

Project Java

Ομάδα :

Μουστάκι Αλέξανδρος (AM:1072532) scrap2bills@gmail.com

Αθανάσιος Κανδρεβιώτης (AM:1067401) thanoskand@gmail.com

Περικλής Λαμπαδητής (AM:1072502) perilamp01@gmail.com

<https://github.com/ARKOUDAKII/ProjectObjectOriented2021>

Εδώ έχουμε ανεβάσει τον κώδικα (Τον έχουμε γράψει στο BlueJ).

Τεκμηριώσει :

Γενικά οι πιο σημαντικές classes στο πρόγραμμα είναι το Menu με το οποίο αλληλεπιδρά ο χρήστης και η class Organization από την οποία μπορεί ο προγραμματιστής να πάρει οποιαδήποτε πληροφορία χρειάζεται (Donators, Beneficiaries, Offers, Requests,). Το Menu φτιαχτεί έτσι ώστε οι διάφορες μεθόδους να πηγαινούν από την μια στην άλλη και πίσω, ώστε πρακτικά να έχουμε πολλά διαφορά μενού, με τον χρήστη να μπορεί να πηγαίνει από το ένα στο άλλο. Η class Organization είναι η πιο «Ανωτερη» class του προγράμματος. Περιέχει λίστες για τους Donators, Beneficiaries που είναι οι χρήστες που μπορούν να κάνουν δωρεές ή να παίρνουν δωρεές. Μετά είναι και η λίστα Stock που δίνει όλα τα αντικείμενα και η ποσότητα που έχει ο οργανισμός. Η λίστα EnList δείχνει στον Donator ποια αντικείμενα/υπηρεσίες μπορεί να προσφέρει. Οι classes Donator και Beneficiary έχουν την δυνατότητα να κάνουν Offers/Request στον/από τον οργανισμό.

Entity είναι η class που χρησιμοποιούμε για περιγράψουμε τα αντικείμενα/υπηρεσίες που θα χρησιμοποιήσουμε. Έχει δύο subclasses. Την Material και την Service. RequestDonation είναι μια class που αποτελείται από ένα entity και μια ποσότητα. Αυτή η class υπάρχει για να μπορούμε να εκφράσουμε request/offers που είναι και οι δύο subclasses του RequestDonation. Υπάρχει και η class Admin που ουσιαστικά ελέγχει τον οργανισμό.

Κώδικας:

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

```
import java.util.Scanner;
```

```
public class Organization
```

```
{
```

```
    //Variable
```

```
    static int foundAt;
```

```
    private String name;
```

```
    private Admin ad = new Admin("",0);
```

```

private ArrayList<RequestDonation> Stock = new ArrayList<RequestDonation>();
private ArrayList<Donator> DonList = new ArrayList<Donator>();
private ArrayList<Beneficiary> BenList = new ArrayList<Beneficiary>();
private ArrayList<Entity> EnList = new ArrayList<Entity>();
Scanner sc = new Scanner(System.in);

//Con
public Organization(String Name ,Admin ad) {
    this.name = Name;
    this.ad = ad;
    this.Stock = new ArrayList<RequestDonation>();
    this.DonList = new ArrayList<Donator>();
    this.BenList = new ArrayList<Beneficiary>();
    this.EnList = new ArrayList<Entity>();
}

//Methods
//Methods to access the 3 lists we have and methods so we have get an orderly list of each of
them

public Admin getAdmin() {return this.ad;}
public void addDon(Donator D) {this.DonList.add(D);}
public void addBen(Beneficiary B) {this.BenList.add(B);}


public ArrayList<Entity> getEnList() {return this.EnList;}
public void addEn(Entity E) {
    EnList.add(E);
}

public void showcaseEnList() {
    for(int i=0;i<this.EnList.size();i++) {
        System.out.println((i+1)+" "+EnList.get(i).getName()+" "+EnList.get(i).getId()+"
"+EnList.get(i).getDetails());
    }
}

public ArrayList<RequestDonation> getStock() {return this.Stock;}

