

CFGS Desenvolupament d'aplicacions multiplataforma (DAM) CFGS Desenvolupament d'aplicacions web (DAW)

Mòdul 5 – Entorns de Desenvolupament. UF3 – Introducció al disseny orientat a objectes

EAC1

(Curs 2019–20 / 1r semestre)

Solució

1. (1.5 punts) Preguntes tipus test de **RESPOSTA ÚNICA**.

Cada apartat dels següents respost correctament suma 0,1 punts a la qualificació de l'exercici. Cada apartat respost incorrectament resta 0,03 punts a la qualificació de l'exercici. Cada apartat deixat en blanc no suma ni resta res a la qualificació de l'exercici. En cas que la qualificació sumi un valor negatiu, es considerarà que s'ha obtingut un 0 a l'exercici.

Cal indicar la resposta a totes les preguntes al requadre del final de l'exercici.

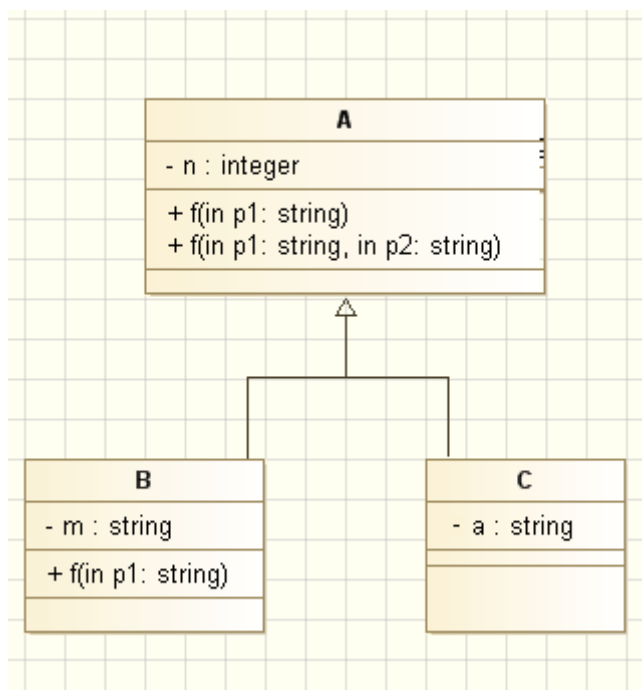



Figura 1

1.1. Tenint en compte la *figura 1*, les dues classes B i C tenen una relació amb la classe A de:

A- Sobreescritura.

B- Herència.

	Codi: I71	Exercici d'avaluació contínua 1	Pàgina 1 de 10
	Versió: 02	DA2_M05_EAC1_Enunciat_1920S1	Lliurament: 18/10/2019

C- Sobrecàrrega.
D- Associació d'agregació.

E- Associació de composició.

1.2. Tenint en compte la *figura 1*, què es dona entre les dues operacions amb un sol paràmetre que hi ha, una a la classe A i una altra a la classe B ?

A- Sobreescritura.
B- Relació d'associació d'agregació.
C- Sobrecàrrega.

D- Relació d'associació de composició.
E- Herència.

1.3. Tenint en compte la *figura 1*, quines classes tenen entre els seus membres l'operació +f(in p1: string, in p2:string) ?

A- Només la classe A .
B- Només la classe B.
C- Només la classe A i la classe B.

D- Només la classe A i la classe C.
E- Les classes A, B i C.

1.4. Tenint en compte la *figura 1*, què es dona entre les dues operacions de la classe A?

A- Sobreescritura.
B- Herència.
C- Relació d'associació d'agregació.

D- Relació d'associació de composició.
E- Sobrecàrrega.

1.5. Quin dels següents és un diagrama de comportament o dinàmic en UML?

A- Diagrama de classes.
B- Diagrama de components.
C- Diagrama d'objectes.

D- Diagrama de seqüència.
E- Diagrama de desplegament.

1.6. Quin dels següents és un diagrama estàtic en UML?

A- Diagrama d'activitats.
B- Diagrama de casos d'ús.
C- Diagrama d'estats.

D- Diagrama d'interacció.
E- Diagrama d'objectes.

1.7. Indiqueu de quin tipus és la relació entre les classes Sèrie i Actor, tenint en compte que volem representar que els actors participen en les sèries. La classe Actor té els atributs codi, nom, cognoms i la classe Sèrie té els atributs codi, títol i númeroDeCapítols:

A- Herència.
B- Agregació.
C- Associació.

