Pràctica 1

L'Estació d'Esquí Vall2000

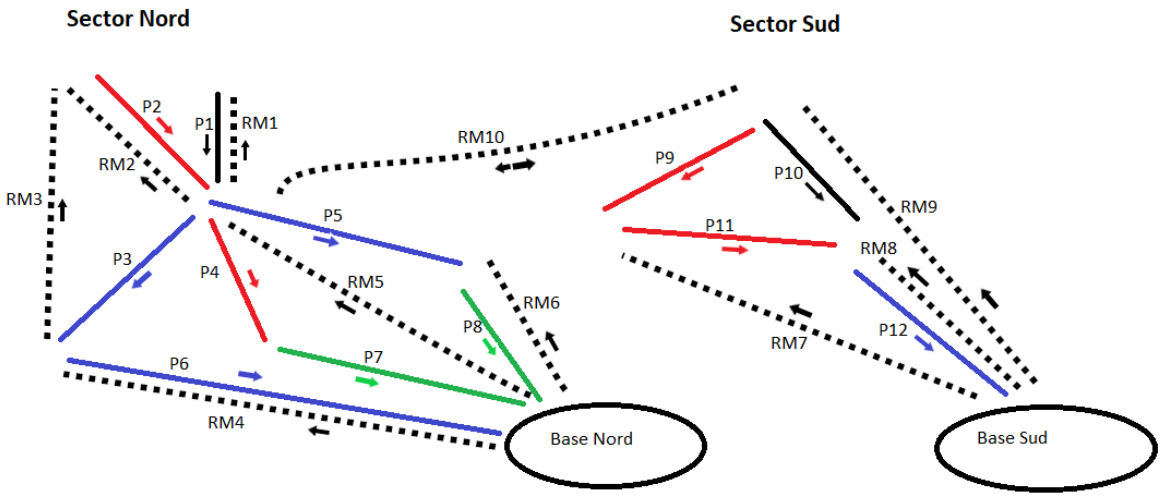

Figura 1. Mapa de l'estació de esquí Vall2000. Hi ha dos sectors: el sector nord, accessible des de la base nord; i el sector Sud, accessible des de la base Sud. Les pistes estan representades amb una línia contínua, que pot ser de color verda, blava, vermella o negra. Els remuntadors mecànics estan representats per una línia negra discontínua.

1. Descripció del Problema

Es vol desenvolupar un sistema que permeti modelar, consultar i canviar la informació sobre l'estat d'una estació d'esquí anomenada **Vall2000**.

Les Pistes. L'estació en qüestió es composa de 12 pistes d'esquí distribuïdes en 2 sectors: el sector Nord, amb 8 pistes, i el sector Sud, més petit, amb només 4 pistes. Les pistes es caracteritzen per un nom, una longitud (en Kilòmetres), un nivell de dificultat, un estat de la pista (que pot set Oberta o Tancada), i un estat de la neu (que pot ser Pols, Primavera o Dura). El nivell de dificultat de la pista es representa mitjançant una color, que pot set Verd (molt fàcil), Blava (fàcil), Vermella (difícil) o Negra (molt difícil). Els detalls de cada pista, així com l'estat actual de les mateixes, es pot trobar a la Taula 1.

Nom	Sector	Nivell de dificultat	Longitud	Estat de la Neu	Estat de la Pista	Accés Amb
P1	Nord	Negra	0.8 Km	Dura	Oberta	RM1
P2	Nord	Vermella	1.1 Km	Dura	Oberta	RM2; RM3
P3	Nord	Blava	1.8 Km	Pols	Oberta	RM5
P4	Nord	Vermella	1.2 Km	Dura	Oberta	RM5
P5	Nord	Blava	2.1 Km	Primavera	Oberta	RM5
P6	Nord	Blava	2.9 Km	Pols	Oberta	RM4
P7	Nord	Verda	1.2 Km	Primavera	Oberta	RM5
P8	Nord	Verda	0.9 Km	Pols	Oberta	RM5; RM6
P9	Sud	Vermella	2.1 Km	Dura	Tancada	RM9
P10	Sud	Negra	0.6 Km	Primavera	Tancada	RM9
P11	Sud	Vermella	1.3 Km	Primavera	Oberta	RM7; RM9
P12	Sud	Blava	1.9 Km	Pols	Oberta	RM8

Taula 1: Informació detallada sobre les pistes de l'estació Vall2000.