```

```

public void ListStock() {
    for(int i=0;i<Stock.size();i++){
        System.out.println((i+1)+" "+Stock.get(i).getEName()+" "+Stock.get(i).getEld()+" "+Stock.get(i).getEn().getDetails()+" "+Stock.get(i).getQuantity());
    }
}

public ArrayList<Donator> getDons() {return this.DonList;}

public void ListDons() {
    for(int i=0;i<DonList.size();i++){System.out.println((i+1)+" "+DonList.get(i).getName()+" "+DonList.get(i).getId());}
}

public ArrayList<Beneficiary> getBens() {return this.BenList;}

public void ListBens() {
    for(int i=0;i<BenList.size();i++){System.out.println((i+1)+" "+BenList.get(i).getName()+" "+BenList.get(i).getId());}
}

//Method for adding a Donator's Offer
public void AddOffer(Offer Of) {
    boolean found=false;
    for (int i=0;i<this.Stock.size();i++){
        if (Stock.get(i).getEld()==Of.getEld()){
            found =true;foundAt=i;
        }
    }
    if (found==true) {
        Stock.get(foundAt).setQuantity(Stock.get(foundAt).getQuantity()+Of.getQuantity());
    }
    else {Stock.add(Of);}
}

//Method to commit a beneficiary's Request

```

```

public void CommitRequest(Request R) {
    for (int i=0;i<Stock.size();i++) {
        if (Stock.get(i).getEld()==R.getEld()) {
            if (Stock.get(i).getQuantity() > R.getQuantity()) {
                Stock.get(i).setQuantity(Stock.get(i).getQuantity()-R.getQuantity());
            }
            else if (Stock.get(i).getQuantity() == R.getQuantity()) {
                Stock.remove(i);
            }
            else {
                System.out.println("Sorry sir , but the requested amount is not available to the
organization");
            }
        }
    }
}

//Method to change an entity's quantity in stock
public void modifyQuantity(int Id,double q) {
    for(int i=0;i<Stock.size();i++) {
        if (Stock.get(i).getEld()==Id) {
            Stock.get(i).setQuantity(q);
        }
    }
}

//Method to remove Beneficiary
public void removeBen(int Id) {
    for(int i=0;i<BenList.size();i++) {
        if (BenList.get(i).getId()==Id) {
            BenList.remove(i);
        }
    }
}

```

```

    }

    //Method to remove Donator
    public void removeDon(int Id) {
        for (int i=0;i<DonList.size();i++){
            if (DonList.get(i).getId()==Id){
                DonList.remove(i);
            }
        }
    }
}

public class User
{
    //Variable
    private String Name;
    private int Id;

    //Con
    public User(String Name,int Id) {this.Name = new String(Name);this.Id = Id;}

    //Method
    public String getName() {return Name;}
    public int getId() {return Id;}
}

import java.util.ArrayList;

public class Donator extends User
{
    //Variable
    static boolean found;
    private ArrayList<Offer> Offers = new ArrayList<Offer>();

    //Con
    public Donator(String Name,int Id) {

```

```

    super(Name,Id);

    this.Ooffers = new ArrayList<Offer>();
}

//Methods

public ArrayList<Offer> getOffers() {

    return this.Ooffers;

}

public void showcaseOffers(){

    for(int i=0;i<Ooffers.size();i++){

        System.out.println(Ooffers.get(i).getENAME());

    }

}

public void addtoOffer(Offer Of) {

    found = false;

    //System.out.println("Hello");

    for(int i=0;i<Ooffers.size();i++){

        if (Ooffers.get(i).getEId()==Of.getEId()){

this.Ooffers.get(i).setQuantity(Ooffers.get(i).getQuantity()+Of.getQuantity());found=true;//System.out.
println("Hello");

        }

    }

    if (found == false){

        this.Ooffers.add(Of);

    }

}

public void clearOffers() {

    this.Ooffers.clear();

}

public void deleteOffer(Offer f) {

    this.Ooffers.remove(f);

}

```

```
}
```

```
public class Admin extends User
```

```
{
```

```
    //Variables
```

```
    //Con
```

```
    public Admin(String Name,int Id) {super(Name,Id);}
```

```
    //Methods
```

```
}
```

