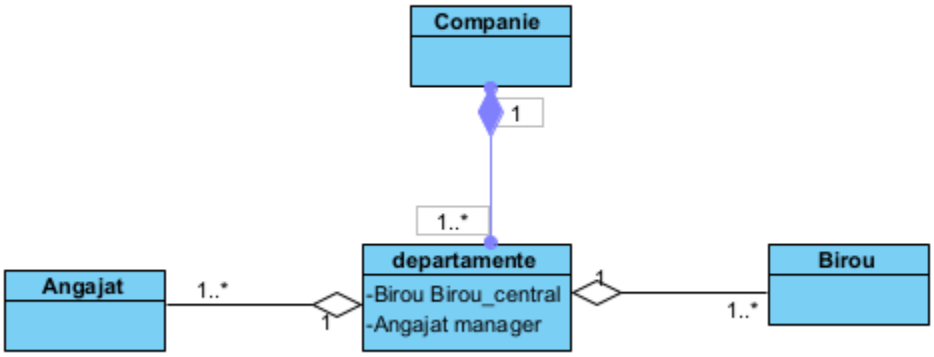
4 lab 5:




```java
public class Angajat {

    @Override
    public String toString() {
        return "Angajat []";
    }
```

```java
public class Birou {

    @Override
    public String toString() {
        return "Birou []";
    }

}
```

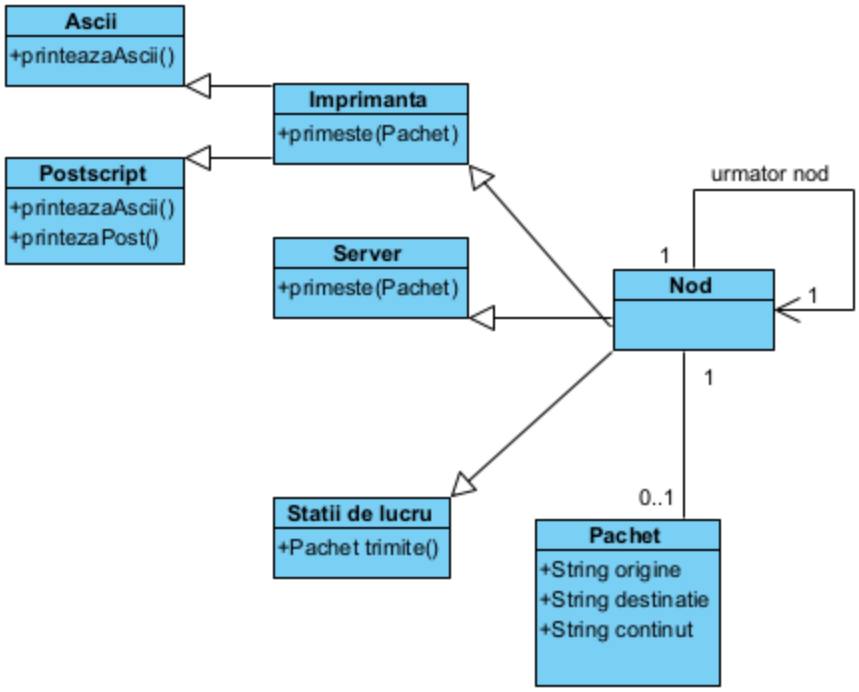
```java
public class Departament {
    public Birou getBiroucentral() {
        return biroucentral;
    }
    public void setBiroucentral(Birou biroucentral) {
        this.biroucentral = biroucentral;
    }
    public ArrayList<Birou> getBirouri() {
        return birouri;
    }
    public void setBirouri(ArrayList<Birou> birouri) {
        this.birouri = birouri;
    }
    public Angajat getManager() {
        return manager;
    }
    public void setManager(Angajat manager) {
        this.manager = manager;
    }
    public ArrayList<Angajat> getAngajati() {
        return angajati;
    }
    public void setAngajati(ArrayList<Angajat> angajati) {
        this.angajati = angajati;
    }
    public Birou biroucentral;
    public ArrayList<Birou> birouri;
    public Angajat manager;
    public ArrayList<Angajat> angajati;
    @Override
    public String toString() {
        return "Departament [biroucentral=" + biroucentral + ", birouri=" + birouri + ", manager=" + mana
                + ", angajati=" + angajati + "]";
    }
}
```

```java
public class Companie {
    public ArrayList<Departament> departamente;

    public Companie() {
        departamente = new ArrayList<Departament>();
    }

    @Override
    public String toString() {
        return "Companie [departamente=" + departamente + "]";
    }
}
```

5:




```java
public class Pachet {
    public String origine;
    public String destinatie;
    public String continut;
}
```

```java
public class Nod {
    public Nod nod_urmator;
    public Pachet pachet;
}
```

```java
public class StatiiDeLucru extends Nod {
    public Pachet trimite()
    {
        return this.pachet;
    }
}
```

```java
public class Server extends Nod {
    public void primeste(Pachet p) {
        this.pachet=p;
    }
}
```

```java
public class Imprimanta extends Nod {
    public void primeste(Pachet p) {
        this.pachet=p;
    }
}
```

```
public class Ascii extends Imprimanta {
    public void printeazaAscii(){
        //
    }
```

```
public class PostScript {
    public void printeazaAscii(){
        //
    }
    public void printeazaPostScript(){
        //
    }
}
```

E1:

A: fals produs este intefata

B:fals Realizare in loc de generalizare

C: fals produs este intefata

D: corect

E: capitol si carte sunt ivers

E2:

A: fals ar trebui sa implementeze

B: fals produs si revista sunt invers

C: corect

D: fals trebuie sa implementeze metoda

E: fals metoda e publica

F: corect

G: fals nu are tipul de return produs

E3:

A:corect

B: fals grade si int trebuie inversate
C:corect
D: fals ar trebui sa fie private
E:fals ar trebui sa specifice int ca tip de return
F:fals nu este generalizare intre ele

G:corect

H:fals e relatie de compozitie , ar trebui vector

E4:

A: fals mostenirea e invers

B: corect

C: fals curs e privat

D:corect

E: fals operatia e publica

F:corect

E5:

A: fals nu e generalizare

B: corect
C: fals profesor si materie sunt invers

D: fals clasa ar trebui sa fie profesor si sa nu fie vector

E6:

A: corect

B: fals e unidirectionala

C: este compozitie nu generalizare

D:fals compozitie in loc de agregat

E: corect

E7:

A: fals nu e generalizare
B: fals nu e generalizare
C: corect
D: fals laborator si materie sut invers
E: fals nu defineste o colectie noua
E8:

A: fals asociere unidirectionala

B: fals compozitie in loc de agregare
C: fals materie nu e superclasa

D: corect

E: fals clasele nu mostenesc

E9:

A: fals implementeaza
B: corect
C:fals pret si float sunt invers
D:fals trebuie implementata
E:corect
F:fals public
G:corect

H:fals nu are tip de return produs
I:corect

E10:

A: fals clasa e abstracta

B: corect

C: fals metoda abstracta nu trebuie implementata
D:fals metoda nu e abstracta

E11:

A: corect

B: fals nu e interfata
C: fals metoda trebuie implementata
D: corect

E12:

A: fals nu e mostenire

B: fals e statica variabila
C:corect
D: corect
E: corect

F: fals metoda nu e statica
E13:

A: fals proiect defineste in loc de student

B: fals defineste agregari cu student
C: corect
D: fals defineste o agregare cu student
E: corect