

Pràctica 2 Programació 2

Continguts:

1. [Llista de tasques](#)
2. [Explicar les classes implementades](#)
3. [Diagrama de relacions](#)
4. [Explicar quins són els atributs de la classe EstacioTreni perquè](#)
5. [Expliqueu perquè el mètode isAccessibilitat de la classe Accés és abstracte](#)
6. [Creieu que l'opció 8 del menú per eliminar una incidència es podria implementar demanant el nom de la via de la incidència, en lloc del número de la incidència que es vol eliminar?](#)
7. [Per què creus que es demana que el mètode carregar de la classe EstacioTren sigui static?](#)
8. [Detallar les proves realitzades per comprovar el correcte funcionament de la pràctica, resultats obtinguts i accions derivades](#)
9. [Observacions generals](#)

1. Llista de tasques

- Implementar la classe Ascensor(Albert Villanueva)
- Implementar la classe CintaTransportadora(Albert Villanueva)
- Implementar la classe Escala(Albert Villanueva)
- Implementar la classe EscalaMecanica(Albert Villanueva)
- Implementar la classe Passadis(Albert Villanueva)
- Implementar la classe EstacioTren(Albert Villanueva)
- Implementar la classe Incidencia(Albert Villanueva)
- Implementar la classe LlistaAccessos(James Chen)
- Implementar la classe LlistaIncidencies(James Chen)
- Implementar la classe LlistaVies(James Chen)
- Implementar la classe Via(Albert Villanueva)

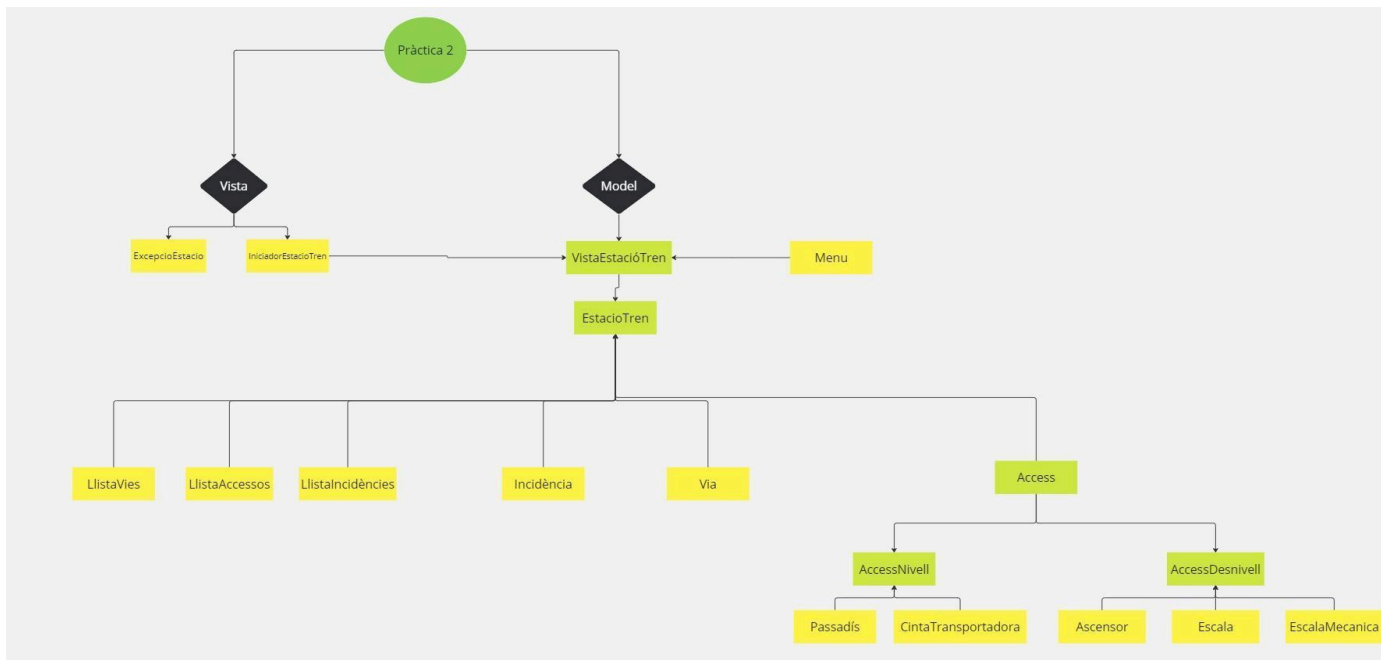
2. Explicar breument les classes implementades

- Ascensor: heretat de accésDesnivell, és accessible
- CintaTransportadora: heretat de accésNivell, no és accessible

