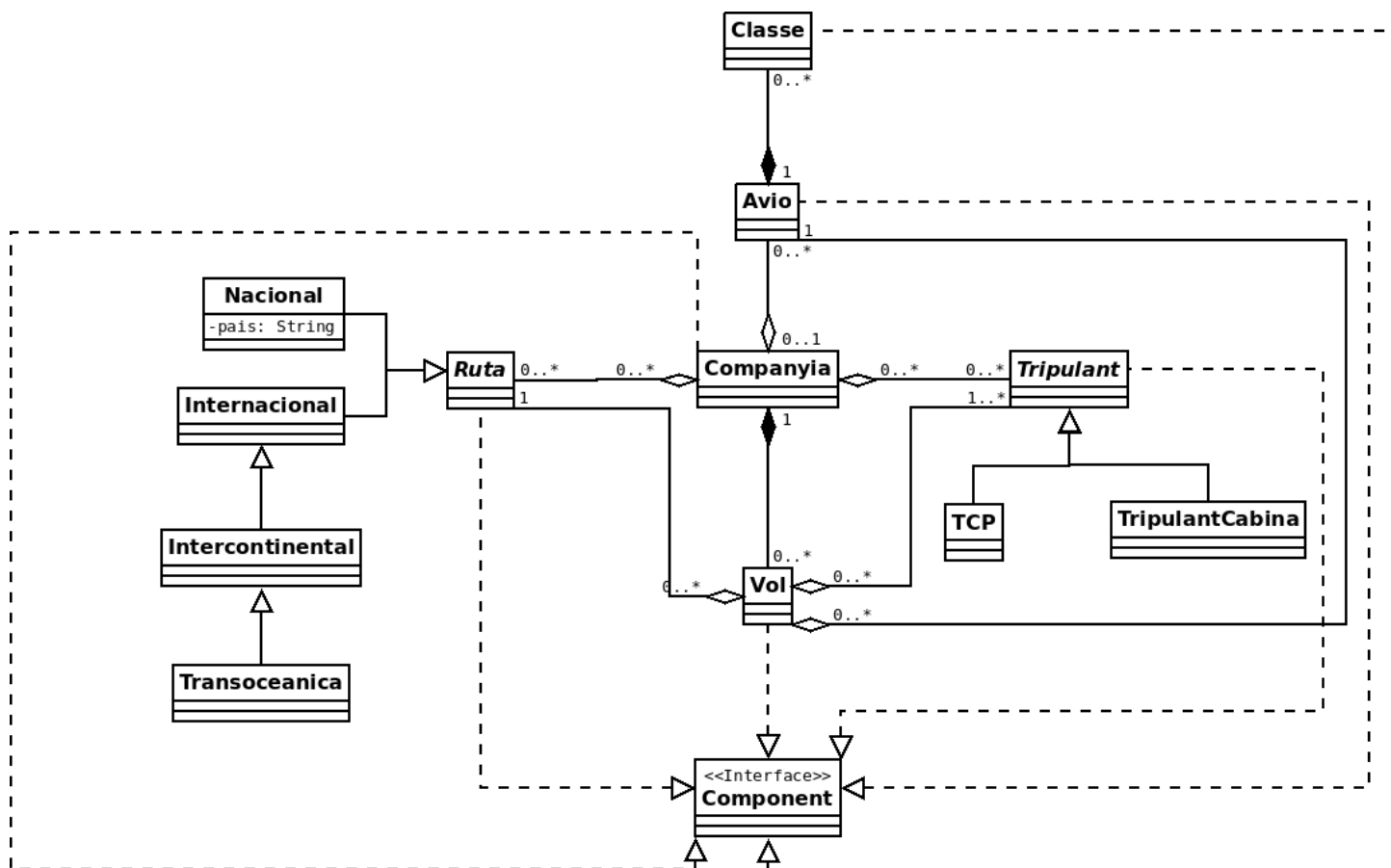


## PRÀCTICA V2 UF4: CARACTERÍSTIQUES AVANÇADES DE LA POO

En aquesta pràctica continuem el desenvolupament de l'aplicació de gestió de vols. El punt de partida és el projecte NetBeans solució de la pràctica anterior, que en el Moodle el trobareu com a "**GestioVolsV2**".