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

```
public class Beneficiary extends User
```

```
{
```

```
    //Variables
```

```
    static boolean found;
```

```
    private int noPeople;
```

```
    private ArrayList<Request> RequestList= new ArrayList<Request>();
```

```
    private ArrayList<Request> ReceivedList= new ArrayList<Request>();
```

```
    //Con
```

```
    public Beneficiary(String Name,int Id,int noPeople) {
```

```
        super(Name,Id);
```

```
        this.noPeople = noPeople;
```

```
        this.RequestList= new ArrayList<Request>();
```

```
        this.ReceivedList= new ArrayList<Request>();
```

```
    }
```

```
    //Methods
```

```
    public int getnoPeople() {return this.noPeople;}
```

```
    public ArrayList<Request> getRequestList() {return this.RequestList;}
```

```
    public ArrayList<Request> getReceivedList() {return this.ReceivedList;}
```

```

public void showcaseReceivedList() {
    for (int i=0;i<ReceivedList.size();i++){
        System.out.println(ReceivedList.get(i).getENAME());
    }
}

public void showcaseRequestList() {
    for (int i=0;i<RequestList.size();i++) {
        System.out.println(RequestList.get(i).getENAME()+" "+RequestList.get(i).getEId()+"
"+RequestList.get(i).getQuantity());
    }
}

public void clearRequestList() {
    this.RequestList.clear();
}

public void clearReceivedList() {
    this.ReceivedList.clear();
}

public void addRequest(Request R) {
    found = false;
    for(int i=0;i<RequestList.size();i++) {
        if (RequestList.get(i).getEId()==R.getEId()) {
            this.RequestList.get(i).setQuantity(RequestList.get(i).getQuantity()+R.getQuantity());found=true;
        }
    }
    if (found == false){
        this.RequestList.add(R); //System.out.println("Hello");
    }
    //this.RequestList.add(R);
}
}

```



```
abstract class Entity
```

```
{
```

```
    //Variables
```

```
    private String Name;
```

```
    private int Id;
```

```
    private String Description;
```

```
    //Con
```

```
    public Entity(String Name,int Id,String Desc) {
```

```
        this.Name = new String(Name);
```

```
        this.Id = Id;
```

```
        this.Description = new String(Desc);
```

```
    }
```

```
    //Methods
```

```
    public String getName() {return this.Name;} 
```

```
    public String getDesc() {return this.Description;} 
```

```
    public int getId() {return this.Id;} 
```

```
    public String getInfo() {return this.Name+" "+this.Description+" "+this.Id;} 
```

```
    abstract String getDetails();
```

```
    abstract double getlevel1();
```

```
    abstract double getlevel2();
```

```
    abstract double getlevel3();
```

```
    public String toString() {return getInfo()+" "+getDetails();}
```

```
}
```

```
public class Service extends Entity
```

```
{
```

```
    //Variable
```

```
    //Con
```

```
    public Service(String Name,int Id,String Description) {super(Name,Id,Description);} 
```

```
    //Methods
```

```
    public double getlevel1() {System.out.println("This is service! It doesnt contain any levels!"
);return 0;}
```

```
    public double getlevel2() {System.out.println("This is service! It doesnt contain any levels!"
);return 0;}
```

```
    public double getlevel3() {System.out.println("This is service! It doesnt contain any levels!"
);return 0;}
```

```
    public String getDetails() {return "Service";}
}
```

```
public class Material extends Entity
```

```
{
```

```
    //Variables
```

```
    double level1;
```

```
    double level2;
```

```
    double level3;
```

```
    //Con
```

```
    public Material(String Name,int Id,String Description,double level1,double level2,double level3) {
```

```
        super(Name,Id,Description);
```

```
        this.level1 = level1;
```

```
        this.level2 = level2;
```

```
        this.level3 = level3;
```

```
    }
```

```
    //Methods
```

```
    public double getlevel1() {return this.level1;}
```

```
    public double getlevel2() {return this.level2;}
```

```
    public double getlevel3() {return this.level3;}
```

```
    public String getDetails() {return "Material";}
```

```
}
```

```
public class RequestDonation
```

```
{
```

```
    //Variables
```

```

private Entity e;
private double Quantity;

//Con
public RequestDonation(Entity e,double Quantity) {this.e = e;this.Quantity = Quantity;}

//Methods
public String getENAME() {return this.e.getName();}
public int getEId() {return this.e.getId();}
public Entity getEn() {return this.e;}
public double getQuantity() {return this.Quantity;}
public void setQuantity(double q) {this.Quantity = q;}
}

```

```

public class Offer extends RequestDonation
{
    //Variables
    //Con
    public Offer(Entity e,double q) {
        super(e,q);
    }
    //Method
}

```

```

public class Request extends RequestDonation
{
    //Variables
    //Con
    public Request(Entity e,double q) {
        super(e,q);
    }
    //Method
}