D- Composició.
E- Classe associativa.

1.8. Indiqueu de quin tipus és la relació entre les classes Sèrie i Actor, tenint en compte que volem representar que els actors participen en les sèries. La classe Actor té els atributs codi, nom, cognoms, la classe Sèrie té els atributs codi, títol i númeroDeCapítols i la relació té un atribut cadena de caràcters que representa el nom del paper que fa l'actor en la sèrie:


A- Herència.
B- Agregació.
C- Associació.

D- Composició.
E- Classe associativa.

1.9. Indiqueu de quin tipus és la relació *FormaPart* entre les classes ÀlbumMúsica i Cançó.

A- Sobreescritura.

B- Agregació.

	Codi: I71	Exercici d'avaluació contínua 1	Pàgina 2 de 10
	Versió: 02	DA2_M05_EAC1_Enunciat_1920S1	Lliurament: 18/10/19

C- Sobrecàrrega.
D- Composició.

E- Herència.

1.10. Indiqueu de quin tipus és la relació *FormaPart* entre les classes *Pàgina* i *peuDePàgina*:

A- Sobreescritura.
B- Agregació.
C- Sobrecàrrega.

D- Composició.
E- Herència.

1.11. Entre la classe *Espai* (que es refereix a una part d'un pis) i les classes *Cuina*, *Dormitori* i *Bany* hi ha una relació de:

A- Sobreescritura.
B- Agregació.
C- Sobrecàrrega.

D- Composició.
E- Herència.

1.12. Als diagrames de seqüència, el missatge des de l'objecte emissor cap al receptor on aquest no espera la resposta abans de continuar es representa amb:

A- Una fletxa de línia contínua i punta oberta.
B- Una fletxa de línia contínua i punta plena.
C- Una fletxa de línia discontinua i punta oberta.

D- Una fletxa de línia discontinua i punta plena.
E- Una línia.

1.13. Als diagrames de casos d'ús, el mecanisme per separar d'un cas d'ús una o més parts que són opcionals és una:

A- Dependència d'extensió.
B- Dependència d'inclusió.
C- Generalització.

D- Especialització.
E- Associació.

1.14. En un diagrama d'activitats es representa amb un rombe:

A- L'estat inicial.
B- L'estat final.
C- Una decisió.

D- Un fork.
E- Un join.

1.15. En un diagrama d'estats, els estats en general es representen amb:


A- Un cercle ple.
B- Un cercle ple amb un cercle al voltant.
C- Un rombe.

D- Una fletxa amb direcció.
E- Un rectangle amb vores arrodonides.

Respostes:

1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	1.10	1.11	1.12	1.13	1.14	1.15
B	A	E	E	D	E	C	E	B	D	E	A	A	C	E

OBSERVACIÓ IMPORTANT: a la resta d'exercicis cal utilitzar el program **Modelio** per realitzar diagrames UML, generar codi automàticament a partir d'aquest i generar diagrames a partir de codi. Al material de l'aula hi podeu trobar un document sobre la instal·lació i el funcionament bàsic d'aquest programa. En aquest document s'inclouen

	Codi: I71	Exercici d'avaluació contínua 1	Pàgina 3 de 10
	Versió: 02	DA2_M05_EAC1_Enunciat_1920S1	Lliurament: 18/10/19

instruccions per aconseguir el fitxer gràfic amb el diagrama que heu fet. Aquests fitxers gràfics resultants els annexareu al fitxer comprimit que heu de lliurar i els donareu el nom que s'indica a cada exercici.

2. (1,5 punts) Diagrama de classes.

Representeu, utilitzant el programari **Modelio**, un diagrama de classes amb les que s'indiquen a continuació. Assegureu-vos que la cardinalitat i la visibilitat dels atributs i operacions és correcta. No heu de representar ni els *setters* ni els *getters*.

Es vol gestionar informació sobre una companyia aèria. Les classes es descriuen a continuació; podeu triar el tipus primitiu UML que considereu més adient per cada dada quan aquest no s'especifiqui:

Classe Aeroport

Conté els següents atributs:

- codi: cadena de caràcters amb el codi assignat a l'aeroport.
- ciutat: cadena de caràcter que conté la ciutat on es troba l'aeroport.

