

PROIECT LA DISCIPLINA PROGRAMARE ORIENTATĂ PE OBIECTE

TEMA: Ghid Turistic

Titular disciplină:
Șef lucr. dr. ing. Ioniță Irina

Studenti:
Brânzea Ana - Maria
Dobrescu Alexandra -Ioana
Hudiță Corina - Elena

PLOIEȘTI
2022

CUPRINS

1. Formularea problemei ce urmează a fi soluționată.....	3
2. Abordare UML.....	4
2.1. Analiza aplicației	4
2.2. Proiectarea aplicației.....	6
2.3. Implementarea aplicației.....	6
2.4. Testarea aplicației.....	12
3. Actorii aplicației.....	13
4. Concluzii	13
5. Biografie	14
6. Anexe	15
Clasa Biserica	15
Clasa ListaBiserici	15
Clasa ListaMuzee	16
Clasa ListaOrase	19
Clasa ListaParcuri	19
Clasa Muzeu	20
Clasa Oras	21
Clasa Parc	23
Main	23

1. Formularea problemei ce urmează a fi soluționată

Aplicația pune la dispoziția utilizatorului toate informațiile necesare despre obiectivele turistice alese spre vizitare. Prin intermediul ei se pot afla, cu ușurință, care sunt punctele de atracție reprezentative pentru destinația aleasă.

În vederea soluționării problemei, prin folosirea claselor și subclaselor, dorim să punem la dispoziție un *ghid turistic* cât mai veridic. Printr-o interfață prietenoasă cu utilizatorul, acesta poate ajunge la informații despre obiectivul ales.

Problema inițială, a ghidului turistic, a fost rezolvată folosind următoarele clase și subclase:

- **ObiectivTuristic**
 - **Oras**
 - **ListaBiserici**
 - **ListaOrase**
 - **ListaParcuri**
 - **ListaMuzee**
 - Parc
 - Biserica
 - Muzeu
- } derivate din **ObiectivTuristic**

Etapetele care au fost urmărite pentru a rezolva subproblemele date sunt:

1. **Analiza** – crearea unui model formal al aplicației.
2. **Proiectarea** – definirea unei structuri și al comportamentului produsului software.
3. **Implementarea** – definirea limbajului de programare folosit și utilizarea acestuia pentru soluționarea problemei.
4. **Testarea** – verificarea aplicației prin analiza unor scenarii de evaluare pentru a elimina deficiențele de funcționare.

2. Abordare UML

2.1 Analiza aplicației

Aplicatia prezentată poate fi văzută ca având două feluri de useri: utilizatori și administratori. Fiecare dintre cei doi are acces la o serie de metode, astfel:

- ❖ Utilizatorii pot alege:
 - orașul dorit de vizitat
 - tipul obiectivului turistic
- ❖ Administratorii pot:
 - să modifice informațiile despre obiectivele turistice
 - să adauge obiective turistice noi
 - să șteargă/adauge categorii de obiective

Diagrama Use Cases

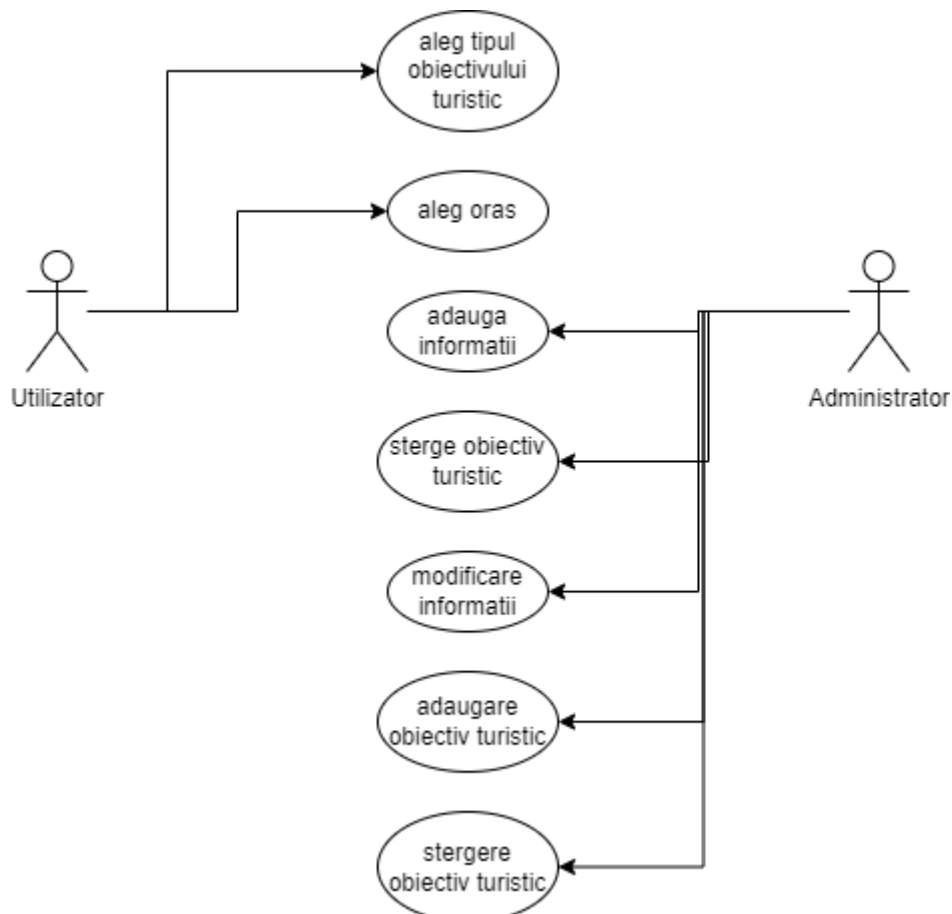
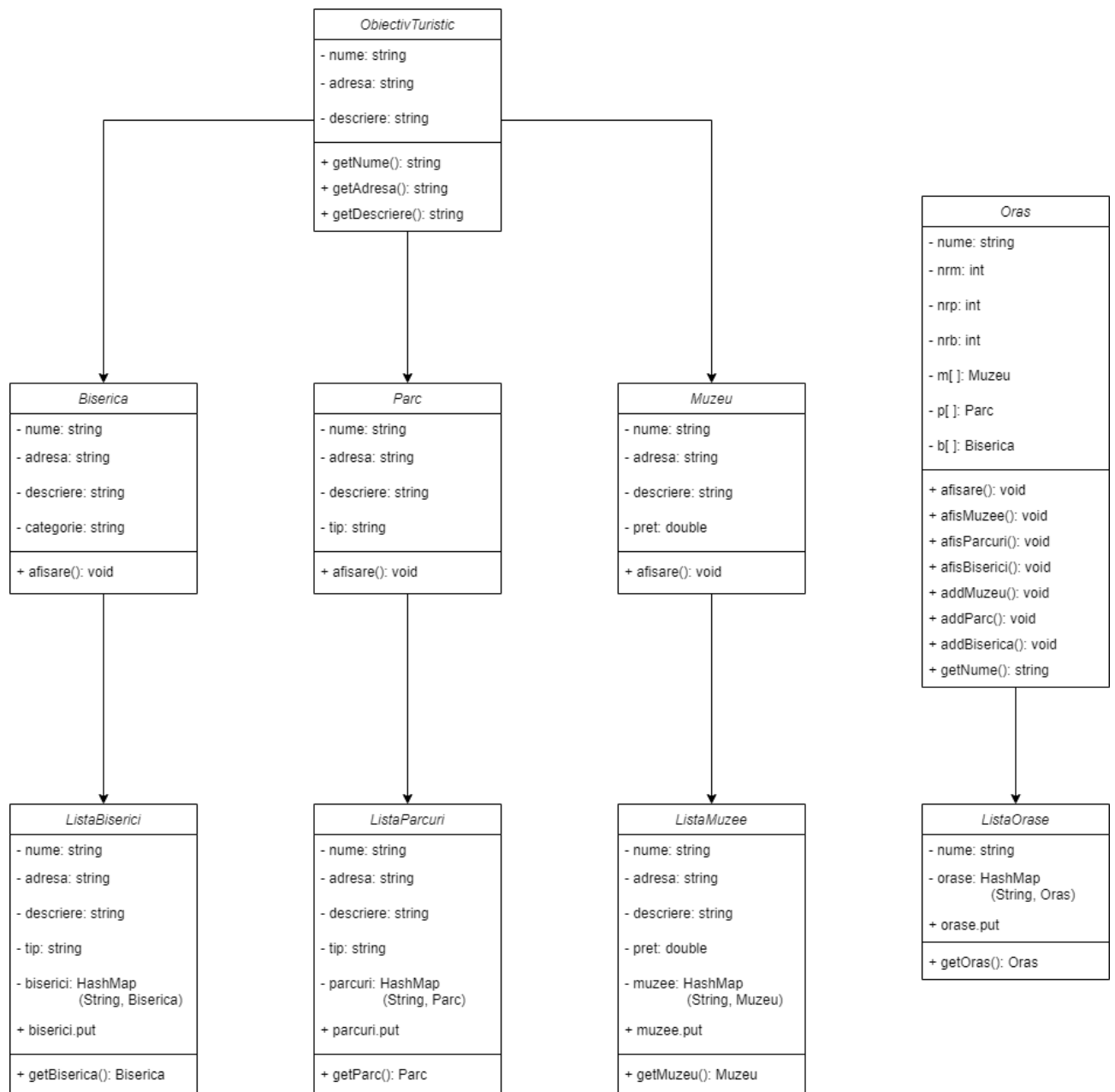