L'objectiu és canviar el projecte incorporant l'herència i interfaces. Aquí teniu el diagrama de classe del projecte corresponent a aquesta segona pràctica:



Heu d'implementar al projecte el que es demana a la graella següent.

Classe o Interfície	Requeriments detallats
Component	<p>És una interface. Com que totes les classes de l'aplicació, menys la classe Aplicacio tenen uns mètodes i una constant en comú, hem decidit crear aquesta interfície per poder agrupar-los.</p> <p>Constant i mètodes:</p> <pre>Scanner DADES= new Scanner(System.in);  void mostrarComponent();  public void modificarComponent() throws ParseException;  Object demanarDades(String peticio, int tipus);</pre> <p>L'últim mètode demanarDades, és un mètode igual per totes les classes que han d'implementar la interfície, per tant l'heu de desenvolupar en la interface. Què ha de fer aquest mètode? Ha de demanar les dades a l'usuari i llegir les dades introduïdes per l'usuari mitjançant el mètode pertinent de la classe Scanner, de la següent manera:</p> <ul style="list-style-type: none"><li>• Demanar a l'usuari les dades que ha d'introduir mitjançant el primer paràmetre peticio que li passem al mètode.</li><li>• Llegirà les dades introduïdes per l'usuari amb el mètode escaient de la classe Scanner. Mitjançant el segon paràmetre tipus, sabrem quin és el mètode de la classe Scanner hem de fer servir, és a dir, si tipus és 1, l'usuari introduirà un enter, si és 2 un token (cadena de caràcters sense espais en blanc), si és 3 un double i si és 4 una cadena de caràcters.</li><li>• L'objecte de la classe Scanner que heu d'utilitzar per cridar als seus mètodes, és la constant DADES definida en aquesta interfície.</li><li>• Retornarà les dades llegides, per això el retorn és un Object, ja que el retorn pot ser de diferents tipus de dades, és a dir, un enter, o un token, o un double o una cadena de caràcters.</li></ul> <p>Aquest mètode substituirà a totes les peticions i lectura de dades</p>

	introduïdes per l'usuari, dels mètodes no estàtics que implementin aquesta interface.
Classe i Avio	<p>Ara implementaran la interface Component, així doncs heu de canviar el que calgui. Els mètodes per mostrarClasse o mostrarAvio, passaran a ser el mètode de la interface mostrarComponent. Aquest mètode ha de continuar fent el mateix que feien mostrarClasse i mostrarAvio. El mateix passa amb els mètodes modificarClasse o modificarAvio, ara passaran a ser modificarComponent de la interface i ha de continuar fent el mateix que fan els altres dos mètodes.</p> <p>Recordeu, que com s'ha especificat abans, s'han de substituir la petició i la lectura de dades introduïdes per l'usuari en els diferents mètodes no estàtics de les dues classes, pel mètode demanarDades de la interface.</p>
Ruta	<p>Classe abstracte que implementa la interface Component</p> <p><b>Atributs</b>  codi, aeroportDes, aeroportOri i distancia (són els atributs comuns a les classes de les diferents rutes. Els tipus d'aquests atributs, són els mateixos que els d'aquestes classes.)</p> <p><b>Constructors</b>  Un constructor que farà el mateix que el constructor de la classe RutaNacional però sense l'atribut pais.</p> <p><b>Mètodes</b></p> <ul style="list-style-type: none"> <li>• accessors</li> <li>• La implementació dels mètodes de la interface modificarComponent i mostrarComponent. El primer modificarà els atributs de la classe, tal com fa, per exemple, modificarRutaNacional de la classe RutaNacional. El segon mostrarà el valor dels atributs de la classe, tal com fa, per exemple, mostrarRutaNacional de la classe RutaNacional.</li> </ul> <p>Recordeu, que com s'ha especificat abans, s'han de substituir la petició i la lectura de dades introduïdes per l'usuari en els diferents mètodes no estàtics, pel mètode demanarDades de la interface.</p>
RutaNacional i RutaInternacional.	<p>Hereten de Ruta.</p> <p>Heu de treure / canviar el que calgui</p> <p>Notes: En heretar de Ruta hi haurà atributs que no necessiteu tornar a declarar, i mètodes que canviaran de nom i/o contingut o desapareixeran. També heu de tenir en compte que potser aquestes classes tenen més atributs que Ruta.</p>
RutaIntercontinental	Hereta de RutaInternacional.