- Classe Vol

Conté els següents atributs:

- codi: cadena de caràcters amb el codi assignat al vol
- horaSortida: atribut de tipus data que conté l'hora de sortida del vol
- Durada: enter que emmagatzema els minuts que triga el vol en anar de l'origen al destí.

i la següent operació:

- obtenirHora(): retorna una cadena de caràcters que conté l'hora d'arribada del vol amb el format «hora:minuts».

Un vol surt d'un determinat aeroport i hi arriba a un altre. D'un aeroport en poden sortir i arribar molts vols.

- Classe TipusAvio

Conté els següents atributs:

- codi: enter amb el codi del tipus d'avió
- capacitat: enter que emmagatzema el número de passatgers d'aquest tipus d'avió

Un vol serà operat per un determinat tipus d'avió i un tipus d'avió pot servir per operar diversos vols.

- Classe PassatgerHabitual:

Conté els següents atributs:

- nif: és una cadena de caràcters que conté el nif del passatger
- nom: és una cadena de caràcters que guarda el nom i els cognoms del passatger
- email: cadena de caràcters amb el correu electrònic del passatger habitual


i la següent operació:

- numeroViatgesRegistrat(): retorna un enter amb la quantitat de viatges en què el passatger es troba enregistrat com habitual.

Els passatgers habituals viatgen freqüentment en el mateixos vols. Un vol podrà tenir molts passatgers habituals i un passatger habitual podrà fer ús de diversos vols. Aquests tipus de passatgers tindran determinats descomptes segons el vol i el número de cops que l'han utilitzat. És per això que es vol representar el número de cops que un passatger ha anat en un determinat vol i també a partir de quants cops se li farà un descompte al passatger (no tots els passatgers obtenen descomptes amb el mateix número de viatges realitzats).

Tingueu present que, a més de les classes que s'esmenten explícitament, cal definir també els diferents tipus de relacions i que pot caldre definir també alguna classe més.

El resultat serà un fitxer gràfic generat a partir del diagrama de classes que heu creat i que s'anomenarà **Exercici2**.

	Codi: I71	Exercici d'avaluació contínua 1	Pàgina 4 de 10
	Versió: 02	DA2_M05_EAC1_Enunciat_1920S1	Lliurament: 18/10/19

3. (2 punts) Diagrama de classes (II), generació automàtica de codi i enginyeria inversa.

3.1. (1,25 punts) Representeu en un diagrama de classes les que s'indiquen a continuació amb el programari **Modelio**. Assegureu-vos que la cardinalitat i la visibilitat dels atributs i operacions és correcta. No heu de representar ni els setters ni els getters.

Les classes es descriuen a continuació i corresponen a la gestió d'una empresa. Podeu triar per a cada dada el tipus primitiu UML que considereu més adient quan aquest no s'especifiqui:

a) Classe Banc

Conté les següents dades:

- codi: enter que emmagatzema el codi del banc

b) Classe Client

Conté les següents dades:

- codi: enter amb el codi intern del client
- nom: cadena de caràcters que conté el nom i els cognoms del client

Un banc tindrà molts clients i un client podrà operar amb diversos bancs.

c) Classe CompteBancari

Conté les següents dades:

- codi: cadena de caràcters amb el codi del compte bancari
- Saldo: real que conté la quantitat de diners que conté el compte

Els comptes bancaris pertanyen a un únic client. Un client pot tenir molts comptes bancaris.

Els comptes bancaris poden ser de dos tipus: comptes corrents o comptes d'estalvi. Dels comptes corrents en volem representar la quantitat màxima de diners que es pot extreure en un moviment o operació. Dels comptes d'estalvi serà necessari representar l'interès que es pagarà pels diners dipositats.

Tingueu present que, a més de les classes que s'esmenten explícitament, cal definir també els diferents tipus de relacions i que pot caldre definir també alguna classe més.


El resultat serà un fitxer gràfic generat a partir del diagrama de classes que heu creat i que s'anomenarà: **Exercici3-1**.

3.2. (0,75 punts) Genereu el codi de les classes anteriors. Heu de tenir en compte abans de fer la generació que les relacions que hàgiu posat en l'apartat anterior entre les classes Banc i la classe Client hauran de ser navegables des de la classe Banc cap a la classe Client (a la línia hi haurà una fletxa al costat de la classe Client). Això pot fer-se des de les propietats de l'associació. També serà necessari que entre la classe Client i la classe CompteBancari es pugui navegar en totes dues direccions.