Els Remuntadors Mecànics. D'altra banda, l'estació conté un conjunt de remuntadors mecànics que permeten als esquiadors desplaçar-se per l'estació. A la Vall2000 hi ha 5 tipus de remuntadors mecànics (RM): cintes transportadores, teleesquí, telecadires, telecabines i telefèrics. Els remuntadors tenen un estat, que pot ser *en servei* o *fora de servei*. En el sector Nord de la Vall200 hi ha 6 remuntadors: una cinta transportadora (RM6), dos teleesquís (RM2 i RM4), dues telecadires (RM1 i RM3) i una telecabina (RM5). En el sector Sud, hi ha només 3 remuntadors: un teleesquí (RM7), una telecadira (RM8) i una telecabina (RM9). A més a més, hi ha un remuntador de tipus telefèric que uneix els dos sectors. Els detalls corresponents als remuntadors mecànics de la Vall2000 es poden trobar a la Taula 2. També es pot trobar un mapa esquemàtic de la estació Vall2000 en la Figura 1, localitzada al principi d'aquest document.

Remuntador Mecànic	Sector	Tipus	Estat
RM1	Nord	Telecadira	En servei
RM2	Nord	Teleesquí	En servei
RM3	Nord	Telecadira	En servei
RM4	Nord	Teleesquí	En servei
RM5	Nord	Telecabina	En servei
RM6	Nord	Cinta Transportadora	En servei
RM7	Sud	Teleesquí	En servei
RM8	Sud	Telecadira	En servei
RM9	Sud	Telecabina	Fora de servei
RM10	Inter-Sector	Teleèric	En servei

Taula 2: Informació detallada sobre es remuntadors mecànics existents a la estació Vall2000.

Les Condicions Meteorològiques. L'estat dels remuntadors mecànics depèn de les condicions meteorològiques, en particular, de la velocitat del vent i de la visibilitat a la Vall2020. La velocitat del vent es caracteritza per un nombre sencer positiu. La visibilitat pot ser *Bona* o *Dolenta*. Els remuntadors de la part alta del sector Nord, és a dir, l'RM1, RM2 i RM3, queden fora de servei quan hi ha vents de més de 35 Km/h. Per contra, si la visibilitat a l'estació és *Dolenta* en lloc de *Bona*, el sector Sud es veu afectat, i el remuntador RM9 es queda fora de servei. Finalment, tot els remuntadors queden fora de servei es cas que el vent sigui de més de 60 Km/h.

La Gestió Automàtica de la Vall2000. Per tal de gestionar de forma eficaç la sensibilitat de la Vall200 a les condicions meteorològiques, s'ha desenvolupat un sistema automàtic. Aquest sistema entra en acció cada vegada que canvien les condicions climàtiques, i ordena als remuntadors mecànics que actualitzin automàticament el seu estat en funció de la nova velocitat de vent, i de la nova visibilitat. Un cop actualitzat l'estat de tots els remuntadors, el sistema fa que totes les pistes actualitzen també el seu estat, tancant-se en cas que els remuntadors que permeten d'accedir-hi estiguin fora de servei. La informació sobre quines pistes s'han de tancar en cas que un remuntador mecànic particular quedi fora de servei es pot trobar a l'última columna de la Taula 1 (anomenada Accés Amb). Aquesta informació també es pot complementar amb el mapa de la Figura 1. Actualment, a la Vall2000, el vent és de 10

Km/h, mentre que la visibilitat és *Dolenta*, el que explica el remuntador RM9 fora de servei i les pistes P9 i P10 tancades.

2. Objectiu i Funcionalitats

L'objectiu d'aquesta pràctica és modelar la informació corresponent a l'estació Vall2000 utilitzant un paradigma de programació orientat a objectes. A més a més, l'aplicació desenvolupada haurà d'oferir el següent conjunt de funcionalitats:

1. *Llistar la informació de totes les pistes*
2. *Llistar la informació de les pistes obertes*
3. *Llistar la informació de les pistes tancades*
4. *Llistar la informació de tots els remuntadors mecànics*
5. *Llistar la informació dels remuntadors en servei*
6. *Llistar la informació dels remuntadors fora de servei*
7. *Modificar l'estat de la neu en una pista*
   - L'aplicació haurà de demanar a l'usuari que doni el nom de la pista, l'estat de la qual es vol modificar. Posteriorment, l'aplicació haurà de demanar també el nou estat de la neu. S'haurà de verificar que l'input donat per l'usuari és vàlid.
8. *Calcular i mostrar el total de kilòmetres de pistes, i de pistes obertes*
   - És a dir, el total de kilòmetres que fan les pistes de l'estació Vall2000, així com el total de kilòmetres esquiables en aquest moment a la Vall2000 (tenint en compte només les pistes obertes).
9. *Modificar la velocitat del vent*
   - És a dir, modificar les condicions relatives al vent que es fa sentir a l'estació Vall2000. Recorda que la velocitat del vent té un impacte directe en l'estat dels remuntadors mecànics i de les pistes, que s'haurà de tenir en compte de forma automàtica.
10. *Modificar la visibilitat*
    - És a dir, modificar les condicions de visibilitat a l'estació Vall2000. Recorda que les condicions de visibilitat tenen un impacte directe en l'estat dels remuntadors mecànics i de les pistes, que s'haurà de modificar de forma automàtica.
11. *Mostrar les condicions meteorològiques actuals*

Com veureu de seguida, el codi base ja us proporciona un menú amb aquestes opcions. Aquest codi us permetrà de seleccionar una de les opcions però, inicialment, un cop seleccionada la opció el programa no fa res. Vosaltres haureu d'afegir-li la funcionalitat desitjada.

3. Material pel lliurament

Per aquest lliurament us proporcionem un codi base que conté tres classes necessàries per aquest lliurament:

- ***IniciadorEstacioEsqui***
- ***Menu***
- ***VistaEstacioEsqui***

Podeu trobar aquest codi al Campus Virtual ("Laboratori: Enunciats -> Codi Base Pr 1") i afegir-lo al vostre projecte. En el mateix fitxer zip, també podeu trobar un fitxer de text amb una funció anomenada *inicialitzaDadesVall2000()* que fareu servir a la classe *EstacioEsqui* tal com s'explica a l'apartat 4.6.

4. Descripció del Projecte Base

Al Campus Virtual de l'assignatura podràs trobar tres classes que et serviran de base. Un cop hagis creat el teu projecte NetBeans (veure **4.1** baix) i hi hagis afegit aquestes classes, podràs obrir-les i executar el codi, així com sortir de l'aplicació seleccionant l'opció 12.

4.1 Creació del Projecte Base amb NetBeans

El primer pas serà crear un projecte al NetBeans, al qual li heu de posar com a nom Cognom1Nom1Cognom2Nom2, on 1 i 2 fan referència als dos membres de la parella, i tenint en compte les següents consideracions:

- La primera lletra de cada part en majúscula i la resta en minúscula.
- Eviteu utilitzar accents i caràcters del tipus ñ o ç.

  Per exemple, una estudiant amb nom Dolça Martínez Castaña, hauria de crear un projecte amb el nom MartinezCastanaDolca.
- La classe principal s'ha de dir ***IniciadorEstacioEsqui***, i el paquet per defecte **prog2.vista**. En el nom del paquet utilitzeu els mateixos criteris anteriors.
- En el NetBeans s'ha d'indicar la classe principal com **prog2.vista.IniciadorEstacioEsqui**.

4.2 Codi Base: Classe IniciadorEstacioEsqui

Aquesta classe té com a responsabilitat llançar el bucle principal de l'aplicació. La classe de la vista ***IniciadorEstacioEsqui*** té un mètode estàtic ***main()*** on es crea un objecte de tipus ***VistaEstacioEsqui*** anomenat *vistaEstacioVall2000*. Després crida el mètode ***gestioEstacio()*** de l'objecte *vistaEstacioVall2000*, on es troba el bucle principal de l'aplicació.

4.3 Codi Base: Classe VistaEstacioEsqui

Aquesta classe és responsable de fer la interfície entre l'usuari i el model (que serà detallat més tard). Al Campus Virtual de Programació II trobaràs un fitxer anomenat *VistaEstacioEsqui.java* que hauràs d'utilitzar com a punt de partida per a desenvolupar aquesta classe. També hi trobaràs la classe *Menu.java*, de la qual depèn la classe *VistaEstacioEsqui*. Estudia la classe *VistaEstacioEsqui* (i en particular els mètodes *gestioEstacio()* i *gestioMenu()*). Amb aquest codi ja pots executar l'aplicació. Examina l'output i nota que hi ha parts de l'output que depenen de mètodes (i classes) no implementats, com indicat en la llegenda de la Figura 2 (baix):

```
------------------------------------------------------------------------
MENU NO IMPLEMENTAT ---> estacio.getNomEstacio()
------------------------------------------------------------------------
        1.- Llistar la informació de totes les pistes
        2.- (...)
```

Figura 2. Part del output (és a dir, del menú amb les diferents opcions) donat pel codi base. Nota que, davant de "MENU" hauria d'aparèixer el nom de l'estació (Vall2000 en el nostre cas). Com que aquest és un atribut de la classe EstacioEsqui, que encara no existeix, de moment el menú no es mostra el nom de l'estació al menú.