```

```

import java.util.Scanner;

import java.util.ArrayList;


public class Menu
{
    //Variables

    static int DonLoc;

    static int BenLoc;

    static int phoneNum;

    static int answer;

    private Scanner sc = new Scanner(System.in);

    private Admin ad = new Admin("",0);

    private Organization Org = new Organization("",ad);

    private ArrayList<RequestDonation> SudoArray = new ArrayList<RequestDonation>();


    //Con

    public Menu(Organization Org) {

        this.Org=Org;

        SudoArray = Org.getStock();

    }

    //Methods

    public int Start() {

        System.out.println("Please enter your phone number : ");

        phoneNum = sc.nextInt();

        if (phoneNum == Org.getAdmin().getId()) {answer=1;}

        for(int i=0;i<Org.getDons().size();i++) {if (Org.getDons().get(i).getId()==phoneNum)
{answer=2;System.out.println("Donator");};}

        for(int i=0;i<Org.getBens().size();i++) {if (Org.getBens().get(i).getId()==phoneNum)
{answer=3;System.out.println("Beneficiary");};}

        if (answer == 1) {AdminMenu();}

        if (answer == 2) {DonMenu();}

        if (answer == 3) {BenMenu();}
    }
}

```

```

else {System.out.println("Please enter a valid option! ");Start();}

return 0;

}

```

```

public int AdminMenu() {

    System.out.println("1.View\n2.Monitor\n3.Back\n4.Logout\n5.Exit");

    answer = sc.nextInt();

    if (answer == 1) {AdminMenu1();}

    if (answer == 2) {AdminMenu2();}

    if (answer == 3) {Start();}

    if (answer == 4) {Start();}

    if (answer == 5) {System.exit(0);}

    else {System.out.println("Please enter a valid option! ");AdminMenu();}

    return 0;

}

```

```

public int AdminMenu1() {

    Org.ListStock();

    System.out.println("Would you like more details on a certain entity ? (1.Yes/2.No) : ");

    answer = sc.nextInt();

    if (answer == 1) {

        System.out.println("Please enter the entity's ID : ");

        int GivenID = sc.nextInt();

        for (int i=0;i<Org.getStock().size();i++){

            if (Org.getStock().get(i).getEId()==GivenID) {

                System.out.println(Org.getStock().get(i).getEn().toString());

                AdminMenu();

            }

        }

    }

}

```

```

else if (answer == 2) {AdminMenu();}

else {System.out.println("Please enter a valid answer!");AdminMenu1();}

return 0;
}

```

```

public int AdminMenu2() {

    System.out.println("1.List Beneficiaries\n2.List Donators\n3.Reset Beneficiaries
ReceivedLists\n4.Back");

    answer = sc.nextInt();

    if (answer==1) {AdminMenu2a();}

    if (answer==2) {AdminMenu2b();}

    if (answer==3) {AdminMenu2c();}

    if (answer==4) {AdminMenu();}

    else {System.out.println("Please enter a valid option! ");AdminMenu2();}

    return 0;
}

```

```

public int AdminMenu2a() {

    Org.ListBens();

    System.out.println("1.See their receivedList\n2.Clear their receivedList\n3.Remove them from
the BeneficiariesList\n4.Back");

    int GivenChoice = sc.nextInt();

    if (GivenChoice == 1) {

        System.out.println("Please enter the Id of the Beneficiary that you would like to manage :");

        int BenID = sc.nextInt();

        for(int i=0;i<Org.getBens().size();i++){

            if (Org.getBens().get(i).getId()==BenID) {

                Org.getBens().get(i).showcaseReceivedList();AdminMenu();

            }

        }

    }

    else if (GivenChoice == 2) {

```

```

        System.out.println("Please enter the name of the Beneficiary that you would like to manage :
");
        String BenName = sc.nextLine();
        for(int i=0;i<Org.getBens().size();i++){
            if (Org.getBens().get(i).getName()==BenName) {
                Org.getBens().get(i).getReceivedList().clear();AdminMenu();
            }
        }
    }
    else if (GivenChoice == 3) {
        System.out.println("Please enter the name of the Beneficiary that you would like to manage :
");
        String BenName = sc.nextLine();
        for(int i=0;i<Org.getBens().size();i++){
            if (Org.getBens().get(i).getName()==BenName) {
                Org.getBens().remove(i);AdminMenu();
            }
        }
    }
    else if (GivenChoice == 4) {AdminMenu2();}
    else {
        System.out.println("Please enter a valid option!");
        AdminMenu2a();
    }
    return 0;
}

