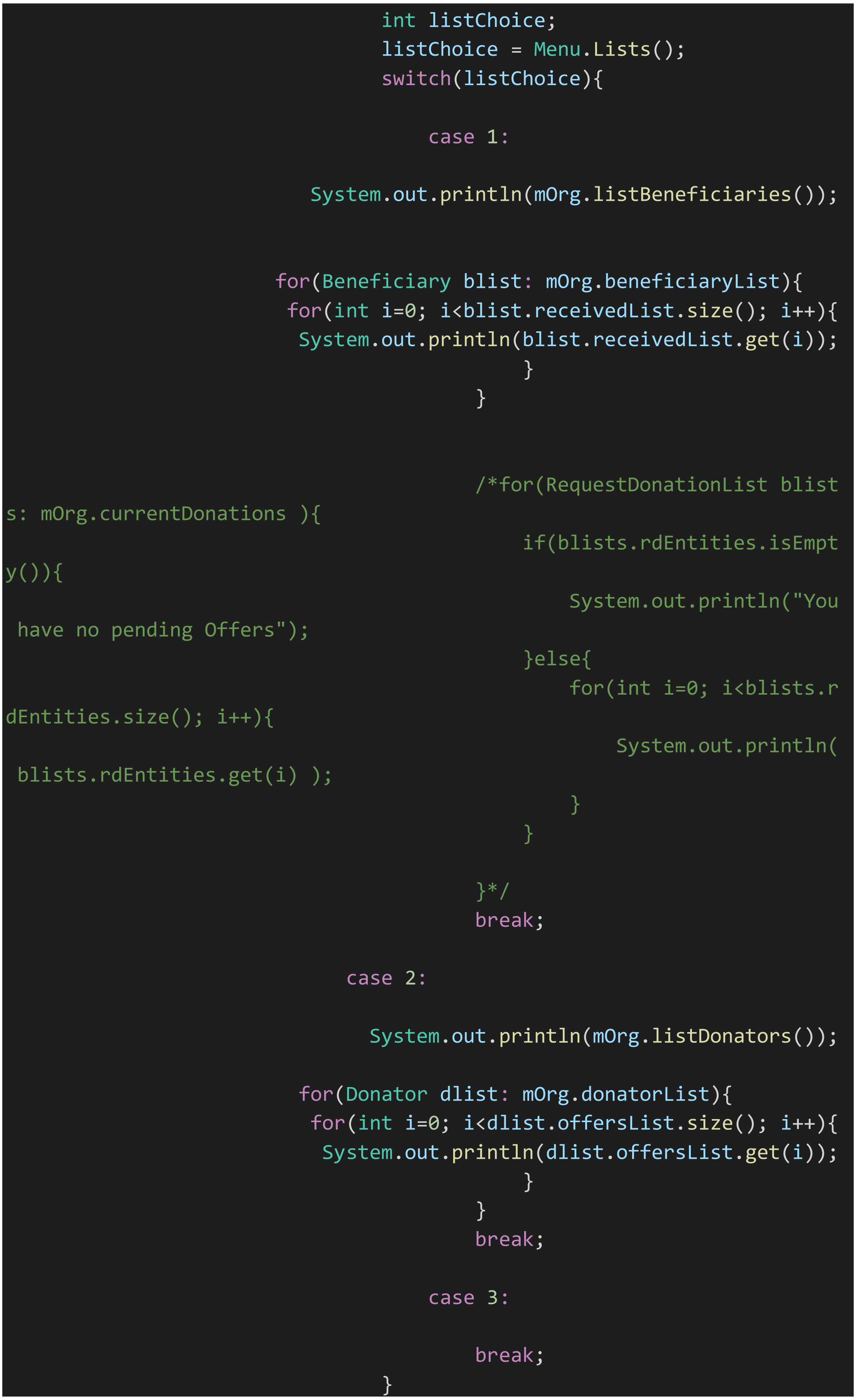
```
                            categoryChoice = Menu.Category();
                            switch(categoryChoice){

                                case 1:
                                    beneficiary.printinfo();
                                    System.out.println("------
All the existing products------\n\n");
                         for(Material material: Materials){

                           System.out.println(material.getDetails());
                           System.out.println(material.toString());
                                    }
                     System.out.println("Do you want to add a request?
