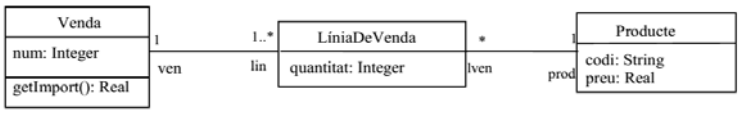
IES - polimorfisme: solucions i codi Java dels exercicis de polimorfisme

FIB

Departament d'Enginyeria de Serveis i Sistemes d'Informació
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Exercici 1



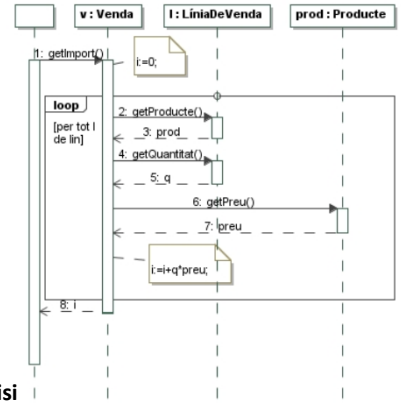
i donat el contracte següent:

context Venda::getImport():Real

post: result= suma de quantitat * preu del producte de totes les línies de la venda

Fer el diagrama de seqüència de l'operació *getImport():Real* i de totes les operacions que siguin invocades en aquesta operació. Poseu comentaris de tot el que no hi aparegui de forma explícita. Cal que indiqueu al diagrama de classes donat la navegabilitat resultant del vostre disseny.

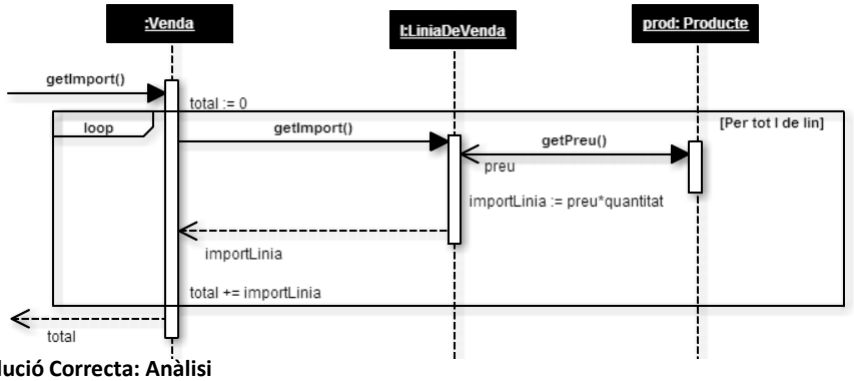
Exercici 1 (Alternativa 1) – Diagrama de Seqüència



Solució Incorrecta: Anàlisi

Acoblament: La classe Venda estaria acoblada amb LíniaDeVenda i Producte.
La classe LíniaDeVenda estaria acoblada amb Producte.
Reusabilitat: No hi ha una operació que ens permeti calcular l'import d'una línia de venda, operació potencialment reusable.
Violació de la llei de Demeter: Venda no ha de parlar amb Producte

Exercici 1 (Alternativa 2) – Diagrama de Seqüència



Solució Correcta: Anàlisi

Acoblament: La classe Venda només estaria acoblada amb LíniaDeVenda.
La classe LíniaDeVenda estaria acoblada amb Producte.
Reusabilitat: L'operació *getImportLín():float* potencialment reusable.
Satisfacció de la llei de Demeter: Venda només parla amb LíniaDeVenda (conegut)

Exercici 1 (Alternativa 2) – El codi


```
public class Venda{
    private int num
    private Set<LiniaDeVenda> lin

    public double getImport(){
        double total = 0
        for(LiniaDeVenda l in lin){
            double importLinia = l.getImport()
            total += importLinia
        }
        return total
    }
}
```