- Escala: heretat de accesDesnivell, no és accessible
- EscalaMecanica: heretat de accesDesnivell, no és accessible
- Passadis: heretat de accesNivell, és accessible
- EstacioTren: On s'inicialitzen les dades de l'estació
- Incidencia: Per editar la il·luminació depenent del tipus d'incidència
- LlistaAccessos: implementa un arrayList que guarda accessos, pot actualitzar els estats dels accessos de la llista, calcular quants accessos són accessibles i la longitud dels accessos a nivell.
- LlistaIncidencies: implementa un arrayList que guarda incidències, pot afegir i eliminar incidències a la llista, i imprimir-les
- LlistaVies: implementa un arrayList que guarda vies, pot afegir i eliminar vies, imprimir-les, comprovar si una via existeix a la llista i retornar vies de la llista.
- Menu: un menu amb una llista d'opcions per que el usuari pugui escollir qué fer

Via: classe que inicialitza vies de la estació.

3. Diagrama de relacions



- Los rombos en negro son los paquetes
- Los rectángulos en verde son las clases madre (abstractas)
- Los rectángulos en amarillo son las clases heredadas
- Las flechas de los rectángulos amarillos a los verdes significan que heredan de ellos

4. Explicar quins són els atributs de la classe EstacioTren i perquè.

Els atributs són, LlistaAccessos llistaAccessos, LlistaVies llistaVies, LlistaIncidencies llistaIncidencies.

La classe EstacioTren serveix per a afegir i eliminar incidències. Per això es necessita l'atribut llistaIncidencies. Per a afegir una incidència, es necessita una via, per obtenir la via que es vol accedir s'ha d'iterar sobre la llista de vies que hi ha a l'estació, per tant, s'utilitza l'atribut llistaVies. Després d'afegir una incidència s'ha d'actualitzar els accessos de la via, ja que una via amb incidència no es pot accedir, i per a actualitzar la via que ara és inaccessible per la incidència s'utilitza l'atribut llistaIncidencies.

5. Expliqueu perquè el mètode isAccessibilitat de la classe Acces és abstracte

Perquè isAccessibilitat s'utilitza en classes diferents i a cada classe el contingut de isAccessibilitat canvia. Ascensor i passadís retornen true. Escala, escala mecànica i cinta transportadora retornen false. Com les classes filles d'accés no comparteixen un comportament en comú del mètode isAccessibilitat, s'implementarà el mètode com abstracte per així obligar les subclasses a proporcionar les seves implementacions.

6. Creieu que l'opció 8 del menú per eliminar una incidència es podria implementar demanant el nom de la via de la incidència, en lloc del número de la incidència que es vol eliminar?

No, ja que una via pot tenir més d'una incidència i amb només el nom de la via no es pot saber quina incidència es vol eliminar.

7. Per què creus que es demana que el mètode carregar de la classe EstacioTren sigui static?

Posant el mètode en estàtic fa que no cal crear un objecte d'estacioTren i directament poder utilitzar-ho amb EstacioTren.carregar(camiOrigen).

8. Detallar les proves realitzades per comprovar el correcte funcionament de la pràctica, resultats obtinguts i accions derivades

Per comprovar el funcionament de la pràctica hem fet les següents proves:

1. Llistar tota la informació de les vies i comparar les dades mostrades per pantalla amb les inicialitzades.
2. Hem afegit una incidència per tancar una via i uns accessos. Hem vist com les vies o accessos obertes o tancades canvien a causa de les incidències; les vies amb incidència es tanquen i es modifica la il·luminació, els accessos d'aquelles vies es tanquen. Així hem comprovat el correcte funcionament de les opcions 2, 3, 4, 5.
3. Després hem afegit una incidència i l'hem llistat per comprovar les opcions 6 i 7. Més tard l'hem eliminat i hem tornat a llistar les incidències per comprovar que efectivament havia sigut eliminat; provant l'opció 8.
4. Les següents dues opcions són càlculs que hem verificat fent-los de manera independent al programa per verificar la seva validesa (opcions 9 i 10).
5. Per comprovar el funcionament de guardar i carregar l'estació, hem introduït incidències i hem guardat l'estació (opció 11). Hem aturat l'execució (sortint del menú amb l'opció 13) i hem executat el programa per carregar l'estació (opció 12) i comprovar que les incidències entrades s'havien guardat.

Aquestes són les proves que hem fet per comprovar la funcionalitat del programa.

9. Observacions generals

Amb aquesta pràctica ens hem endinsat en la programació orientada a objectes en Java i en el desenvolupament d'aplicacions simples. Hem treballat amb nous elements com els streams i la persistència de dades i hem desenvolupat un nombre de classes més extens.

Quant a problemes en el desenvolupament de la pràctica ens ha costat entendre el funcionament dels streams i la seva implementació. Tanmateix, hem tingut dificultats en saber quins atributs tenia cada classe i quin era el seu tipus.