Y/N");
                      Scanner material_donate = new Scanner(System.in);
                                String Md = material_donate.nextLine();

                                    if(Md.matches("[Yy]")){
                                        beneficiary.printinfo();
                                        System.out.println("Amount of q
uantity you want to request:");
                                       Md = material_donate.nextLine();
                                        //donator.addOffers();
                         System.out.println("Amount requested:" + Md);}
                                        else if(Md.matches("[Nn]")){
                                         System.out.println("Goodbye");
                                             System.exit(0);
                                        }


                                    break;

                                case 2:

                           for(Service service: Services){
                              System.out.println(service.getDetails());
                              System.out.println(service.toString());
                                    }
                       System.out.println("Do you want to request somet
hing? Y/N");
                        Scanner service_hours = new Scanner(System.in);
                                  String Sh = service_hours.nextLine();

                                    if(Sh.matches("[Yy]")){
                                        System.out.println("How many se
rvice hours do you want?");
                                        Sh   = service_hours.nextLine();
                           System.out.println("Hours requested:" + Sh);
```



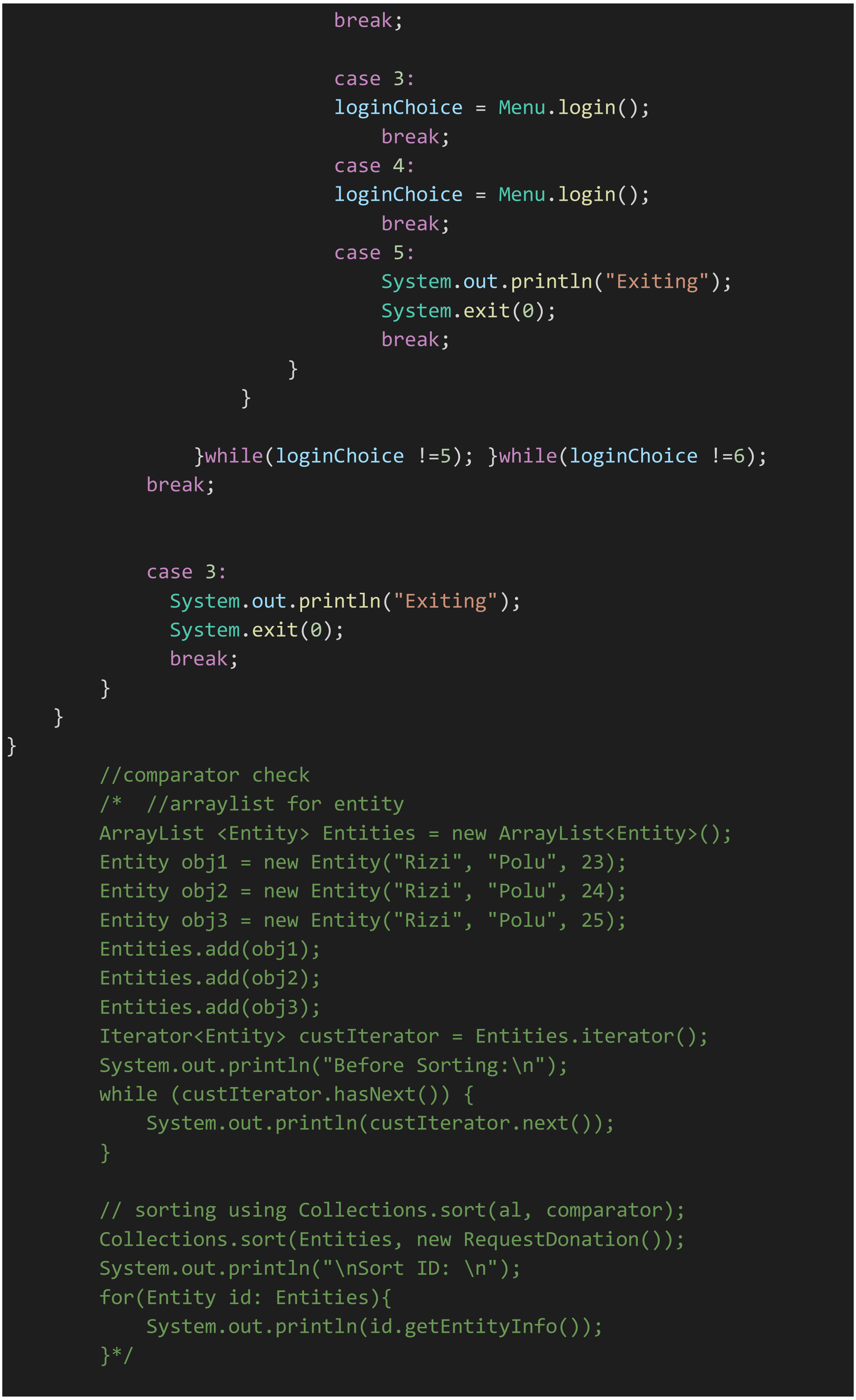
```
                                    }else if(Sh.matches("[Nn]")){
                                        System.out.println("Goodbye");
                                         System.exit(0);
                                    }
                                    break;
                            }
                            case 2:
                            System.out.println("\n\n------
All the existing requests on materials------\n");
                            for(Material material: Materials){
                             System.out.println(material.getDetails());
                               System.out.println(material.toString());
                            }
                            System.out.println("\n\n------
All the existing requests on services------\n");
                            for(Service service: Services){
                                System.out.println(service.getDetails()
);

                                System.out.println(service.toString());
                            }

                          /*  System.out.println("IF YOU WANT TO GO BAC
K, PRESS 4");
                            Scanner go_back = new Scanner(System.in);
                            int back = go_back.nextInt();

                            while(back !=4){
                                System.out.println("Incorrect, Try Agai
n.");

                                 break;}

                            if(back == 4){

                                loginChoice = Menu.Beneficiary();
                            }*/
                                break;
                            case 3:
                                break;
                            case 4:
                                 loginChoice = Menu.login();
                                break;
                            case 5:
                                loginChoice = Menu.login();
                                break;
                            case 6:
                                System.out.println("Exiting");
                                System.exit(0);
                                break;
```



```
                        }
                    }
                }

               do{

                    if(phone.equals(admin.getPhone()) && admin instance
of Admin){
                        int adminChoice;
                        System.out.println("-----
Welcome "+admin.getName()+"------\n");
                        System.out.println("-----Admin Info----- \n");
                        admin.printinfo();
                        adminChoice = Menu.Admin();
                        while (adminChoice < 1 || adminChoice > 5) {
                            System.out.print("\n----
Error! Incorrect choice.----\n");
                        }