Diagrama UML a claselor



2.2 Proiectarea aplicației

Programul a fost scris folosind compilatorul **NetBeans** și acesta este format din nouă clase, după cum urmează:

- **ObiectivTuristic** - conține datele caracteristice ale unui obiectiv (nume,adresa,descriere)
 - **Oras** - este format din vectorii ce formează lista muzeelor, parcurilor și a bisericilor, aparținând orașului respectiv
 - **ListaBiserici** - stochează datele despre biserici
 - **ListaOrase** - stochează numele orașelor
 - **ListaParcuri** - stochează datele despre parcuri
 - **ListaMuzee** - stochează datele despre muzee
 - **Parc** - subclasa **ObiectivTuristic**
 - **Biserica** - subclasa **ObiectivTuristic**
 - **Muzeu** - subclasa **ObiectivTuristic**
 - **Main** - pentru începerea programului
- } conțin funcții de afișare

2.3 Implementarea aplicației (codarea propriu-zisă)

Clasa ObiectivTuristic cuprinde ca date membru: numele,adresa și descrierea fiecarui obiectiv. Ca metode, am folosit **constructorul cu parametrii** și funcții de tip **get**.

```
public class ObiectivTuristic {
    private String nume, adresa, descriere;

    public ObiectivTuristic(String nume, String adresa, String descriere){
        this.nume = nume;
        this.adresa = adresa;
        this.descriere = descriere;
    }

    public String getNume(){
        return nume;
    }

    public String getAdresa(){
        return adresa;
    }

    public String getDescriere()
    {
        return descriere;
    }
}
```

În **clasa Oras** s-au format obiectele de tip vector specifice fiecărei categorii (biserică, muzeu, parc) ce ne vor ajuta la adăugarea de date pentru un anumit oraș. De asemenea, s-au folosit metode de tipul:

- constructor implicit
- constructor cu parametri
- metodă ce afișează date membru
- metodă ce afișează toate datele membru(individual)
- metodă de adăugare a categoriilor de obiective (addMuzeu(), addParc(), addBiserica())

```
public void addMuzeu(Muzeu... xn)
{
    for(Muzeu x : xn) {
        m[nrm] = x;
        nrm++;
    }
}
```

```
public void addParc(Parc... xn)
{
    for(Parc x : xn) {
        p[nrp] = x;
        nrp++;
    }
}

public void addBiserica(Biserica... xn)
{
    for(Biserica x : xn) {
        b[nrb] = x;
        nrb++;
    }
}
```

- metodă de tip get pentru nume

Clasa ListaOrașe s-a folosit **clasa Java HashMap**. Aceasta implementează interfața Map care ne permite să stocăm datele în perechi (cheie, valoare) și le putem accesa printr-un index de alt tip(în cazul nostru, tipul String). Astfel, se vor putea asocia numele orașelor cu listele obiectivelor turistice. Se mai adaugă și metoda de tip **get** pentru orașe.

```
import java.util.HashMap;

public class ListaOrașe {

    private HashMap<String, Oras> orase;

    public ListaOrașe()
    {
        String nume[] = {"Bucuresti", "Targoviste", "Ploiesti", "Timisoara", "Piatra Neamt"};
        orase = new HashMap<String, Oras>();

        for(String n : nume) {
            orase.put(n, new Oras(n));
        }
    }
}
```

Pentru ca în continuare să stocăm informațiile necesare despre fiecare muzeu, parc, biserică am creat **clasele ListaMuzee, ListaParcuri și ListaBiserici**. Toate cele trei se folosesc de **clasa HashMap** pentru a grupa obiectivele turistice în funcție de orașul în care se afla. De asemenea, datele membru sunt: nume, adrese, descrieri, tip(doar pentru parcuri și biserici), pret (doar pentru muzee). Fiecare clasa dispune și de o metodă **get**.

```
import java.util.HashMap;

public class ListaMuzee {

    private HashMap<String, Muzeu> muzee;

    public ListaMuzee()
    {
        String nume[] = {"Muzeul National \"George Enescu\"", "Muzeul de Artă din Tim"};
        String adrese[] = {"Calea Victoriei nr. 141, sector 1", "Piata Unirii, nr. 1"};
        double pret[] = {8, 7, 10, 9, 12, 15, 10, 12, 8, 8, 15, 23, 17, 11, 12, 11};
        String descrieri[] = {"Muzeul Național „George Enescu” prezintă expoziții per
                                Muzeul de Artă din Timișoara este un muzeu de artă afl
                                Muzeul Național al Banatului este un muzeu din Timișoa
                                Muzeul Satului Bănățean din Timișoara este singurul mu
                                Colecția de artă bisericească veche a Arhiepiscopiei T
                                Cunoscut sub acronimul MNAC, promovează nume relevante
                                Muzeul Național al Satului \"Dimitrie Gusti\" din Bucu
                                Muzeul Ceasului este probabil cel mai cunoscut muzeu d
                                Muzeul a fost inaugurat în anul 1962, la aniversarea a
                                Casa de Targovet este cea mai veche casa din Ploiesti,
                                Curtea Domnească din Târgoviște a fost concepută ca un
                                Muzeul Scriitorilor Dâmbovițeni este un muzeu județean
                                Muzeul Național al Poliției Române este unicul muzeu c
                                Constituind unitatea de bază a Complexului Muzeal Jude
                                Muzeul de Artă Eneolitică Cucuteni a fost inaugurat pe
                                Consolidarea și punerea în valoare a ruinelor fostelor
                            };

        muzee = new HashMap<String, Muzeu>();
        for(int i=0;i<nume.length;i++){
            muzee.put(nume[i], new Muzeu(nume[i], adrese[i], descrieri[i], pret[i]));
        }
    }

    public Muzeu getMuzeu(String nume)
    {
        return muzee.get(nume);
    }
}
```