Accediu a editar les classes Java generades, copieu-les una a una al porta-retalls i enganxeu-les a continuació (el resultat és textual i conté el codi d'aquestes classes):

```
import java.util.ArrayList;
import java.util.List;
import com.modeliosoft.modelio.javadesigner.annotations.objid;

@objid ("376927c6-097c-44d0-a4de-6fa838ff7a7d")
public class Banc {
```

	Codi: I71	Exercici d'avaluació contínua 1	Pàgina 5 de 10
	Versió: 02	DA2_M05_EAC1_Enunciat_1920S1	Lliurament: 18/10/19

```

@objid ("6e04fb50-5277-4031-9cc1-21ae7113284b")
private int codi;

@objid ("27c9955e-904f-4c6b-b0f0-ef7e16e067c7")
public List<Client> client = new ArrayList<Client> ();
}

import java.util.ArrayList;
import java.util.List;

import com.modeliosoft.modelio.javadesigner.annotations.objid;

@objid ("86ba3152-77e9-40e3-a67c-c6aa723f24ab")
public class Client {
    @objid ("36f7df6d-6aa2-43c7-931e-c9b88e4b0e04")
    private int codi;

    @objid ("619c542a-ccd5-4ed2-a43a-fb2ed6575cbb")
    private String nom;

    @objid ("230c7ea1-c8db-4493-aac0-57ff05d4b09d")
    public List<CompteBancari> comptes = new ArrayList<CompteBancari> ();
}

import java.util.ArrayList;
import java.util.List;
import com.modeliosoft.modelio.javadesigner.annotations.objid;

@objid ("e4ffe6fe-fba6-4975-8210-cb1b2b6509ba")
public class CompteBancari {
    @objid ("024a8d74-6606-40f3-8377-adde1ad8e0c5")
    private String codi;


    @objid ("ed4e8ec3-3c80-46d6-93c7-5912592e5773")
    private double saldo;

    @objid ("60b53988-f2b2-4ffd-a231-29997a14891e")
    public Client client;
}

import com.modeliosoft.modelio.javadesigner.annotations.objid;

@objid ("e2bd4cfc-6990-4f83-b763-1b4dae8194e6")
public class CompteCorrent extends CompteBancari {
    @objid ("2f3d7e16-bbce-4418-99a5-cad16379b077")
    private double quantitatMaxima;

```

	Codi: I71	Exercici d'avaluació contínua 1	Pàgina 6 de 10
	Versió: 02	DA2_M05_EAC1_Enunciat_1920S1	Lliurament: 18/10/19

```

}

import com.modeliosoft.modelo.javadesigner.annotations.objid;

@objid ("83e5c9ad-f2b6-4172-b084-249a0104e3ce")
public class CompteEstalvi extends CompteBancari {
    @objid ("a60e9ef3-1e3d-46ce-b834-2f60b3e2f416")
    private double interes;

}

```

Importeu al diagrama anterior el codi indicat a continuació.

```

import java.util.Date;

public class Moviment
{
    private int numero;
    private String descripcio;
    private float quantitat;
    private Date data;

    public void setNumero(int numero){
        this.numero=numero;
    }

    public int getNumero(){
        return this.numero;
    }

    public void setDescripcio(String descripcio){
        this.descripcio=descripcio;
    }

    public String getDescripcio(){
        return this.descripcio;
    }

    public void setQuantitat(float quantitat){
        this.quantitat=quantitat;
    }

    public float getQuantitat(){
        return this.quantitat;
    }

    public void setData(Date data){
        this.data=data;
    }
}

```

```

    }

    public Date getData() {
        return this.data;
    }
}

```

Després, afegiu-hi una associació entre CompteBancari i Moviment. Tingueu en compte que aquesta associació representa els moviments que es fan en els comptes bancaris (extreure diners del compte, fer un pagament, ingressar diners,...) i que, per tant, en un compte s'hi podran fer molts moviments i que un moviment serà realitzat per a un únic compte bancari.

Aquesta relació haurà de ser navegable des Compte cap a Moviment (a la línia hi haurà una fletxa al costat de Moviment). Això pot fer-se des de les propietats de l'associació. Genereu el gràfic del diagrama de classes resultant i graveu-lo al fitxer **Exercici3-2**, que caldrà lliurar.