En la seva versió final, la classe *VistaEstacioEsqui* haurà de contenir un atribut privat de tipus *EstacioEsqui*, anomenat *estacio*, que conté tota la informació sobre l'estació d'esquí.

4. Desenvolupament del Projecte

Les classes necessàries per a desenvolupar la Vall2000 hauran de formar part del paquet prog2.model (que hauràs de crear). Com a regla general, totes les classes del model hauran de tenir un constructor, i un conjunt de *getters* y *setters*. Provisionalment, en aquesta pràctica, i per tal de protegir les classes contra formes d'utilització no desitjades, alguns setters retornaran com a resultat un *String* amb un missatge. El missatge contindrà informació sobre si la modificació del valor de l'atribut s'ha pogut fer o no correctament i, en cas que no, quin és el motiu (per exemple, un valor negatiu de velocitat de vent). Aquesta estratègia de programació serà substituïda en les pràctiques següents per l'ús d'excepcions.

4.1 Classe Meteo

La classe *Meteo* conté la informació meteorològica actual. Conté dos atributs, un per a guardar la velocitat del vent, i una altra per a guardar la visibilitat, així com un conjunt de *getters* i *setters*. Finalment, té també un mètode *toString()*, que transforma la informació meteorològica en un *String*.

4.2 Classe Remuntador i Classes Filles

Per modelar els diferents tipus de remuntadors mecànics utilitzarem el mecanisme d'herència. S'haurà de crear una classe abstracta anomenada ***Remuntador***, de la següent manera:

```
public abstract class Remuntador {

}
```

Aquesta classe contindrà els atributs comuns a tots els remuntadors, és a dir, nom, sector, estat, límit de vent a partir del qual es queda inoperatiu, i si és o no susceptible de ser afectat per les condicions de visibilitat. Totes aquestes dades hauran de ser inicialitzades en el constructor. Els mètodes principals de la classe Remuntador són:

```
public abstract String tipus();

public void actualitzaEstat(Meteo meteo);

public void toString();
```

Fins que no arribis a la funcionalitat 9 (veure Secció 2), la funció actualitzaEstat pot quedar sense implementació. Un cop creada la classe abstracta *Remuntador*, s'hauran de crear 5 classes filles anomenades *Telecabina*, *Telecadira*, *Teleesqui*, *Teleferic* i *CintaTransportadora*. Aquestes classes implementaran cadascuna el seu mètode *tipus()*, que retorna el tipus de remuntador. El mètode *toString()* per qualsevol tipus de remuntador s'ha d'implementar a la classe mare (*Remuntador*) en una sola funció. Exemple d'output:

```
Remuntador: RM9, Tipus: Telecabina, Sector: Sud, Estat: Fora de Servei
```

4.3 Classe Pista

La classe ***Pista*** conté tots els atributs d'una pista, és a dir, nom, sector, color, longitud, estat de la neu i estat de la pista. Més tard, afegirem també la possibilitat de guardar la

informació de quins són els remuntadors dels que depèn la pista. A part dels *getters* i *setters*, els mètodes de la classe Pista són els següents:

```
public void actualitzaEstat();
public void afegirDependencia(Remuntador rm);
public void toString();
```

Per ara, deixarem sense implementar els mètodes *actualitzaEstat()* i *afegirDependencia*(). El mètode *toString()* haurà de retornar un *String* amb tota la informació de la pista en una sola línia. Per exemple:

```
Pista: P12, Sector: Sud, Color: Blava, Longitud: 1.9, Estat de la Neu: Pols, Estat de la Pista: Oberta, Dependències: RM8
```

4.4 Classe *LlistaRemuntadors*

Crea la classe ***LlistaRemuntadors*** que, com el nom indica, té com a objectiu guardar i gestionar un conjunt de remuntadors. Per tal de guardar els remuntadors, cal que aquesta classe tingui un atribut privat de tipus *ArrayList<Remuntador>* anomenat *llista*. Els seus mètodes principals són els següents:

```
public void actualitzaEstat(Meteo meteo);
public void afegirRemuntador (Remuntador rm);
public boolean totsForaDeServei();
public String llistarRemuntadors(String estat);
public String getNoms();
```

El mètode *actualitzaEstat()* el veurem més tard. El mètode *afegirRemuntador()* permet afegir un nou remuntador a la *llista*. El mètode *totsForaDeServei()* ens indica si tots els remuntadors de la llista estan *Fora de Servei* o no. El mètode *llistarRemuntadors(String estat)* ens haurà de retornar un *String* amb els detalls de tots els remuntadors que estiguin en l'estat *estat*. L'estat *estat* podrà ser "En Servei", "Fora de Servei" o "Tots". En l'últim cas es donaran els detalls de tots els remuntadors independentment del seu estat. Finalment, el mètode *getNoms()* ens haurà de retornar un *String* amb els noms de tots els remuntadors de la llista.

