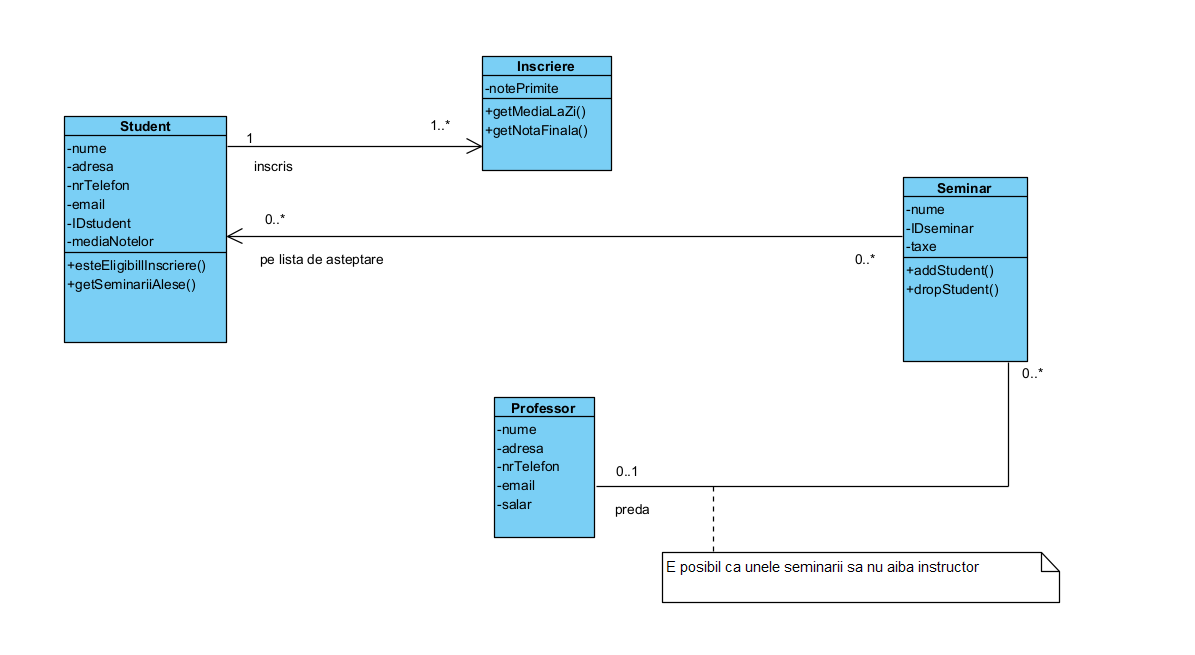
LAB 6

2.

* Un student se inscrie la unul sau mai multe seminarii si poate primi note. Relatia este de asociere unidirectionala intre Student si Inscriere.
* Niciunul sau mai multi studenti pot fi pe lista de asteptare la niciunul sau mai multe seminarii. Relatia este de asociere unidirectionala intre Seminar si Student.
* Seminariile pot sau nu sa aiba un professor care sa predea. Relatia este de asociere bidirectionala intre Seminar si Profesor.

3.

1. (d) Realizare între clasa Carte și interfața Produs; clasa Carte implementează interfața Produs; clasa Carte e o compoziție de obiecte de tip Capitol.

2. (c) private Collection<Articol> capitole = new Collection();

(f) class Revista implements Produs{...}

3. (a) private CodSursa theCode;

(c) private Collection<DiagramaUML> diagrams = new Collection();

(g) public void setGrade(int g){...}

4. (b) Un obiect de tip Student conține o colecție de obiecte de tip Proiect

(d) Clasa Student are operația publică adaugăProiect

(f) Clasa Student este superclasă pentru clasa StudentBursier

5. (b) class Profesor {

private Materie preda; ...}

6. (a) Clasa Materie definește o compoziție de obiecte de tip Curs

(d) Clasa Materie definește un agregat de obiecte de tip Laborator.