Genereu novament el codi corresponent a la classe CompteBancari, accediu a editar-lo, copieu-lo al porta-retalls i enganxeu-lo a continuació (el resultat és textual i conté el codi d'aquesta classe):

```

import java.util.ArrayList;
import java.util.List;
import com.modeliosoft.modelio.javadesigner.annotations.objid;

@objid ("e4ffe6fe-fba6-4975-8210-cb1b2b6509ba")
public class CompteBancari {
    @objid ("024a8d74-6606-40f3-8377-adde1ad8e0c5")
    private String codi;

    @objid ("ed4e8ec3-3c80-46d6-93c7-5912592e5773")
    private double saldo;

    @objid ("60b53988-f2b2-4ffd-a231-29997a14891e")
    public Client client;

    @objid ("487fe04a-3b51-4fb6-a6d1-2b9af42b900a")
    public List<Moviment> moviment = new ArrayList<Moviment> ();
}

```


4. (1,5 punts) Diagrama de casos d'ús.

Una aplicació de borsa es connecta periòdicament (cada 30 segons, per exemple) a un proveïdor per oferir informació de les cotitzacions de diferents valors.

L'aplicació permet als usuaris:

- consultar els diferents valors
- marcar una sèrie de valors per tal de poder fer-ne un seguiment.

Si a l'hora d'actualitzar les cotitzacions marcades es produeix una gran variació (una pujada o una baixada molt alta), s'informarà d'aquest fet a l'usuari.

Podeu realitzar suposicions o precisions raonables sobre l'enunciat, sempre i quan les detalleu juntament amb la resposta.

El resultat serà un fitxer gràfic generat a partir del diagrama de casos d'ús que heu creat i que s'anomenarà: **Exercici4**.

5. (2 punts) Diagrama de seqüència.


Representeu en un diagrama de seqüència les interaccions que es produeixen quan es calcula el retard d'un vol determinat:

1. Un Usuari pregunta a un Controlador Aeri quin és el retard d'un determinat vol.
2. El Controlador pregunta a aquest Vol determinat quina era l'hora de sortida i, també, quina era la durada prevista per al viatge.
3. El Controlador pregunta a un Servidor d'horari quina és l'hora actual.
4. Amb les dades anteriors, el Controlador pot calcular el retard i respondre a l'usuari.

Podeu realitzar suposicions o precisions raonables sobre l'enunciat, sempre i quan les detalleu juntament amb la resposta.

Malgrat que tots dos tenen una temàtica similar, aquest exercici és independent de l'exercici 2.

El resultat serà un fitxer gràfic generat a partir del diagrama de seqüència que heu creat i que s'anomenarà: **Exercici5**.

	Codi: I71	Exercici d'avaluació contínua 1	Pàgina 9 de 10
	Versió: 02	DA2_M05_EAC1_Enunciat_1920S1	Lliurament: 18/10/19


6. (1.5 punts) Diagrama d'activitats.

Representeu en un diagrama d'activitats el següent procés de facturació o checkin d'un vol:

1. El passatger presenta la reserva del vol juntament amb el seu DNI. Si es detecta que no és correcta la reserva (perquè el vol s'ha cancel·lat o per altres causes) s'envia al passatger a l'oficina d'atenció al client.
2. En cas contrari, es pregunta al passatger per les seves preferències (seient amb finestra o al costat del passadís, ...).
3. Si el passatger porta equipatge per facturar, es pesa i es posa sobre la cinta transportadora i es lliura el resguard de l'equipatge factura.
4. El sistema assigna seient al passatger. Aquesta activitat es podrà fer en paral·lel amb l'anterior.
5. Es lliuren el bitllets d'embarcament al passatger; en ells s'indiquen el seient assignat i la porta d'embarcament.

Podeu realitzar suposicions o precisions raonables sobre l'enunciat, sempre i quan les detalleu juntament amb la resposta.

El resultat serà un fitxer gràfic que contindrà el diagrama resultant i s'anomenarà **Exercici6**.

	Codi: I71	Exercici d'avaluació contínua 1	Pàgina 10 de 10
	Versió: 02	DA2_M05_EAC1_Enunciat_1920S1	Lliurament: 18/10/19