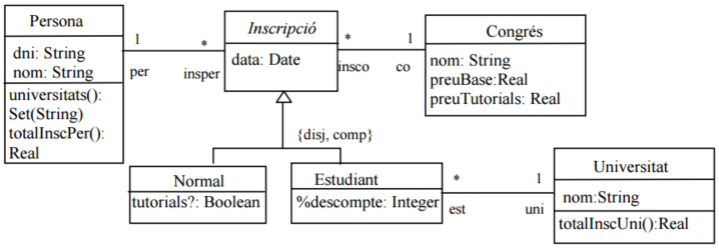
Exercici 1 (Alternativa 2) – El codi


```
public class LiniaVenda{
    private int quantitat
    private Producte prod

    public double getImport(){
        double preu = prod.getPreu()
        double importLinia = preu*quantitat
        return importLinia
    }
}
```

Exercici 1 (Alternativa 2) – El codi


```
public class Producte{
        private String codi
        private double preu

        public double getPreu(){
                return preu
        }
}
```

Exercici 2



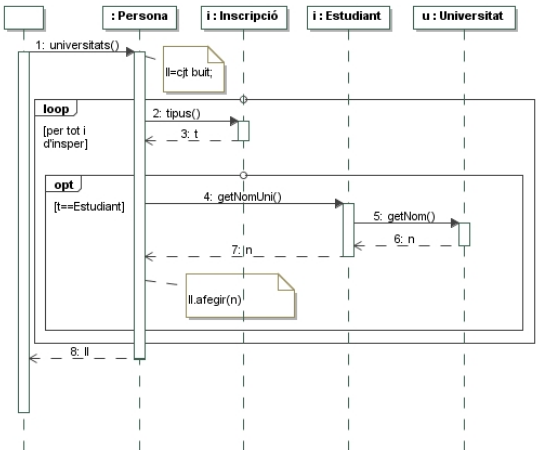
i donat el contracte següent:

context Persona::universitats():Set(String)

post: result= tots els noms de les universitats per les quals la persona té alguna inscripció com a estudiant

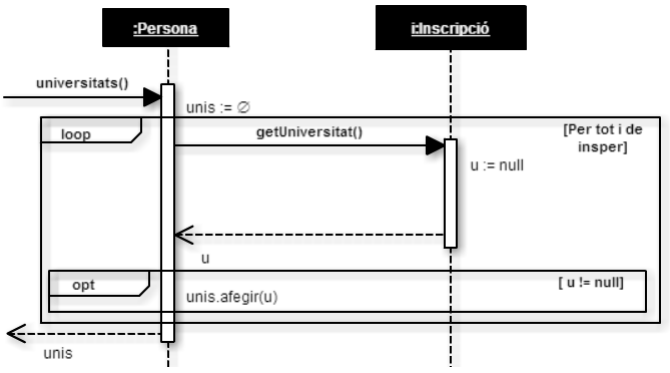
Fer el diagrama de seqüència de l'operació *universitats():Set(String)* i de totes les operacions que siguin invocades en aquesta operació. Poseu comentaris de tot el que no hi aparegui de forma explícita. Cal que indiqueu al diagrama de classes donat la navegabilitat resultant del vostre disseny.

Exercici 2 (Alternativa 1) – Diagrama de Seqüència



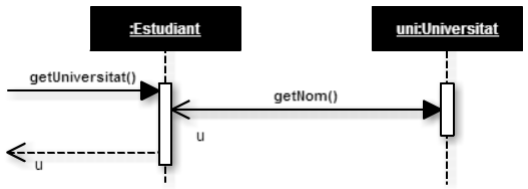
Solució Incorrecta: Anàlisi
Violació del principi d'Obert-Tancat