Impacte a la classe Pista. Un cop hagis desenvolupat la classe *LlistaRemuntadors*, hauràs d'afegir un atribut d'aquest tipus a la classe *Pista* (anomenat *dependencies*). També podràs implementar el mètode *afegirDependencia(Remuntador rm)* de la classe *Pista*, que permetrà afegir a una pista els remuntadors dels qui depèn per poder funcionar. Caldrà utilitzar el mètode *getNoms()* en el mètode *toString()* de la classe *Pista*.

4.5 Llista de Pistes

La classe ***LlistaPistes*** és molt similar a la classe *LlistaRemuntadors*, però per a *Pistes*. Els seus mètodes principals són:

```
public void actualitzaEstat();

public void afegirPista (Pista p);

public Pista getPista (String nom);

public String llistarPistes(String estat);

public float calculaKmsPistes(String estat);
```

El mètode *getPista()* ens permet trobar una pista pel seu nom. El *llistarPistes()* retorna un *String* amb les pistes en l'estat *estat*. El mètode *calculaKmsPistes()* retorna el total de Kms de pistes en l'estat *estat*. En tots dos casos, l'estat podrà ser "Oberta", "Tancada" o "Tots". En l'últim cas es donaran els detalls de totes les pistes independentment del seu estat.

4.6 La Classe EstacioEsqui

La classe ***EstacioEsqui*** conté tota la informació sobre l'estació. L'hauràs de desenvolupar de forma autònoma, seguint el teu criteri de programador(a). A través d'aquesta classe, la vista (és a dir, la classe *VistaEstacioEsqui*) pot demanar informació sobre l'estació i, eventualment, canviar el seu estat.

Inicialització de les dades de l'Estació Vall2000: En el Campus Virtual pots trobar una funció *inicialitzaDadesVall2000()* que inicialitza les dades de l'estació amb la informació de les Taules 1 i 2. Cal enganxar-la en la classe ***EstacioEsqui***. Aquesta funció haurà de ser cridada des del constructor de la vista.

4.7 Automatització en Funció de les Condicions Climàtiques

Per tal de tenir una solució automatitzada que obri i tanqui pistes i remuntadors en funció de les condicions climàtiques, hauràs d'implementar mètodes *actualitzaEstat* a les classes *EstacioEsqui*, *LlistaRemolcadors*, *LlistaPistes*, *Remolcador* i *Pista*.

5. Format del Lliurament

El lliurament consistirà en tot el codi generat en els diferents punts de l'enunciat, juntament amb la documentació especificada en aquest apartat.

En concret, cal generar un fitxer comprimit (ZIP) amb el nom dels dos membres de la parella: **Cognom1Nom1Cognom2Nom2_L1**, que contingui:

A. El projecte sencer de NetBeans

Tot el codi generat ha d'estar correctament comentat per a poder executar el JavaDoc, generant automàticament la documentació en línia del codi.

B. La memòria del lliurament.

La memòria ha de contenir els punts descrits en la normativa de pràctiques i els punts següents:

1. Explicar les classes implementades.
2. Dibuixar el diagrama de relacions entre les classes que has utilitzat a la teva pràctica. No cal incloure la llista d'atributs i mètodes.
3. Explicar quins són els atributs de la classe **EstacioEsqui** i perquè.
4. Com creieu que podríeu haver desenvolupat el projecte sense les classes LlistaPistes i LlistaRemuntadors? Quin seria l'impacte en l'estructura del vostre codi?
5. Que passaria si cada objecte de tipus Pista creés els seus propis remolcadors (dels quals depèn) i els guardéssiu en la seva llista de remolcadors?
6. Detallar les proves realitzades per comprovar el correcte funcionament de la pràctica, resultats obtinguts i accions derivades.
7. Observacions generals.

6. Data Límit del Lliurament

Consultar el calendari de lliuraments al Campus Virtual.