Clasele Biserica, Muzeu și Parc extind (moștenesc) clasa ObiectivTuristic,
fiecare având date membru specifice:

- **Clasa Biserica** - categorie (String)
- **Clasa Muzeu** -pret (double)
- **Clasa Parc** - tip (String)

Ca metode, s-au folosit constructori cu parametri (folosindu-se suprascrierea) și metode de afisare.


```

public class Parc extends ObiectivTuristic{
    private String tip;
    public Parc(String nume, String adresa, String descriere, String tip)
    {
        super(nume, adresa, descriere);
        this.tip=tip;
    }

    public void afisare(){
        System.out.println(getNume() + "\n    Tip: " + tip + "\n    Adresa: " + getAdresa()+"\n"+getDescriere());
        System.out.println("-----");
    }
}

public class Biserica extends ObiectivTuristic {
    private String categorie;
    public Biserica(String nume,String adresa, String descriere, String categorie)
    {
        super(nume, adresa, descriere);
        this.categorie=categorie;
    }

    public void afisare(){
        System.out.println(getNume() + "\n    Categorie: " + categorie + "\n    Adresa: " + getAdresa()+"\n"+ getDescriere());
        System.out.println("-----");
    }
}

public class Muzeu extends ObiectivTuristic{
    private double pret;
    public Muzeu(String nume, String adresa, String descriere, double pret)
    {
        super(nume, adresa, descriere);
        this.pret = pret;
    }

    public void afisare()
    {
        System.out.println(getNume() + "\n    Adresa: " + getAdresa()+"\n Pret intrare: "+pret+" RON"+"\n"+getDescriere());
        System.out.println("-----");
    }
}

```

În ceea ce privește **functia main**, aici s-au construit obiectele: **LO** din clasa ListaOrașe, **LM** din ListaMuzee, **LP** din ListaParcuri, **LB** din ListaBiserici. Fiecareia i s-au atribuit obiectivele turistice caracteristice prin intermediul **funcției add**.

```

LO.getOras("Timisoara").addMuzeu(LM.getMuzeu("Muzeul de Arta din Timisoara"), LM.getMuzeu("Muzeul Banatului din Timis-
LO.getOras("Timisoara").addParc(LP.getParc("Parcul Civic"), LP.getParc("Parcul Rozelor"), LP.getParc("Parcul Copiilor
LO.getOras("Timisoara").addBiserica(LB.getBiserica("Catedrala Mitropolitana Ortodoxa"), LB.getBiserica("Biserica Nast

LO.getOras("Bucuresti").addMuzeu(LM.getMuzeu("Muzeul National \"George Enescu\""), LM.getMuzeu("Muzeul National de Art
LO.getOras("Bucuresti").addParc(LP.getParc("Parcul IOR"), LP.getParc("Parcul Bazilescu"));
LO.getOras("Bucuresti").addBiserica(LB.getBiserica("Catedrala Sf.Iosif"), LB.getBiserica("Biserica Elefterie"));

LO.getOras("Ploiesti").addMuzeu(LM.getMuzeu("Muzeul Ceasului"), LM.getMuzeu("Muzeul I.L. Caragiale"), LM.getMuzeu("Ca
LO.getOras("Ploiesti").addParc(LP.getParc("Parcul Tineretului"), LP.getParc("Parcul Bucov"));
LO.getOras("Ploiesti").addBiserica(LB.getBiserica("Catedrala Sfântul Ioan Botezătorul"), LB.getBiserica("Biserica Rom

LO.getOras("Targoviste").addMuzeu(LM.getMuzeu("Complexul Muzeal \"Curtea Domnească\""), LM.getMuzeu("Muzeul Scriitori
LO.getOras("Targoviste").addParc(LP.getParc("Parcul Chindia"),LP.getParc("Parcul Mitropoliei"));
LO.getOras("Targoviste").addBiserica(LB.getBiserica("Biserica \"Sf. Nifon - Sarbi\" Targoviste"),LB.getBiserica("Casa

LO.getOras("Piatra Neamt").addMuzeu(LM.getMuzeu("Muzeul de Istorie și Arheologie Piatra-Neamț"), LM.getMuzeu("Muzeul
LO.getOras("Piatra Neamt").addParc(LP.getParc("Parcul Tineretului"), LP.getParc("Belvedere"), LP.getParc("Parcul Petr
LO.getOras("Piatra Neamt").addBiserica(LB.getBiserica("Biserica Sfântul Ioan"),LB.getBiserica("Biserica Nașterea Sfân