	<p>Heu de treure / canviar el que calgui</p> <p>Notes: En heretar de RutaInternacional hi haurà atributs que no necessiteu tornar a declarar, i mètodes que canviaran de nom i/o contingut o desapareixeran. També heu de tenir en compte que potser aquesta classe té més atributs que RutaInternacional.</p>
RutaTransoceanica	<p>Hereta de RutaIntercontinental.</p> <p>Heu de treure / canviar el que calgui</p> <p>Notes: En heretar de RutaIntercontinental hi haurà atributs que no necessiteu tornar a declarar, i mètodes que canviaran de nom i/o contingut o desapareixeran. També heu de tenir en compte que potser aquesta classe té més atributs que RutaTransoceanica.</p>
Tripulant	<p>Classe abstracte que implementa la interface Component</p> <p><b>Atributs</b></p> <p>passaport, nom, edat, dataAlta, horesVol i rang (són els atributs comuns a les classes TripulantCabina i TCP. Els tipus d'aquests atributs, són els mateixos que els d'aquestes classes.)</p> <p><b>Constructors</b></p> <p>Un constructor que farà el mateix que el constructor de la classe TCP.</p> <p><b>Mètodes</b></p> <ul style="list-style-type: none"> <li>accessors</li> <li>La implementació dels mètodes de la interface modificarComponent i mostrarComponent. El primer modificarà els atributs de la classe, tal com fa modificarTCP de la classe TCP. El segon mostrarà el valor dels atributs de la classe, tal com fa mostrarTCP de la classe TCP.</li> </ul>
TripulantCabina TCP	<p>Hereta de Tripulant.</p> <p>Heu de treure / canviar el que calgui</p> <p>Notes: En heretar de Tripulant hi haurà atributs que no necessiteu tornar a declarar, i mètodes que canviaran de nom i/o contingut o desapareixeran. També heu de tenir en compte que potser aquestes classes tenen més atributs que Tripulant.</p>
Vol	<p>Ara implementarà la interface Component, així doncs heu de canviar el que calgui.</p> <p>Es requereix el següent:</p> <ul style="list-style-type: none"> <li>L'atribut tipusRuta l'eliminem, ja que ara ja no ens farà falta.</li> <li>Heu d'afegir un nou atribut cap de tipus TCP, on guardarem el TCP del vol que farà de cap de la cabina de passatgers del vol. En el mètode constructor s'ha d'inicialitzar a null, ja</li> </ul>

	<p>que quan es crea un vol, encara no s'ha assignat un cap de cabina de passatgers.</p> <ul style="list-style-type: none"> <li>Els vectors tripulacioCabina i TCPs que teniu com a atributs de Vol s'han d'eliminar i substituir-los per un únic vector de tripulants del vol. Amb els atributs posicioXXX heu de fer el mateix. Així, la declaració del vector i el comptador ha de ser:</li> </ul> <pre>private Tripulant[] tripulacio; private int posicioTripulacio;</pre> <p>En el mètode constructor, el vector tripulacio s'ha d'inicialitzar buit i amb una capacitat per 32 tripulants, i l'atribut posicioTripulacio amb la primera posició del vector.</p> <ul style="list-style-type: none"> <li>El mètode modificarVol, passarà a ser el mètode de la interface modificarComponent. Aquest mètode ha de continuar fent el mateix que feia el mètode modificarVol.</li> <li>El mètode mostrarVol, passarà a ser el mètode de la interface mostrarComponent. Aquest mètode ha de continuar fent el mateix que feia el mètode mostrarVol, tenint en compte que s'ha d'adaptar a les modificacions fetes en les classes dels objectes que s'han de mostrar.</li> <li>Els mètodes afegirTripulantCabina i afegirTCP, ara s'han de substituir per un únic mètode afegirTripulant(Tripulant tripulant), que ha d'afegir el tripulant passat per paràmetre al vector tripulacio. Si el tripulant passat per paràmetre és de tipus TCP, li hem de demanar a l'usuari si vol que sigui cap de cabina de passatgers o no. Si la resposta és si, modificarem l'atribut cap amb el nou tripulant afegit i modificarem el rang del nou cap a "cap" i el del cap anterior a null.</li> <li>Recordeu, que com s'ha especificat abans, s'han de substituir la petició i la lectura de dades introduïdes per l'usuari en els diferents mètodes pel mètode demanarDades de la interface.</li> <li>Heu de fer els canvis necessaris en els diferents mètodes per ajustar-los als nous mètodes i atributs.</li> </ul>
Companyia	Ara implementarà la interface Component, així doncs heu de canviar el que calgui.