Exercici 2 (Alternativa 2) – Diagrama de Seqüència

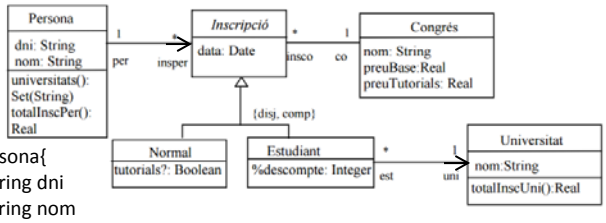


Solució Correcta: Anàlisi
Satisfacció del principi d'Obert-Tancat

Exercici 2 (Alternativa 2) – Diagrama de Seqüència

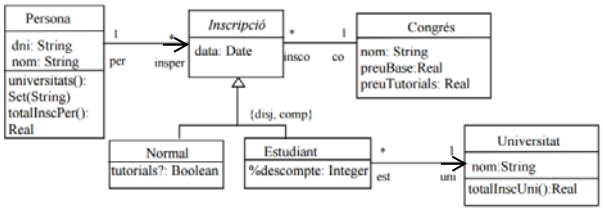


Exercici 2 (Alternativa 2) – El codi



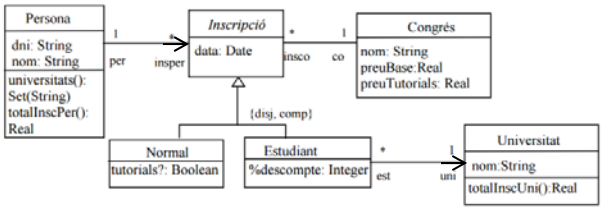

```
public class Persona{
        private String dni
        private String nom
        private Set<Inscripcio> insper

        public Set<String> universitats(){
                Set<String> unis = new Set()
                for (Inscripcio i in insper){
                        String u = i.getUniversitat()
                        if(u != null) unis.add(u)
                }
                return unis
        }

}
```

Exercici 2 (Alternativa 2) – El codi



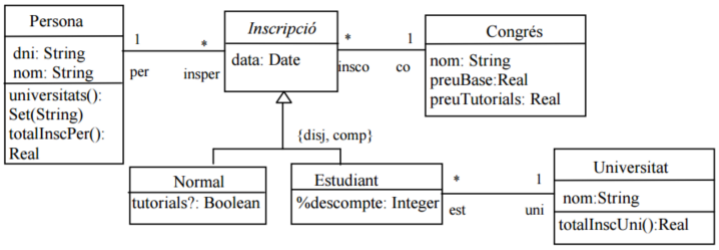

```
public abstract class Inscripcio{
        private Date data

        public String getUniversitat(){
                String u = null
                return u
        }

}
```

Exercici 2 (Alternativa 2) – El codi




```
public class InscripcioEstudiant extends Inscripcio{
        private int descompte
        private Universitat uni

        public String getUniversitat(){
                String u = uni.getNom()
                return u
        }

}
```

Exercici 3



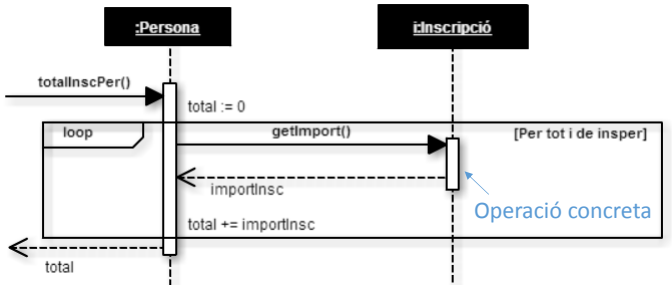
i donat el contracte següent:

context Persona::totalInscPer():Real

post: result= suma dels imports de totes les inscripcions de la persona. L'import d'una inscripció es calcula com segueix:

- si la inscripció és de tipus estudiant l'import és el preu base del congrés menys el descompte segons el %descompte que tingui
- si la inscripció és de tipus normal l'import és el preu base del congrés més el preu dels tutorials si tutorials? és cert.