```

Mai departe, prin intermediul **variabilei op** introdusă de către utilizator de la tastatura, se v-a alege un oraș despre care se vor afla mai multe informații. Cu ajutorul funcției **switch()** se realizează filtrarea tuturor informațiilor și afișarea acelor care depind de opțiunea aleasa.

```
int op=in.nextInt();
while(op!=0)
{
    switch (op)
    {
        case 1:
        {
            LO.getOras("Timisoara").afisare();
            System.out.print("Alegeti categoria: 1.Biserici/2.Parcuri/3.Muzee\n");
            int t1=in.nextInt();
            while(t1!=4)
            {
                if(t1==1) LO.getOras("Timisoara").afisBiserici();
                if(t1==2) LO.getOras("Timisoara").afisParcuri();
                if(t1==3) LO.getOras("Timisoara").afisMuzee();
                System.out.println("Doriti sa alegeti o alta categorie? 1.Biserici/2.Parcuri/3.Muzee/4.Nu");
                t1=in.nextInt();
            }
            break;
        }
        case 2:
        {
            LO.getOras("Bucuresti").afisare();
            System.out.print("Alegeti categoria: 1.Biserici/2.Parcuri/3.Muzee\n");
            int t1=in.nextInt();
            while(t1!=4)
            {
                if(t1==1) LO.getOras("Bucuresti").afisBiserici();
                if(t1==2) LO.getOras("Bucuresti").afisParcuri();
                if(t1==3) LO.getOras("Bucuresti").afisMuzee();
                System.out.println("Doriti sa alegeti o alta categorie? 1.Biserici/2.Parcuri/3.Muzee/4.Nu ");
                t1=in.nextInt();
            }
            break;
        }
        case 3:
        case 3:
        {
            LO.getOras("Ploiesti").afisare();
            System.out.print("Alegeti categoria: 1.Biserici/2.Parcuri/3.Muzee\n");
            int t1=in.nextInt();
            while(t1!=4)
            {
                if(t1==1) LO.getOras("Ploiesti").afisBiserici();
                if(t1==2) LO.getOras("Ploiesti").afisParcuri();
                if(t1==3) LO.getOras("Ploiesti").afisMuzee();
                System.out.println("Doriti sa alegeti o alta categorie? 1.Biserici/2.Parcuri/3.Muzee/4.Nu");
                t1=in.nextInt();
            }
            break;
        }
    }
}
```

```
case 4:
{
    LO.getOras("Targoviste").afisare();
    System.out.print("Alegeti categoria: 1.Biserici/2.Parcuri/3.Muzee\n");
    int t1=in.nextInt();
    while(t1!=4)
    {
        if(t1==1) LO.getOras("Targoviste").afisBiserici();
        if(t1==2) LO.getOras("Targoviste").afisParcuri();
        if(t1==3) LO.getOras("Targoviste").afisMuzee();
        System.out.println("Doriti sa alegeti o alta categorie? 1.Biserici/2.Parcuri/3.Muzee/4.Nu");
        t1=in.nextInt();
    }
    break;
}
case 5:
{
    LO.getOras("Piatra Neamt").afisare();
    System.out.print("Alegeti categoria: 1.Biserici/2.Parcuri/3.Muzee\n");
    int t1=in.nextInt();
    while(t1!=4)
    {
        if(t1==1) LO.getOras("Piatra Neamt").afisBiserici();
        if(t1==2) LO.getOras("Piatra Neamt").afisParcuri();
        if(t1==3) LO.getOras("Piatra Neamt").afisMuzee();
        System.out.println("Doriti sa alegeti o alta categorie? 1.Biserici/2.Parcuri/3.Muzee/4.Nu");
        t1=in.nextInt();
    }
    break;
}
}
```

De asemenea, se poate reveni la meniul principal dacă utilizatorul dorește să afle informații despre alte orașe.

```
System.out.println("Doriti sa reveniti la meniul principal? 1.Da/2.Nu");
int v=in.nextInt();
if(v==2){ System.out.println("Va multumim pentru vizita!"); break;}
else if(v==1){
    System.out.print("\n Alegeti un oras:\n 1.Timisoara\n 2.Bucuresti\n 3.Ploiesti\n 4.Targoviste\n 5.Piatra Neamt\n OPTIUNEA- ");
    op=in.nextInt();
}
else {System.out.println("S-a introdus o optiune invalida");break;}
```

2.4 Tastarea aplicației

Aplicația a fost supusa diferitor teste, astfel încât să ofere un rezultat sigur, fără erori. Așadar, ca studiu de caz, analizăm interacțiunea utilizatorului cu aplicația în cauza:

Odată cu deschiderea aplicației, utilizatorul este pus să aleagă un oraș dintre cele disponibile:

```
Bine ati venit!  
Alegeti un oras:  
1.Timisoara  
2.Bucuresti  
3.Ploiesti  
4.Targoviste  
5.Piatra Neamt  
OPTIUNEA- |
```

După introducerea numărului ce este precedat de numele orașului dorit, se va confirma opțiunea aleasă. Mai apoi, utilizatorul este pus din nou în poziția de a-și alege categoria care îl interesează, iar după ce aplicația afișează informații, mai poate alege alte categorii.

```
Bine ati venit!  
Alegeti un oras:  
1.Timisoara  
2.Bucuresti  
3.Ploiesti  
4.Targoviste  
5.Piatra Neamt  
OPTIUNEA- 3  
  
* Orasul Ploiesti *  
Alegeti categoria: 1.Biserici/2.Parcuri/3.Muzee  
2  
  
Parcuri:  
1. Parcul Tineretului  
Tip: Antropic  
Adresa: Ploiesti  
Fara descriere.  
-----  
2. Parcul Bucov  
Tip: Antropic  
Adresa: Strada Daciei nr.11, Comuna Bucov  
Fara descriere.  
-----  
Doriti sa alegeti o alta categorie? 1.Biserici/2.Parcuri/3.Muzee/4.Nu
```

Dacă se merge pe ramura 4.Nu, se va pune o întrebare: “Doriți să reveniți la meniul principal? 1.Da/2.Nu”. În funcție de alegere, se afișează:

- Cazul 1:

Doriti sa reveniti la meniul principal? 1.Da/2.Nu
1

Alegeti un oras:

- 1.Timisoara
- 2.Bucuresti
- 3.Ploiesti
- 4.Targoviste
- 5.Piatra Neamt
- OPTIUNEA-

- Cazul 2:

Doriti sa alegeti o alta categorie? 1.Biserici/2.Parcuri/3.Muzee/4.Nu
4

Doriti sa reveniti la meniul principal? 1.Da/2.Nu
2

Va multumim pentru vizita!

3.Actorii aplicației

1. Actori interni – administrator
2. Actori externi – utilizator

4. Concluzii

După interacționarea cu aplicația prezentată mai sus, considerăm ca este începutul unei aplicații ce poate avea un real succes pe viitor în ceea ce privește promovarea turismului.

Puncte slabe: - lipsa unei interfete grafice
- număr restrans de orașe și obiective turistice

Ideii pentru îmbunătățirea aplicației

- adăugarea unui program pentru muzee
- adăugarea unei secțiuni de recenzii

Bibliografie

1. https://www.w3schools.com/java/java_hashmap.asp
2. Ioniță, L., et al, Diagramele UML 2. Dicționar. Studii de caz. Aplicație Web, Editura Universității Petrol-Gaze din Ploiești, 2015.
3. https://www.w3schools.com/java/java_hashset.asp
4. <https://stackoverflow.com/questions/33779319/java-util-hashmap-neo4j-multi-threaded-hanging-forever>
5. <https://www.geeksforgeeks.org/java-util-hashmap-in-java-with-examples/>

Anexe

Codul claselor din cadrul aplicației:

Clasa Biserica:

```
package proiect;

public class Biserica extends ObiectivTuristic {
    private String categorie;
    public Biserica(String nume,String adresa, String descriere, String categorie)
    {
        super(nume, adresa, descriere);
        this.categorie=categorie;
    }

    public void afisare(){
        System.out.println(getNume() + "\n Categorie: " + categorie + "\n Adresa: "
+ getAdresa()+"\n"+ getDescriere());
        System.out.println("-----");
    }
}
```