```

```

public int AdminMenu2b() {
    Org.ListDons();
    System.out.println("Please enter the name of the Donator that you would like to manage : ");
    String DonName = sc.nextLine();
}

```

```
System.out.println("1.Show Donator's OfferList\n2.Remove Donator from  
DonatorList\n3.Back");
```

```
int GivenChoice = sc.nextInt();
```

```
if (GivenChoice == 1) {
```

```
    for(int i=0;i<Org.getDons().size();i++) {
```

```
        if (Org.getDons().get(i).getName()==DonName) {
```

```
            Org.getDons().get(i).showcaseOffers();AdminMenu();
```

```
        }
```

```
    }
```

```
}
```

```
if (GivenChoice == 2) {
```

```
    for(int i=0;i<Org.getDons().size();i++) {
```

```
        if (Org.getDons().get(i).getName()==DonName) {
```

```
            Org.getDons().remove(i);AdminMenu();
```

```
        }
```

```
    }
```

```
}
```

```
if (GivenChoice == 3) {AdminMenu2();}
```

```
else {System.out.println("Please enter a valid option!");AdminMenu2b();}
```

```
return 0;
```

```
}
```

```
public int AdminMenu2c() {
```

```
    for (int i=0;i<Org.getBens().size();i++) {
```

```
        Org.getBens().get(i).clearReceivedList();
```

```
    }
```

```
return 0;
```

```
}
```

```
public int DonMenu() {
```



```

System.out.println("1.Add Offer\n2.Show Offers\n3.Commit\n4.Back\n5.Exit");
answer = sc.nextInt();
for (int i=0;i<Org.getDons().size();i++) {
    if (Org.getDons().get(i).getId()==phoneNum) {
        DonLoc = i;
        //System.out.println("Hi");
    }
}
if (answer == 1) {DonMenu1();}
else if (answer == 2) {DonMenu2();}
else if (answer == 3) {DonMenu3();}
else if (answer == 4) {Start();}
else if (answer == 5) {System.exit(0);}
else {System.out.println("Please enter a valid option");DonMenu();}
return 0;
}

```

```

public int DonMenu1() {
    Org.showcaseEnList();
    System.out.println("Would you like to add an offer? (1.Yes/2.No)");
    answer = sc.nextInt();
    if (answer==1){
        System.out.println("What would you like to add ? (Enter Id And Quantity) ");
        answer = sc.nextInt();
        double newQ = sc.nextDouble();
        for (int i=0;i<Org.getEnList().size();i++) {
            if (Org.getEnList().get(i).getId()==answer) {
                System.out.println("Is the entity a Material or a Service ? (1.Material/2.Service)");
                answer = sc.nextInt();
                if (answer ==1){
                    String Name = Org.getEnList().get(i).getName();

```

```

        int Id = Org.getEnList().get(i).getId();

        String Desc = Org.getEnList().get(i).getDesc();

        double lvl1=Org.getEnList().get(i).getlevel1();
        double lvl2=Org.getEnList().get(i).getlevel2();
        double lvl3=Org.getEnList().get(i).getlevel3();

        Material M = new Material(Name,Id,Desc,lvl1,lvl2,lvl3);

        Offer f = new Offer(M,newQ);

        Org.getDons().get(DonLoc).addtoOffer(f);

        DonMenu();
    }

    else if (answer==2) {

        String Name = Org.getEnList().get(i).getName();

        int Id = Org.getEnList().get(i).getId();

        String Desc = Org.getEnList().get(i).getDesc();

        Service Ser = new Service(Name,Id,Desc);

        Offer f = new Offer(Ser,newQ);

        Org.getDons().get(DonLoc).addtoOffer(f);

        //System.out.println(Org.getDons().get(DonLoc).getOffers().get(0));

        DonMenu();

    }

    else {

        System.out.println("Please enter a valid option! ");DonMenu1();

    }

}

}

}

if (answer==2) {DonMenu();}

else {System.out.println("Please enter a valid option");DonMenu1();}

return 0;

}