Exercici 3 (Alternativa 1) – Diagrama de Seqüència



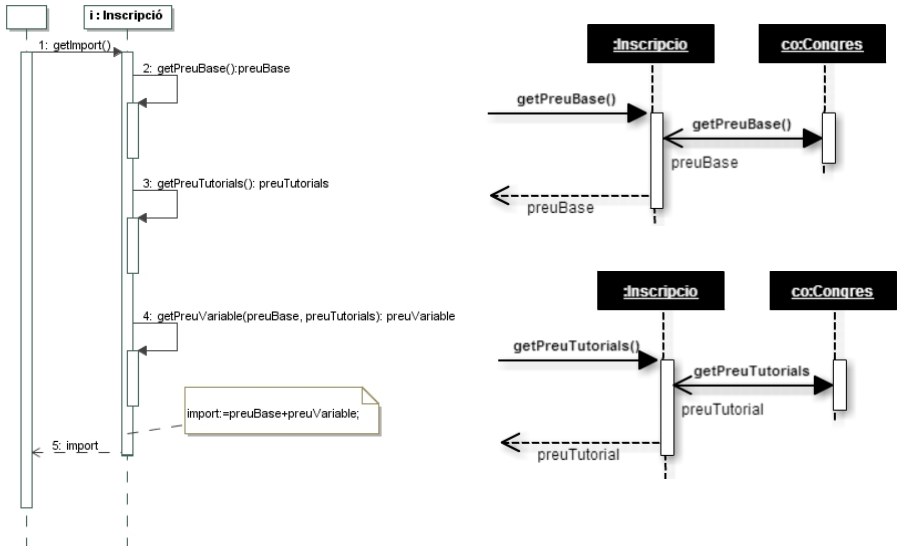
Solució Correcta: Anàlisi

No repetició de codi

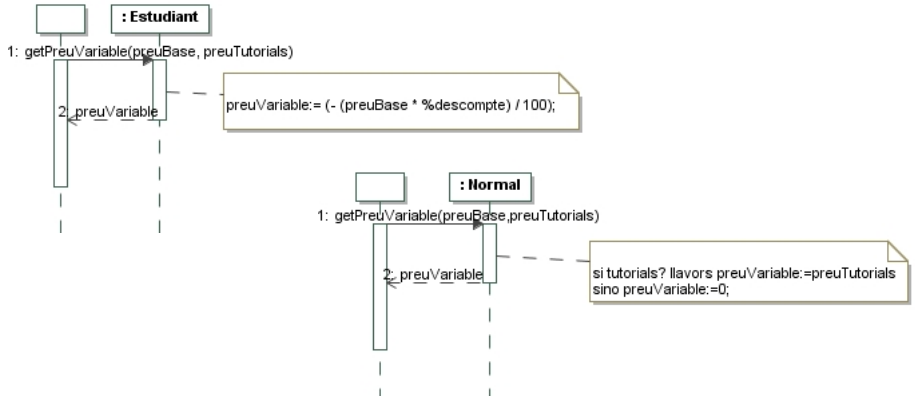
Ineficient: Es calcula el preuTutorials abans de saber si cal fer-ho.

Llegibilitat (understandability) del disseny: Passem paràmetres que no són utilitzats. Afecta a la canviabilitat.

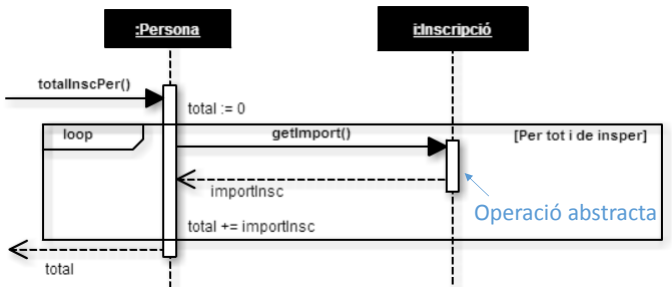
Exercici 3 (Alternativa 1) – Diagrama de Seqüència