Clasa ListaBiserici:

```
package proiect;

import java.util.HashMap;

public class ListaBiserici {

    private HashMap<String, Biserica> biserici;

    public ListaBiserici()
    {
        String nume[] = {"Catedrala Mitropolitana Ortodoxa", "Biserica Nasterea Maicii Domnului", "Biserica Adormirea Maicii Domnului", "Catedrala Sf.Iosif", "Biserica Elefterie", "Catedrala Sfântul Ioan Botezătorul", "Biserica Romano-Catolică \"Cristos Rege\"", "Biserica \"Sf. Nifon - Sarbi\" Targoviste", "Casa de rugaciune evanghelica Targoviste", "Biserica Domnească din Târgoviște", "Biserica Sfântul Ioan", "Biserica Nașterea Sfântului Ioan Botezătorul din Piatra Neamț"};
        String adrese[] = {"Bulevardul Regele Ferdinand I", "Piata Alexandru Mocioni nr. 9", "Strada Dej", "Palatul Arhiepiscopal, Strada General H. M. Berthelot 19", "Strada Sfântul Elefterie 3", "Bd. Republicii nr. 12", "Strada Ștefan cel Mare 13", "Strada Bucuresti 19", "Strada Filipescu Nicolae 47", "Calea Domnească 181", "Strada Privighetorii 1", "Str. Ștefan cel Mare"};
        String tip[] = {"Ortodoxa", "Ortodoxa", "Protestanta", "Catolica", "Ortodoxa", "Orthodox", "Catholic", "Ortodoxa", "Evangelic", "Ortodoxa", "Ortodoxa", "Ortodoxa"};
        String descrieri[] = {"Catedrala Mitropolitana din Timisoara este cel mai mare edificiu religios din Timisoara, catedrala a Mitropoliei Banatului, cu hramul „Trei Ierarhi”. A fost construita intre 1936 si 1941 si este un simbol al orasului."};
    }
}
```

"Biserica este amplasata in Piata Alexandru Mocioni, cunoscuta in timp si sub numele de Piata Küttel, Piata Asanesti sau Piata Stefan Furtuna. Desi se considera ca partea sudica a pietei (unde este amplasata biserica) face parte din cartierul Elisabetin, biserica s-a construit pentru enoriasii ortodocsi din cartierul Iosefin (care inconjoara piata pe celelalte trei parti).",

"Fara descriere.",

"Monument istoric, cultural și arhitectonic înscris în patrimoniul național și european, Catedrala Sfântul Iosif este un reper vizibil al Bisericii Catolice pe pământul românesc și totodată un punct de atracție, spiritual și artistic.",

"Un momunemt impunator atat din afara, cat si in interior, unde se afla o impresionanta catapeteasma din lemn de cires, unicat in Romania. La fel ca in cazul bisericilor precedente, costul ceremoniei este considerabil, dar slujba oficiata de patru preoti si corul absolut incantator va pot convinge. Am observat ca a intrat in lucrari de renovare, inasa e posibil ca pana la vara sa fie finalizate.",

"Catedrala Sfântul Ioan Botezătorul, situată pe Bulevardul Republicii nr. 12, este cea mai impunătoare biserică din Ploiești și unul din principalele repere arhitectonice ale municipiului. Clopotnița catedralei este încadrată pe lista monumentelor istorice din județul Prahova.",

"Fara descriere.",

"Fara descriere.",

"Fara descriere.",

"Fara descriere.",

"Fara descriere.",

"Fara descriere.",

};

```
biserici = new HashMap<String, Biserica>();
for(int i=0;i<nume.length;i++){
    biserici.put(nume[i], new Biserica(nume[i], adrese[i], descrieri[i],
tip[i]));
}
}
public Biserica getBiserica(String nume)
{
    return biserici.get(nume);
}
}
```

Clasa ListaMuzee:

```
package proiect;
```

```
import java.util.HashMap;
```

```
public class ListaMuzee {
```

```
    private HashMap<String, Muzeu> muzee;
```

```
    public ListaMuzee()
    {
```

```
        String nume[] = {"Muzeul National \"George Enescu\"", "Muzeul de Arta din Timisoara", "Muzeul Banatului din Timisoara", "Muzeul Satului Banatean", "Muzeul Catedralei Mitropolitane", "Muzeul National de Arta Contemporana", "Muzeul Satului \"Dimitrie Gusti\"", "Muzeul Ceasului", "Muzeul I.L. Caragiale", "Casa de
```


Universitatea Petrol-Gaze din Ploiești
Facultatea de Litere și Științe
Specializarea Informatică

Targovet", "Complexul Muzeal \"Curtea Domnească\"", "Muzeul Scriitorilor Târgovișteni", "Muzeul Național al Poliției Române", "Muzeul de Istorie și Arheologie Piatra-Neamț", "Muzeul de Artă Eneolitică Cucuteni", "Expoziția Curtea Domnească Piatra-Neamț"};

String adrese[] = {"Calea Victoriei nr. 141, sector 1", "Piata Unirii, nr. 1", "Piata Iancu Huniade, nr. 1", "Strada Avram Imbroane, nr. 1", "Bulevardul Regele Ferdinand I", "Strada Izvor 2-4, Aripa E4 în incinta Palatului Parlamentului", "Soseaua Kiseleff nr. 28-30", "Strada Nicolae Simache nr. 1", "Str. Kutuzov, nr. 1", "Str. Democratiei, nr. 2", "Str. Calea Domneasca nr.181", "Str. Justiției, Nr.7", "Str: Calea Domnească nr. 187", "Strada Mihai Eminescu nr.10", "Strada Ștefan cel Mare nr.3", "Piața Curtea Domnească nr.4"};

double pret[] = {8, 7, 10, 9, 12, 15, 10, 12, 8, 8, 15, 23, 17, 11, 12, 11};

String descrieri[] = {"Muzeul Național „George Enescu” prezintă expoziții permanente la toate sediile, expoziții ce ilustrează viața și opera marelui muzician, cât și expoziții temporare, cu teme diverse. Potrivit specificului său aparte, Muzeul Național „George Enescu” susține, prin programele culturale permanente pe care le organizează, promovarea creației muzicale românești și străine, precum și a artei interpretative."},

"Muzeul de Artă din Timișoara este un muzeu de artă aflat în Palatul Baroc din Timișoara. Muzeul a luat ființă după desprinderea secțiunii de artă a Muzeului Banatului, care a funcționat o perioadă într-o aripă din actuala clădire.",

"Muzeul Național al Banatului este un muzeu din Timișoara cu sediul central în Castelul Huniade. A fost înființat în 1872, sub numele de \"Societatea de Istorie și Arheologie\". Adăpostește cea mai importantă colecție de obiecte arheologice din Banat.",

"Muzeul Satului Bănățean din Timișoara este singurul muzeu cu profil etnografic din România care cuprinde centrul civic al satului, format din Primărie, Biserică, Școală, Casă Națională și birt, în care se desfășoară majoritatea activităților cultural-educative și științifice ale unei localități. ",

"Colecția de artă bisericească veche a Arhiepiscopiei Timișoarei așezată în subsolul Catedralei mitropolitane din Timișoara. Patrimoniul de artă bisericească adăpostește astăzi aproape 1.000 de piese printre care un însemnat număr de icoane pe lemn, pe sticlă și pânză, obiecte de cult brodate, sculptate sau lucrate în metal și un important fond de carte veche religioasă.",