Document:	Data: 03/12/18
Pràctica V1 UF4	Pàgina 5 de 9

Es requereix el següent:

- Els atributs vectors s'han d'eliminar i substituir-los per un únic vector de components de la companyia. Amb els atributs posicioXXX heu de fer el mateix. Així, la declaració del vector i el comptador ha de ser:

```
private Component[] components;  
private int posicioComponents;
```

En el mètode constructor, el vector components s'ha d'inicialitzar buit i amb una capacitat per 2945 components, i l'atribut posicioComponents amb la primera posició del vector.

- El mètode modificarCompanyia, passarà a ser el mètode de la interface modificarComponent. Aquest mètode ha de continuar fent el mateix que feia el mètode modificarCompanyia.
- El mètode mostrarCompanyia, passarà a ser el mètode de la interface mostrarComponent. Aquest mètode ha de continuar fent el mateix que feia el mètode mostrarCompanyia.
- Heu de crear un nou mètode public int seleccionarComponent(int tipus, String id) què substituirà i eliminarà tots els mètodes seleccionar de la classe Companyia. Per desenvolupar-lo podeu partir del codi de seleccionarAvio(). Heu de fer el mateix que a seleccionarAlvio(), però ara heu de saber quin tipus de component es vol seleccionar (Avio, Ruta, Tripulant o Vol) i per això es passa per paràmetre el tipusComponent que definim com 1=Avio, 2=Ruta, 3=Tripulant i 4=Vol.

Com a segon paràmetre li passarem l'identificador (codi o passaport) del component a seleccionar. Aquest paràmetre pot ser null. Si és null, li demanarem a l'usuari quin és el codi o passaport (segons el tipus de component) del component a cercar.

Un cop se sap el tipus i l'identificador, llavors la selecció es fa com a seleccionarAvio però tenint en compte què tots els components estan en un mateix vector, per tant, hem de saber de quin tipus de component es tracta quan accedim a un component del vector i comprovar si és del tipus passat per paràmetre, per després poder comprovar si el component és el que volem seleccionar.