(e) Un obiect de tip Materie conține o colecție de obiecte de tip Test

7. (c) class Materie {

private Collection <Laborator> laboratoare = new Collection();...}

class Laborator {

private Materie materie;

...}

8. (d) Asociere unidirecțională, numită preda, de la clasa Profesor la clasa Materie; compoziție între clasele Materie(compozit) și Curs(componenta); compoziție între clasele Materie(compozit) și Laborator(componenta); compoziție între clasele Materie(compozit) și Test(componenta).

9. (b) private float pret;

(e) class Carte implements Produs{...}

(g) public float getPret(){...}

(i) private Collection<Capitol> capitole = new Collection();

10. (b) abstract class Angajat {

private String nume;

private int CNP;

public abstract void calculPlata();

...}

11. (a) class AngajatCuOra extends Angajat{...}

(d) public calculPlata(){...};

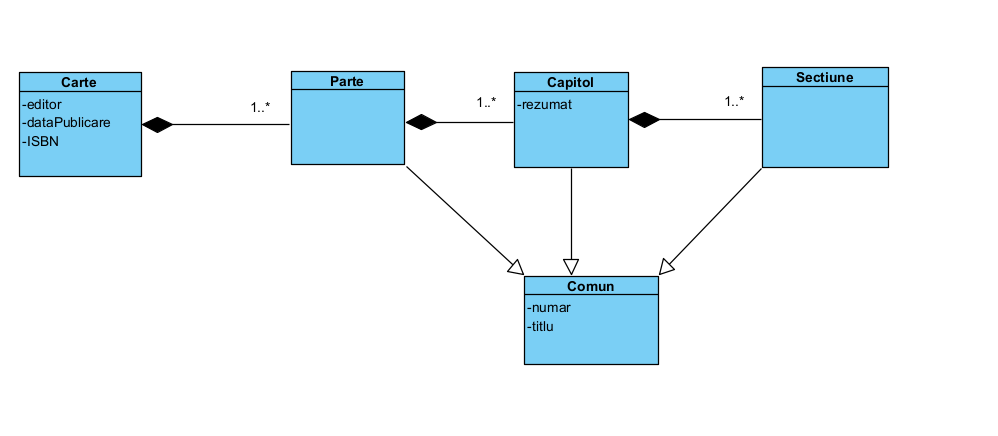
12. (c) private static int urmatorulCod = 1;

(d) public static int getUrmatorulCod(){...};

(e) private Client client;

13. (c) Clasa Proiect definește un agregat de obiecte de tip Student și compoziții de obiecte de tip DiagrameUML și de tip RaportTestare.

(e) Clasa Proiect este în relație de asociere cu clasa CodSursa.

4.

import java.util.ArrayList;

import java.util.List;

class Carte {

    private String editor;

    private String dataPublicare;

    private String ISBN;

    private List<Parte> parti;

    public Carte(String editor, String dataPublicare, String ISBN) {

        this.editor = editor;

        this.dataPublicare = dataPublicare;

        this.ISBN = ISBN;

        this.parti = new ArrayList<>();

    }

    public void adaugaParte(Parte parte) {

        parti.add(parte);

    }

    public List<Parte> getParti() {

        return parti;

    }

}

class Parte {

    private String titlu;

    private int numar;

    private List<Capitol> capitole;

    public Parte(String titlu, int numar) {

        this.titlu = titlu;

        this.numar = numar;

        this.capitole = new ArrayList<>();

    }

    public void adaugaCapitol(Capitol capitol) {

        capitole.add(capitol);

    }

    public List<Capitol> getCapitole() {

        return capitole;

    }

}

class Capitol {

    private String titlu;

    private int numar;

    private String rezumat;

    private List<Sectiune> sectiuni;

    public Capitol(String titlu, int numar, String rezumat) {

        this.titlu = titlu;

        this.numar = numar;

        this.rezumat = rezumat;

        this.sectiuni = new ArrayList<>();

    }

    public void adaugaSectiune(Sectiune sectiune) {

        sectiuni.add(sectiune);

    }

    public List<Sectiune> getSectiuni() {

        return sectiuni;

    }

}

class Sectiune {

    private String titlu;

    private int numar;

    public Sectiune(String titlu, int numar) {

        this.titlu = titlu;

        this.numar = numar;

    }

}