"Cunoscut sub acronimul MNAC, promovează nume relevante ale artei contemporane românești și artiști internaționali de renume.",

"Muzeul Național al Satului \"Dimitrie Gusti\" din București, România este \"Muzeul din inima capitalei\", care și-a deschis porțile pentru public în anul 1936. Expoziția permanentă acoperă o suprafață de 14 ha având 380 de monumente, 60.000 de obiecte în colecțiile de patrimoniu și peste 250.000 de documente de arhivă referitoare la sat și viața sa tradițională.",

"Muzeul Ceasului este probabil cel mai cunoscut muzeu din Ploiești. Este unic în Europa iar în prezent muzeul funcționează într-o clădire monument de arhitectură din anul 1890. Muzeul a fost înființat în anul 1963 de către profesorul ploiestean Nicolae Simache. Muzeul are o colecție de aproape 1000 de piese printre care: cadran solar, pendul construit din lemn în 1634, ceas pus în mișcare printr-o cadere de apă (Londra, 1654), primul ceas de buzunar („oul de Nürnberg”).",

"Muzeul a fost inaugurat în anul 1962, la aniversarea a 110 ani de la nașterea marelui dramaturg, în clădirea monument de arhitectură din secolul XVIII – Casa Dobrescu. Ion Luca Caragiale s-a născut la 30 ianuarie 1852 în satul Haimanale (care astăzi îi poartă numele), de lângă Ploiești, județul Prahova. Clasele primare și gimnaziul le-a absolvit la Ploiești, iar între anii 1868 – 1870, a frecventat cursurile de declamație și mimă la Conservatorul din București ale unchiului său, Costache Caragiale.",

"Casa de Targovet este cea mai veche casa din Ploiești, având peste 200 de ani. Construită în 1785 de bogatul negustor Hagi Prodan, remarcata

Universitatea Petrol-Gaze din Ploiești
Facultatea de Litere și Științe
Specializarea Informatică

de Nicolae Iorga, casa devine după Primul Razboi Mondial „Muzeul Prahovei”. Din 1953, prin eforturile profesorului Nicolae Simache, muzeul își schimbă numele în Muzeul „Hagi Prodan”.

"Curtea Domnească din Târgoviște a fost concepută ca un complex de edificii care să servească atât ca centru al administrației țării, cât și ca locuință pentru domnitor, familia sa și pentru numeroșii curteni și slujitori. Acesta era locul în care se țineau consfătuirile divanului, se efectuau judecățile mai importante și se eliberau actele cancelariei. Tot aici erau primiți oaspeții și călătorii străini aflați cu diferite prilejuri prin Țara Românească.",

"Muzeul Scriitorilor Dâmbovițeni este un muzeu județean din Târgoviște, amplasat în Str. Justiției nr. 7. Casa în care a trăit, a visat și a creat scriitorul Ioan Alexandru Brătescu-Voinești găzduiește acum Muzeul Scriitorilor Dâmbovițeni, inaugurat la 30 decembrie 1968.",

"Muzeul Național al Poliției Române este unicul muzeu cu acest profil din țara noastră, fiind inaugurat la data de 7 iulie 2000. Este așezat vizavi de Curtea Domnească, în aceeași incintă cu Muzeul de Artă și cel de Istorie. O vizită la muzeu ne prilejuiește cunoașterea diferitelor etape de dezvoltare ale Poliției Române, oglindind drumul parcurs de această instituție în evoluția sa, atât pe plan național, cât și internațional.",

"Constituind unitatea de bază a Complexului Muzeal Județean Neamț - înființat în 1978, Muzeul de Istorie și Arheologie Piatra-Neamț se înscrie în categoria celor mai prestigioase instituții de profil din România. Contribuția sa la cercetarea, conservarea, restaurarea, gestionarea, evidența și valorificarea patrimoniului istoric și arheologic din estul țării este absolut remarcabilă, așa după cum este și activitatea sa cultural-educativă.",

"Muzeul de Artă Eneolitică Cucuteni a fost inaugurat pe data de 24 iunie 2005, de Ziua municipiului Piatra-Neamț, în preajma aniversării a 120 de ani de la descoperirea stațiunii eponime a culturii Cucuteni și a 70 de ani de la înființarea Muzeului de Istorie și Arheologie. La baza înființării sale a stat Hotărârea Guvernului României, nr. 109 din 5 februarie 2004, prin care clădirile dintr-o serie de municipii reședință de județ, între care și Piatra-Neamț, ce aparținuseră filialelor Băncii Naționale a României, erau trecute în subordinea muzeelor județene din zonele respective, cu scopul de a fi transformate în unități muzeale.",

"Consolidarea și punerea în valoare a ruinelor fostelor Case domnești de la Piatra Neamț este rodul strădaniilor Preotului Constantin Matasă. De altfel, la insistențele acestuia s-au și descoperit ruinele curții de epocă ștefaniană în anii 1954-1955 când, cu prilejul cercetărilor arheologice de la Suceava, s-a găsit de cuviință a se include în proiect și platoul unde se presupunea existența unei astfel de construcții."

};

```
    muzee = new HashMap<String, Muzeu>();
    for(int i=0;i<nume.length;i++){
        muzee.put(nume[i], new Muzeu(nume[i], adrese[i], descrieri[i], pret[i]));
    }
    public Muzeu getMuzeu(String nume)
    {
        return muzee.get(nume);
    }
}
```

Clasa ListaOrase:

```
package proiect;
```

```
import java.util.HashMap;

public class ListaOrase {

    private HashMap<String, Oras> orase;

    public ListaOrase()
    {
        String nume[] = {"Bucuresti", "Targoviste", "Ploiesti", "Timisoara", "Piatra
Neamt"};
        orase = new HashMap<String, Oras>();

        for(String n : nume){
            // Oras o = new Oras(n);
            // orase.put(n, o);
            orase.put(n, new Oras(n));
        }
    }
    public Oras getOras(String nume)
    {
        return orase.get(nume);
    }
}
```

Clasa ListaParcuri:

```
package proiect;

import java.util.HashMap;

public class ListaParcuri {

    private HashMap<String, Parc> parcuri;

    public ListaParcuri()
    {
        String nume[] = {"Parcul Civic", "Parcul Rozelor", "Parcul Copiilor Ion
Creanga", "Parcul IOR", "Parcul Bazilescu", "Parcul Tineretului", "Parcul
Bucov", "Parcul Chindia", "Parcul Mitropoliei", "Parcul Tineretului", "Belvedere", "Parcul
Petrodava"};
        String adrese[] = {"Centru", "Centru", "Bulevardul Constantin Diaconovici Loga
43", "Sector 3", "Cartierul Bucurestii Noi", "", "", "Calea Domneasca", "Bulevardul
Mircea cel Bătrân", "", "", ""};
        String tip[] = {"Antropic", "Natural", "Antropic", "Natural", "Antropic",
"Natural", "Antropic", "Natural", "Antropic", "Antropic", "Antropic", "Natural"};
        String descrieri[] = {"Parcul Civic sau Parcul Cetății este situat în zona
centrală a Timișoarei, în spatele Hotelului Continental, având o suprafață de 76.000
mp.",
"Parcul Rozelor se află în zona centrală a orașului, în
apropierea Catedralei Mitropolitane Ortodoxe și reprezintă una dintre atracțiile
turistice ale Timișoarei. Peste 10.000 de trandafiri roșii, roz, albi, galbeni și mov
sunt grupați în funcție de soiuri, fiind delimitați prin garduri vii de buxus.",
"Parcul Copiilor "Ion Creangă" din Timișoara a fost
amenajat pentru prima dată în anul 1858, sub numele de Parcul Franz Iosef.",
"Parcul IOR, actualmente Parcul Alexandru Ioan Cuza, este
situat în sectorul 3 și este unul dintre cele mai mari parcuri din
```