Exercici 3 (Alternativa 1) – Diagrama de Seqüència

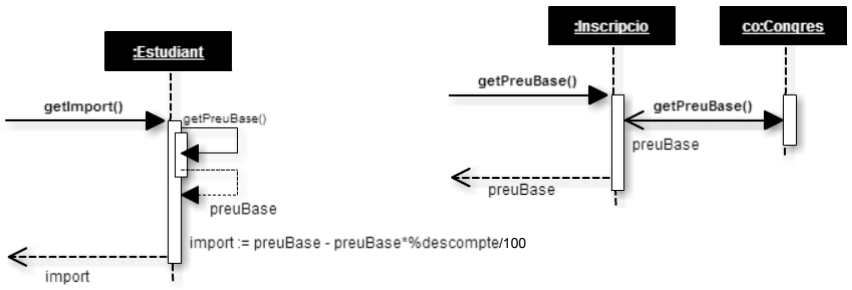


Exercici 3 (Alternativa 2) – Diagrama de Seqüència

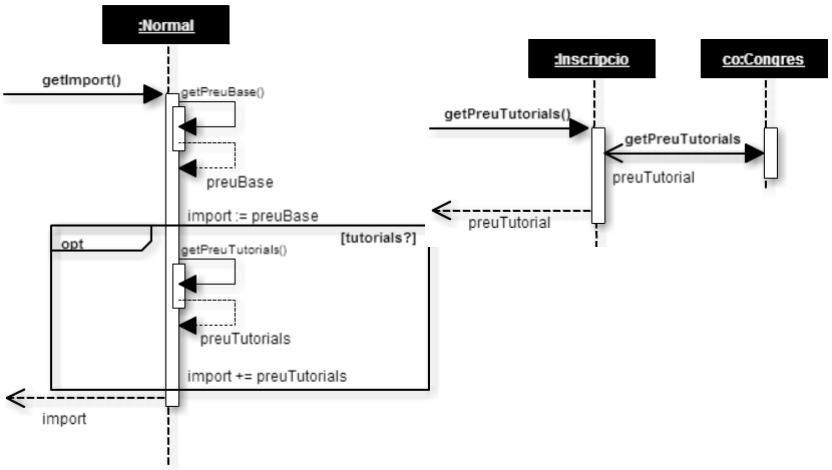


Solució Correcta: Anàlisi
Repetició de codi: Les incrispcions normals i d'estudiant tenen una part del càlcul de l'import que es repeteix. Repetir el codi a les dues operacions va en contra de la canviabilitat.

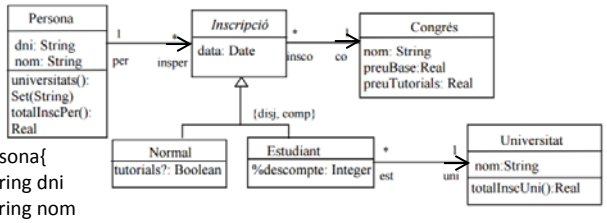
Exercici 3 (Alternativa 2) – Diagrama de Seqüència



Exercici 3 (Alternativa 2) – Diagrama de Seqüència

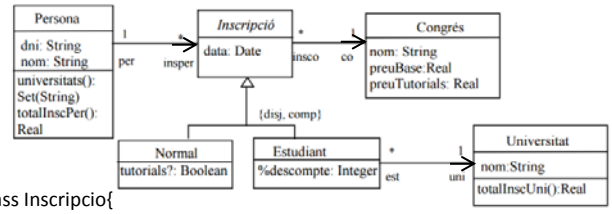


Exercici 3 (Alternativa 2) – El codi



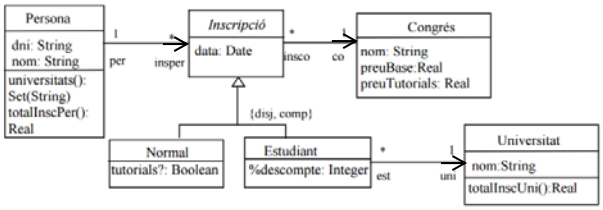

```
public class Persona{
        private String dni
        private String nom
        private Set<Inscripcio> insper

        public double totalInscPer(){
                double total = 0
                for(Inscripcio i in insper){
                        double importInsc = i.getImport()
                        total += importInsc
                }
                return total
        }
}
```