	<ul style="list-style-type: none"> <li>Els mètodes <code>afegirXXX</code> per afegir els diferents components a una companyia, s'han de modificar per comprovar si existeixen en la companyia abans d'afegir-los. Si no existeixen, els afegirem, i si existeixen, no els afegirem i li mostrarem a l'usuari el missatge "\nAquest XXX ja existeix", on XXX serà el tipus de component escaient.</li> <li>Els mètodes <code>afegirTripulantCabinaVol</code> i <code>afegirTCPVol</code>, s'han d'eliminar i substituir pel mètode <code>afegirTripulantVol(int tipus)</code> que farà el mateix que els dos mètodes anteriors però tenint en compte quin tipus de tripulant s'ha d'afegir segons el tipus passat per paràmetre que definim com 6=TripulantCabina i 7=TCP.</li> <li>Els mètodes <code>afegirRutaVol(int tipus)</code> s'ha de modificar perquè fagi el mateix que el mètode <code>afegirTripulantVol(int tipus)</code> però amb els diferents tipus de rutes tenint en compte quin tipus de ruta s'ha d'afegir segons el tipus passat per paràmetre que definim com 2=RutaNacional, 3=RutaInternacional, 4=RutaIntercontinental i 5=RutaTransoceanica.</li> <li>Recordeu, que com s'ha especificat abans, s'han de substituir la petició i la lectura de dades introduïdes per l'usuari en els diferents mètodes pel mètode <code>demanarDades</code> de la interfície.</li> <li>Heu de fer els canvis necessaris en els diferents mètodes per ajustar-los als nous mètodes i atributs.</li> </ul>
Aplicacio	<p>Heu d'aconseguir només tenir el <code>menuPrincipal</code>, el <code>menuCompanyies</code> i un nou <code>menuComponents</code> (substituint i eliminant els d'avions, rutes, tripulants i vols).</p> <p>En resum:</p> <p>Els requeriments individuals són els següents:</p> <ul style="list-style-type: none"> <li>Heu d'adaptar els mètodes de la classe, a les novetats i modificacions de les altres classes del projecte.</li> <li><b>MenuPrincipal:</b> <ul style="list-style-type: none"> <li>L'opció 1, continuarà cridant al <code>menuCompanyies</code>.</li> <li>Les opcions de la 2 a la 9 cridaran al <code>menuComponents</code>.</li> </ul> </li> <li><code>menuComponents(int tipus)</code> → partint dels menús d'avions, rutes, tripulants i vols, heu d'unificar-los per crear el nou</li> </ul>

menú. Penseu que hi haurà accions que seran comuns als menús i que ara passaran a ser només una per tots, i altres que no seran de tots els menús, en aquests casos heu de tenir en compte que segons el tipus de component passat per paràmetre, se'ls ha de mostrar en el menú unes opcions o altres per poder seleccionar-les i fer el que calgui. Els valors que pot prendre el paràmetre tipus són: 1=Avio 2=RutaNacional, 3=RutaInternacional, 4=RutaIntercontinental, 5=RutaTransoceanica, 6=TripulantCabina, 7=TCP i 8=Vol.

#### Consideracions importants:

- Després d'haver practica la classe LocalTime en la primera practica, s'ha canviat el tipus de l'atribut hores de vol de les classes TripulantCabina i TCP, per un enter en lloc d'un LocalTime, d'aquesta manera les hores de vol no es limitaran a 24.
- En general, i sobretot quan als requeriments es diu substituir i eliminar atributs o mètodes, és molt aconsellable partir d'un dels que s'han d'eliminar i canviar-lo
- L'ordre d'implementació és important, és a dir, és millor seguir l'ordre de canvis de la taula de requeriments (podeu treballar en paral·lel les classes Classe, Avió, rutes i tripulants)
- Continuem sense controlar els errors, de tipus l'usuari introdueix una lletra quan es demana una opció numèrica.
- Heu d'afegir a les classes les importacions que calguin.

#### La valoració d'aquesta pràctica es basarà en els criteris següents:

- Correcte desenvolupament de les següents classes:
  - **Component** → **1 punt**
  - **Classe** → **0,25 punts**
  - **Avio** → **0,25 punts**
  - **Ruta** → **0,75 punt**
  - **RutaNacional** → **0,25 punt**
  - **RutaInternacional** → **0,25 punts**
  - **RutaIntercontinental** → **0,25 punts**
  - **RutaTransoceanica** → **0,25 punts**
  - **Tripulant** → **0,75 punt**
  - **TripulantCabina** → **0,25 punt**
  - **TCP** → **0,25 punt**

Document:	Data: 03/12/18
Pràctica V1 UF4	Pàgina 8 de 9



- **Vol → 1 punts**
- **Companyia → 1,5 punts**
- **Aplicació:**
  - **menuPrincipal → 0,75 punts**
  - **menuCompanyia → 0,25**
  - **menuComponents → 2 punts**

La pràctica l'heu de fer en grups de 2 persones.

Teniu temps per lliurar la pràctica fins **dimecres dia 12 de desembre a les 20:00** en l'enllaç que trobareu en el curs del Moodle.

Es corregirà a partir del dia **12/12/2018** de manera presencial en les hores de classe.

<b>Document:</b>	<b>Data:</b> 03/12/18
Pràctica V1 UF4	<b>Pàgina</b> 9 <b>de</b> 9