Universitatea Petrol-Gaze din Ploiești
Facultatea de Litere și Științe
Specializarea Informatică

București. Acesta este compus din Parcul Alexandru Ioan Cuza și Parcul Titan. În parc se află Biserica \"Pogorârea Sfântului Duh\" construită din lemn în stil maramureșan între anii 1994-1996.\",

\"Parcul Bazilescu (fostul parc Nicolae Bălcescu) este un parc aflat în cartierul Bucureștii Noi și are o suprafață de 13,5 hectare. În interiorul parcului se află Teatrul de Vară și Clubul Copiilor Sector 1 - Bazilescu, unde școlarii pot merge la diferite cercuri educative: muzică, teatru, gimnastică ritmică, limba engleză, educație civică, rugby.\",

\"\",
\"\",

\"Parcul Chindia a fost amenajat pe fosta grădină domnească, ce era situată în spatele Curții Domnești în anul 1970. În parc se găsește un lac de agrement pe care se pot face plimbări cu barca și hidrobicicleta. De asemenea, este amenajată o pistă de ciclism, patinaj pe role și atletism. În anul 2006 a fost inaugurată Aleea celor 33 de voievozi care au domnit în vechea Cetate de Scaun de la Târgoviște.\",

\"Parcul Mitropoliei este foarte cunoscut târgoviștenilor. Pentru că se află în centrul orașului și pentru că este singura oază de liniște în agitația cotidiană, parcul de aici poate fi considerat un loc perfect pentru o pauză de la serviciu sau pentru relaxarea de după program.\",

\"_\",
\"_\",
\"_\"

```
};

    parcuri = new HashMap<String, Parc>();
    for(int i=0;i<nume.length;i++){
        parcuri.put(nume[i], new Parc(nume[i], adrese[i], descrieri[i], tip[i]));
    }
}
public Parc getParc(String nume)
{
    return parcuri.get(nume);
}
}
```

Clasa Muzeu:

```
package proiect;
```

```
public class Muzeu extends ObiectivTuristic{
```

```
    private double pret;
```

```
    public Muzeu(String nume, String adresa, String descriere, double pret)
```

```
    {
        super(nume, adresa, descriere);
        this.pret = pret;
    }
```

```
    public void afisare()
```

```
    {
        System.out.println(getNume() + "\n    Adresa: " + getAdresa()+"\n Pret intrare: "+pret+" RON"+"\\n"+getDescriere());
        System.out.println("-----");
    }
```

}

Clasa ObiectivTuristic:

```
package proiect;
```

```
public class ObiectivTuristic {  
    private String nume, adresa, descriere;  
  
    public ObiectivTuristic(String nume, String adresa, String descriere){  
        this.nume = nume;  
        this.adresa = adresa;  
        this.descriere = descriere;  
    }  
  
    public String getNume(){  
        return nume;  
    }  
  
    public String getAdresa(){  
        return adresa;  
    }  
  
    public String getDescriere()  
    {  
        return descriere;  
    }  
}
```

Clasa Oras:

```
package proiect;
```

```
public class Oras {  
    private String nume;  
    Muzeu [] m;  
    int nrm;  
    Parc [] p;  
    int nrp;  
    Biserica [] b;  
    int nrb;  
    public Oras()  
    {  
        nume="";  
        m=new Muzeu[50];  
        nrm=0;  
        p = new Parc[50];  
        nrp = 0;  
        b = new Biserica[50];  
        nrb = 0;  
    }  
    public Oras(String nume)  
    {  
        this.nume= nume;
```

```
        m=new Muzeu[50];
        nrm=0;
        p = new Parc[50];
        nrp = 0;
        b = new Biserica[50];
        nrb = 0;
    }

    public void afisare()
    {
        System.out.println("\n* Orasul "+ nume+" *");
    }

    public void afisMuzee()
    {
        System.out.println("Muzee:");
        for(int i=0;i<nrm;i++){
            System.out.print((i + 1) +". ");
            m[i].afisare();
        }
    }

    public void afisParcuri()
    {
        System.out.println("Parcuri:");
        for(int i=0;i<nrp;i++){
            System.out.print((i + 1) +". ");
            p[i].afisare();
        }
    }

    public void afisBiserici()
    {
        System.out.println("Biserici:");
        for(int i=0;i<nrb;i++){
            System.out.print((i + 1) +". ");
            b[i].afisare();
        }
    }

    public void addMuzeu(Muzeu... xn)
    {
        for(Muzeu x : xn){
            m[nrm] = x;
            nrm++;
        }
    }

    public void addParc(Parc... xn)
    {
        for(Parc x : xn){
            p[nrp] = x;
            nrp++;
        }
    }

    public void addBiserica(Biserica... xn)
    {
        for(Biserica x : xn){
            b[nrb] = x;
            nrb++;
        }
    }
}
```

```
    public String getNume()
    {
        return nume;
    }
}
```

Clasa Parc:

```
package proiect;

public class Parc extends ObiectivTuristic{
    private String tip;
    public Parc(String nume, String adresa, String descriere, String tip)
    {
        super(nume, adresa, descriere);
        this.tip=tip;
    }

    public void afisare(){
        System.out.println(getNume() + "\n    Tip: " + tip + "\n    Adresa: " +
getAdresa()+"\n"+getDescriere());
        System.out.println("-----");
    }
}
```

main:

```
package proiect;
import java.util.Scanner;

public class PROIECT {
    public static void main(String[] args) {
        ListaOrase LO = new ListaOrase();
        ListaMuzee LM = new ListaMuzee();
        ListaParcuri LP = new ListaParcuri();
        ListaBiserici LB = new ListaBiserici();
        Scanner in=new Scanner(System.in);
        LO.getOras("Timisoara").addMuzeu(LM.getMuzeu("Muzeul de Arta din Timisoara"),
LM.getMuzeu("Muzeul Banatului din Timisoara"), LM.getMuzeu("Muzeul Satului Banatean"),
LM.getMuzeu("Muzeul Catedralei Mitropolitane"));
        LO.getOras("Timisoara").addParc(LP.getParc("Parcul Civic"), LP.getParc("Parcul
Rozelor"), LP.getParc("Parcul Copiilor Ion Creanga"));
        LO.getOras("Timisoara").addBiserica(LB.getBiserica("Catedrala Mitropolitana
Ortodoxa"), LB.getBiserica("Biserica Nasterea Maicii Domnului"),
LB.getBiserica("Biserica Adormirea Maicii Domnului"));

        LO.getOras("Bucuresti").addMuzeu(LM.getMuzeu("Muzeul National \"George
Enescu\""),LM.getMuzeu("Muzeul National de Arta Contemporana"), LM.getMuzeu("Muzeul
Satului \"Dimitrie Gusti\""));
        LO.getOras("Bucuresti").addParc(LP.getParc("Parcul IOR"), LP.getParc("Parcul
Bazilescu"));
```