Exercici 3 (Alternativa 2) – El codi



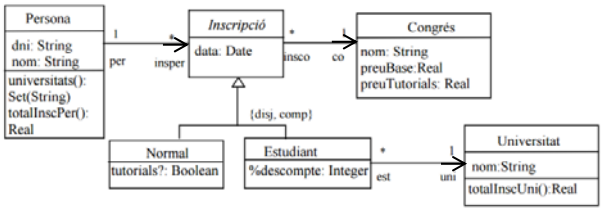

```
public abstract class Inscripcio{
        private Date data
        private Congres co

        public abstract double getImport()

        public double getPreuBase(){
                return co.getPreuBase()
        }

        public double getPreuTutorial(){
                return co.getPreuTutorial()
        }
}
```

Exercici 3 (Alternativa 2) – El codi



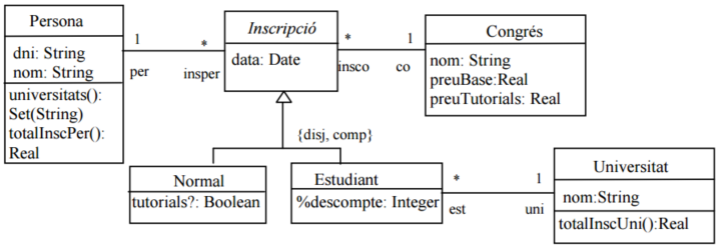

```
public class InscripcioNormal extends Inscripcio {
        private bool tutorials

        public double getImport(){
                double preuBase = this.getPreuBase()
                double import = preuBase
                if(tutorials){
                        double preuTutorials = this.getPreuTutorials()
                        import += preuTutorials
                }
                return import
        }
}
```

Exercici 3 (Alternativa 2) – El codi




```
public class InscripcioEstudiant extends Inscripcio {
        private int descompte
        private Universitat uni

        public double getImport(){
                double preuBase = this.getPreuBase()
                double import = preuBase – (preuBase*%descompte/100)
                return import
        }

}
```

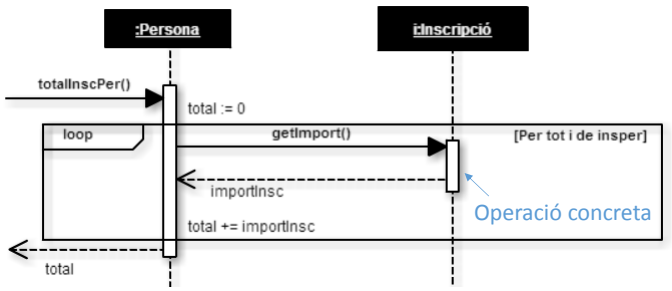
Exercici 3 - Variant



context Persona::totalInscPer():Real
post: result= suma dels imports de totes les inscripcions de la persona. L'import d'una inscripció es calcula com segueix:

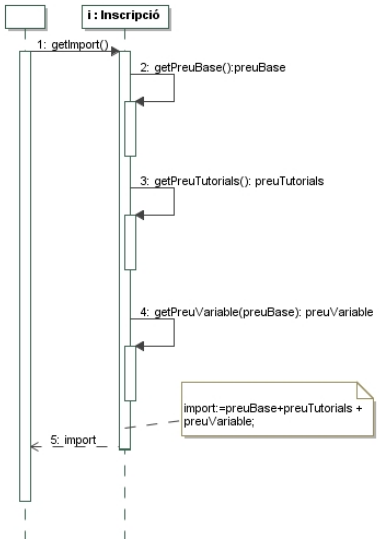
- si la inscripció és de tipus estudiant, l'import és el preu base del congrés més el preu dels tutorials menys el descompte que tingui aplicat al preu base
- si la inscripció és de tipus normal, l'import és el preu base del congrés més el preu dels tutorials menys una reducció del 10% del preu base si tutorials? és fals.