```

```

public int DonMenu2() {
    Org.getDons().get(DonLoc).showcaseOffers();

    System.out.println("1.Delete offer/Change Quantity of offer\n2.Clear OfferList\n3.Back");
    answer = sc.nextInt();

    if (answer ==1) {
        System.out.println("Please enter the entity's ID :");
        int entityID = sc.nextInt();

        System.out.println("1.Delete this offer\n2.Change quantity ");
        answer = sc.nextInt();

        if (answer == 1) {
            for (int i=0;i<Org.getDons().get(DonLoc).getOffers().size();i++) {
                if (Org.getDons().get(DonLoc).getOffers().get(i).getEId()==entityID) {

Org.getDons().get(DonLoc).deleteOffer(Org.getDons().get(DonLoc).getOffers().get(i));DonMenu();

                }
            }
        }

        if (answer == 2) {
            System.out.println("What is the entity's new quantity? ");
            double newQ = sc.nextDouble();

            for (int i=0;i<Org.getDons().get(DonLoc).getOffers().size();i++) {
                if (Org.getDons().get(DonLoc).getOffers().get(i).getEId()==entityID) {
                    Org.getDons().get(DonLoc).getOffers().get(i).setQuantity(newQ);DonMenu();
                }
            }
        }

        if (answer ==2) {Org.getDons().get(DonLoc).clearOffers();DonMenu();}

        if (answer ==3) {DonMenu();}

        return 0;
    }
}

```

```

public int DonMenu3() {
    for (int i=0;i<Org.getDons().get(DonLoc).getOffers().size();i++) {
        Org.AddOffer(Org.getDons().get(DonLoc).getOffers().get(i));
    }
    Org.getDons().get(DonLoc).clearOffers();
    DonMenu();
    return 0;
}

```

```

public int BenMenu() {
    System.out.println("1.Add Request\n2.Show Requests\n3.Commit\n4.Back\n5.Exit");
    answer = sc.nextInt();
    for (int i=0;i<Org.getBens().size();i++) {
        if (Org.getBens().get(i).getId()==phoneNum) {
            BenLoc = i;
        }
    }
    if (answer == 1) {BenMenu1();}
    if (answer == 2) {BenMenu2();}
    if (answer == 3) {BenMenu3();}
    if (answer == 4) {Start();}
    if (answer == 5) {System.exit(0);}
    else {System.out.println("Please enter a valid option");DonMenu();}
    return 0;
}

```

```

public int BenMenu1() {
    Org.ListStock();
    System.out.println("Would you like to add a request ? (1.Yes/2.No)");
}

```

```

answer = sc.nextInt();
if (answer==1){
    System.out.println("What would you like to add ? (1.Material/2.Service) ");
    answer = sc.nextInt();
    if (answer == 1) {
        System.out.println("What is the ID of the desired material? ");
        int entityID = sc.nextInt();
        for (int i=0;i<Org.getStock().size();i++){
            if (Org.getStock().get(i).getEId()==entityID) {
                int noPeople = Org.getBens().get(BenLoc).getnoPeople();
                if (noPeople == 1) {
                    double level1 = Org.getStock().get(i).getEn().getlevel1();
                    if(level1>Org.getStock().get(i).getQuantity()){
                        System.out.println("Sorry but this amount isnt available!");
                        BenMenu1();
                    }
                }
                else {
                    String Name = Org.getStock().get(i).getEName();
                    String Desc=Org.getStock().get(i).getEn().getDesc();
                    Service s = new Service(Name,entityID,Desc);
                    Request R = new Request(s,level1);
                    Org.getBens().get(BenLoc).addRequest(R);
                    BenMenu1();
                }
            }
        }
    }
    else if (noPeople<=4 && noPeople>1) {
        double level2 = Org.getStock().get(i).getEn().getlevel2();
        if(level2>Org.getStock().get(i).getQuantity()){
            System.out.println("Sorry but this amount isnt available!");
            BenMenu1();
        }
    }
}

```

```

else {
    String Name = Org.getStock().get(i).getENAME();
    String Desc = Org.getStock().get(i).getEn().getDesc();
    Service s = new Service(Name, entityID, Desc);
    Request R = new Request(s, level2);
    Org.getBens().get(BenLoc).addRequest(R);
    BenMenu1();
}
}
else {
    double level3 = Org.getStock().get(i).getEn().getlevel3();
    if (level3 > Org.getStock().get(i).getQuantity()) {
        System.out.println("Sorry but this amount isnt available!"); BenMenu1();
    }
    else {
        String Name = Org.getStock().get(i).getENAME();
        String Desc = Org.getStock().get(i).getEn().getDesc();
        Service s = new Service(Name, entityID, Desc);
        Request R = new Request(s, level3);
        Org.getBens().get(BenLoc).addRequest(R);
        BenMenu1();
    }
}
}
}
}

if (answer == 2) {
    System.out.println(" Please enter the Service's ID and the requested quantity ");
    int entityID = sc.nextInt();
    double reQ = sc.nextDouble();
    for (int i = 0; i < Org.getStock().size(); i++) {