                        admin.printinfo();
                        switch(adminChoice){

                            case 1:
                                int categoryChoice;
                                categoryChoice = Menu.Category();
                                switch(categoryChoice){

                                    case 1:

                         for(Material material: Materials){

                             System.out.println(material.getDetails());
                             System.out.println(material.toString());
                                        }
                                        break;

                       case 2:

                            for(Service service: Services){
                              System.out.println(service.getDetails());
                              System.out.println(service.toString());
                                        }

                                    break;
                                }
                            break;

                            case 2:
```



```
                                int listChoice;
                                listChoice = Menu.Lists();
                                switch(listChoice){

                                    case 1:

                          System.out.println(mOrg.listBeneficiaries());


                       for(Beneficiary blist: mOrg.beneficiaryList){
                        for(int i=0; i<blist.receivedList.size(); i++){
                         System.out.println(blist.receivedList.get(i));
                                            }
                                        }



                                        /*for(RequestDonationList blist
s: mOrg.currentDonations ){
                                            if(blists.rdEntities.isEmpt
y()){
                                                System.out.println("You
 have no pending Offers");
                                            }else{
                                                for(int i=0; i<blists.r
dEntities.size(); i++){
                                                    System.out.println(
 blists.rdEntities.get(i) );
                                                }
                                            }

                                        }*/
                                        break;

                             case 2:

                               System.out.println(mOrg.listDonators());

                         for(Donator dlist: mOrg.donatorList){
                          for(int i=0; i<dlist.offersList.size(); i++){
                           System.out.println(dlist.offersList.get(i));
                                            }
                                        }
                                        break;

                                    case 3:

                                        break;
                                }
```



```
                            break;

                            case 3:
                            loginChoice = Menu.login();
                                break;
                            case 4:
                            loginChoice = Menu.login();
                                break;
                            case 5:
                                System.out.println("Exiting");
                                System.exit(0);
                                break;
                        }
                    }

                }while(loginChoice !=5); }while(loginChoice !=6);
            break;


            case 3:
              System.out.println("Exiting");
              System.exit(0);
              break;
        }
    }
}
        //comparator check
        /*  //arraylist for entity
        ArrayList <Entity> Entities = new ArrayList<Entity>();
        Entity obj1 = new Entity("Rizi", "Polu", 23);
        Entity obj2 = new Entity("Rizi", "Polu", 24);
        Entity obj3 = new Entity("Rizi", "Polu", 25);
        Entities.add(obj1);
        Entities.add(obj2);
        Entities.add(obj3);
        Iterator<Entity> custIterator = Entities.iterator();
        System.out.println("Before Sorting:\n");
        while (custIterator.hasNext()) {
            System.out.println(custIterator.next());
        }

        // sorting using Collections.sort(al, comparator);
        Collections.sort(Entities, new RequestDonation());
        System.out.println("\nSort ID: \n");
        for(Entity id: Entities){
            System.out.println(id.getEntityInfo());
        }*/
```