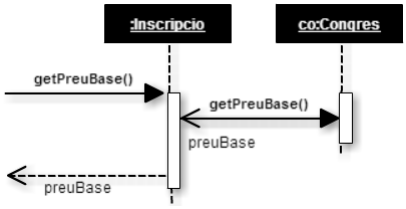
Exercici 3 (Variant) – Diagrama de Seqüència

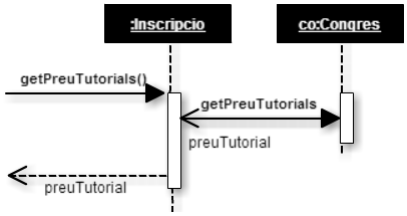


Solució Correcta: Anàlisi
No repetició de codi

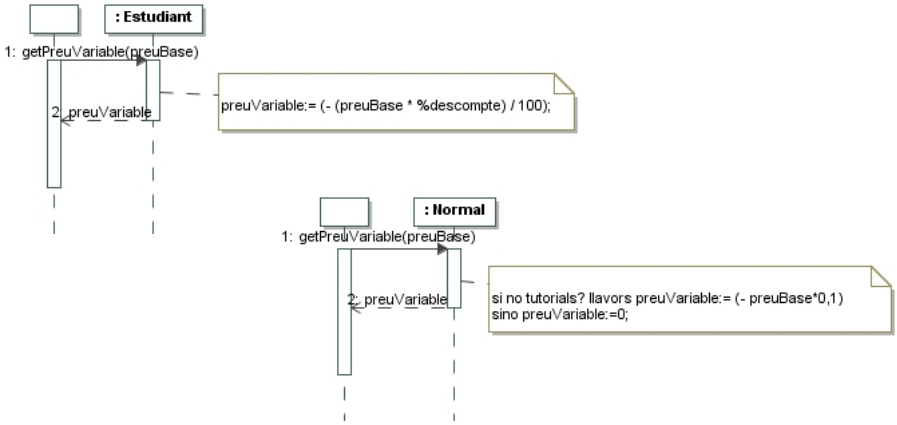
Exercici 3 (Variant) – Diagrama de Seqüència



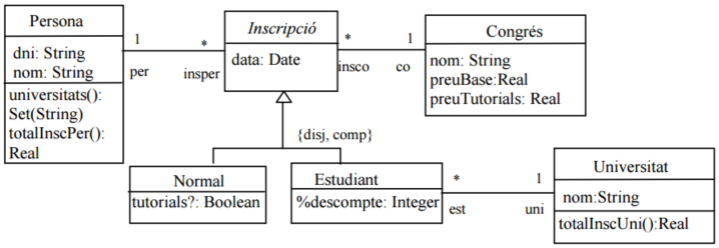




Exercici 3 (Variant) – Diagrama de Seqüència



Exercici 4

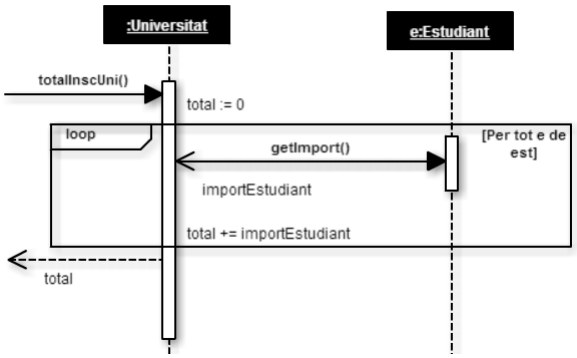


i donat el contracte següent:

context Universitat::totalInscUni():Real

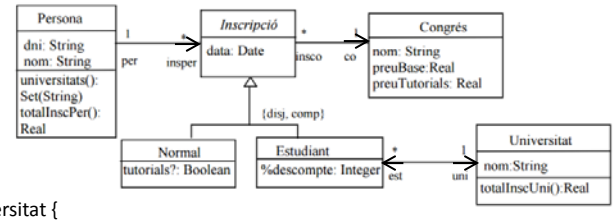
post: result= suma de l'import de totes les inscripcions de tipus estudiant de la universitat.

Exercici 4 – Diagrama de Seqüència



Solució Correcta

Exercici 4 – El codi




```
public class Universitat {
        private String nom
        private Set<Estudiant> est

        public double totalInscUni(){
                double total = 0
                for (Estudiant e in est){
                        double importEstudiant = e.getImport()
                        total += importEstudiant
                }
                return total
        }

}
```