```

```

        if (Org.getStock().get(i).getEId()==entityID) {
            if(reQ>Org.getStock().get(i).getQuantity()) {
                System.out.println("Sorry but this amount isnt available!");
                BenMenu1();
            }
            else {
                String Name = Org.getStock().get(i).getENAME();
                String Desc =Org.getStock().get(i).getEn().getDesc();
                Service s = new Service(Name,entityID,Desc);
                Request R = new Request(s,reQ);
                Org.getBens().get(BenLoc).addRequest(R);
            }
        }
    }
}

else {System.out.println("Please enter a valid option! ");BenMenu1();}

}

if (answer==2) {BenMenu();}

else {System.out.println("Please enter a valid option");BenMenu1();}

return 0;

}

```

```

public int BenMenu2() {
    Org.getBens().get(BenLoc).showcaseRequestList();
    System.out.println("1.Delete Request/Change Quantity\n2.Clear Requests\n3.Back");
    answer = sc.nextInt();
    if (answer ==1) {
        System.out.println("Which Request would you like to modify ? (enter ID) ");
        int entityID=sc.nextInt();
        System.out.println("1.Delete Request\n2.Change Quantity");
        answer=sc.nextInt();
    }
}

```

```

if (answer==1){
    for(int i=0;i<Org.getBens().get(BenLoc).getRequestList().size();i++) {
        if (Org.getBens().get(BenLoc).getRequestList().get(i).getEId()==entityID) {
            Org.getBens().get(BenLoc).getRequestList().remove(i);
            BenMenu();
        }
    }
}

if (answer==2){
    System.out.println("Please enter a new quantity ");
    double newQ = sc.nextDouble();
    for(int i=0;i<Org.getBens().get(BenLoc).getRequestList().size();i++) {
        if (Org.getBens().get(BenLoc).getRequestList().get(i).getEId()==entityID) {
            Org.getBens().get(BenLoc).getRequestList().get(i).setQuantity(newQ);
            BenMenu();
        }
    }
}

else {
    System.out.println("Please enter a valid option ");
    BenMenu2();
}

if (answer ==2) {
    Org.getBens().get(BenLoc).clearRequestList();
    BenMenu();
}

if (answer ==3) {BenMenu();}

return 0;
}

```



```

public int BenMenu3() {
    for (int i=0;i<Org.getBens().get(BenLoc).getRequestList().size();i++){
        Org.CommitRequest(Org.getBens().get(BenLoc).getRequestList().get(i));
    }
    Org.getBens().get(BenLoc).clearRequestList();
    BenMenu();
    return 0;
}
}
import java.util.ArrayList;

```

```

public class Main
{
    public static void main(String[] args) {
        Admin Admin = new Admin("Alexander",694666420);
        Organization Org = new Organization("jerw gw bale ena onoma edw",Admin);
        //Service test = new Service("",0,"");
        //Offer f = new Offer(test,0);
        //Org.AddOffer(f);
        Material milk = new Material("Milk",001,"It comes in a carton",1.5,4,7);
        Material sugar = new Material("Sugar",002,"It comes in bags",1,3,4);
        Material rice = new Material("Rice",003,"It comes in bags",3,12,15);
        Service MedSup = new Service("MedicalSupport",004,"It not mental support!");
        Service NursSup = new Service("NurserySupport",005,"Its not medical support!");
        Service BabySit = new Service("BabySitting",006,"They come to take care of your
        babyyy!:DDD");
        Donator D1 = new Donator("D1",691);
        Beneficiary B1 = new Beneficiary("Bobby",692,1);
        Beneficiary B2 = new Beneficiary("B2",693,3);
        Org.addDon(D1);
    }
}

```

```

Org.addBen(B1);

Org.addBen(B2);

Org.addEn(milk);

Org.addEn(sugar);

Org.addEn(rice);

Org.addEn(MedSup);

Org.addEn(NursSup);

Org.addEn(BabySit);

```

```

Menu m = new Menu(Org);

```

```

m.Start();

```

```

}

```

```

}

```