Universitatea Petrol-Gaze din Ploiești
Facultatea de Litere și Științe
Specializarea Informatică

```
LO.getOras("Bucuresti").addBiserica(LB.getBiserica("Catedrala Sf.Iosif"),  
LB.getBiserica("Biserica Elefterie"));
```

```
LO.getOras("Ploiesti").addMuzeu(LM.getMuzeu("Muzeul Ceasului"),  
LM.getMuzeu("Muzeul I.L. Caragiale"), LM.getMuzeu("Casa de Targovet"));  
LO.getOras("Ploiesti").addParc(LP.getParc("Parcul Tineretului"),  
LP.getParc("Parcul Bucov"));  
LO.getOras("Ploiesti").addBiserica(LB.getBiserica("Catedrala Sfântul Ioan  
Botezătorul"), LB.getBiserica("Biserica Romano-Catolică \"Cristos Rege\""));
```

```
LO.getOras("Targoviste").addMuzeu(LM.getMuzeu("Complexul Muzeal \"Curtea  
Domnească\""), LM.getMuzeu("Muzeul Scriitorilor Târgovișteni"),LM.getMuzeu("Muzeul  
Național al Poliției Române"));  
LO.getOras("Targoviste").addParc(LP.getParc("Parcul  
Chindia"),LP.getParc("Parcul Mitropoliei"));  
LO.getOras("Targoviste").addBiserica(LB.getBiserica("Biserica \"Sf. Nifon -  
Sarbi\" Targoviste"),LB.getBiserica("Casa de rugaciune evanghelică  
Targoviste"),LB.getBiserica("Biserica Domnească din Târgoviște"));
```

```
LO.getOras("Piatra Neamt").addMuzeu(LM.getMuzeu("Muzeul de Istorie și  
Arheologie Piatra-Neamț"), LM.getMuzeu("Muzeul de Artă Eneolitică Cucuteni"),  
LM.getMuzeu("Expoziția Curtea Domnească Piatra-Neamț"));  
LO.getOras("Piatra Neamt").addParc(LP.getParc("Parcul Tineretului"),  
LP.getParc("Belvedere"), LP.getParc("Parcul Petrodava"));  
LO.getOras("Piatra Neamt").addBiserica(LB.getBiserica("Biserica Sfântul  
Ioan"),LB.getBiserica("Biserica Nașterea Sfântului Ioan Botezătorul din Piatra  
Neamț"));
```

```
System.out.print("          Bine ati venit! \n Alegeti un oras:\n 1.Timisoara\n  
2.Bucuresti\n 3.Ploiesti\n 4.Targoviste\n 5.Piatra Neamt\n OPTIUNEA- ");  
int op=in.nextInt();  
while(op!=0)  
{  
    switch (op)  
    {  
        case 1:  
        {  
            LO.getOras("Timisoara").afisare();  
            System.out.print("Alegeti categoria: 1.Biserici/2.Parcuri/3.Muzee\n");  
            int t1=in.nextInt();  
            while(t1!=4)  
            {  
                if(t1==1) LO.getOras("Timisoara").afisBiserici();  
                if(t1==2) LO.getOras("Timisoara").afisParcuri();  
                if(t1==3) LO.getOras("Timisoara").afisMuzee();  
                System.out.println("Doriti sa alegeti o alta categorie?  
1.Biserici/2.Parcuri/3.Muzee/4.Nu");  
                t1=in.nextInt();  
            }  
            break;  
        }  
        case 2:  
        {  
            LO.getOras("Bucuresti").afisare();  
            System.out.print("Alegeti categoria: 1.Biserici/2.Parcuri/3.Muzee\n");  
            int t1=in.nextInt();
```



```
        while(t1!=4)
        {
            if(t1==1) LO.getOras("Bucuresti").afisBiserici();
            if(t1==2) LO.getOras("Bucuresti").afisParcuri();
            if(t1==3) LO.getOras("Bucuresti").afisMuzee();
            System.out.println("Doriti sa alegeti o alta categorie?
1.Biserici/2.Parcuri/3.Muzee/4.Nu ");
            t1=in.nextInt();
        }
        break;
    }
    case 3:
    {
        LO.getOras("Ploiesti").afisare();
        System.out.print("Alegeti categoria: 1.Biserici/2.Parcuri/3.Muzee\n");
        int t1=in.nextInt();
        while(t1!=4)
        {
            if(t1==1) LO.getOras("Ploiesti").afisBiserici();
            if(t1==2) LO.getOras("Ploiesti").afisParcuri();
            if(t1==3) LO.getOras("Ploiesti").afisMuzee();
            System.out.println("Doriti sa alegeti o alta categorie?
1.Biserici/2.Parcuri/3.Muzee/4.Nu");
            t1=in.nextInt();
        }
        break;
    }
    case 4:
    {
        LO.getOras("Targoviste").afisare();
        System.out.print("Alegeti categoria: 1.Biserici/2.Parcuri/3.Muzee\n");
        int t1=in.nextInt();
        while(t1!=4)
        {
            if(t1==1) LO.getOras("Targoviste").afisBiserici();
            if(t1==2) LO.getOras("Targoviste").afisParcuri();
            if(t1==3) LO.getOras("Targoviste").afisMuzee();
            System.out.println("Doriti sa alegeti o alta categorie?
1.Biserici/2.Parcuri/3.Muzee/4.Nu");
            t1=in.nextInt();
        }
        break;
    }
    case 5:
    {
        LO.getOras("Piatra Neamt").afisare();
        System.out.print("Alegeti categoria: 1.Biserici/2.Parcuri/3.Muzee\n");
        int t1=in.nextInt();
        while(t1!=4)
        {
            if(t1==1) LO.getOras("Piatra Neamt").afisBiserici();
            if(t1==2) LO.getOras("Piatra Neamt").afisParcuri();
            if(t1==3) LO.getOras("Piatra Neamt").afisMuzee();
            System.out.println("Doriti sa alegeti o alta categorie?
1.Biserici/2.Parcuri/3.Muzee/4.Nu");
            t1=in.nextInt();
        }
        break;
    }
```

```
        }  
    }  
    System.out.println("Doriti sa reveniti la meniul principal? 1.Da/2.Nu");  
    int v=in.nextInt();  
    if(v==2){ System.out.println("Va multumim pentru vizita!"); break;}  
    else if(v==1){  
        System.out.print("\n Alegeti un oras:\n 1.Timisoara\n 2.Bucuresti\n  
3.Ploiesti\n 4.Targoviste\n 5.Piatra Neamt\n OPTIUNEA- ");  
        op=in.nextInt();  
    }  
    else {System.out.println("S-a introdus o optiune invalida");break;}  
}  
}
```