

### UF3. Activitats

*Alejandro Garcia Dopico*

**Implementa una solució pels següents problemes (aplicant el disseny modular en tots els casos)**

**4. Prova els diferents mètodes per a llegir un fitxer. Fes una graella comparativa.**

Classe	Descripció	Pros	Contres
<b>FileReader</b>	Classe per llegir caràcters d'un fitxer	Fàcil d'utilitzar, llegint caràcters	No es pot llegir directament una línia completa o un tipus de dades primitiu, sinó que s'han de convertir a través de mètodes externs
<b>BufferedReader</b>	Classe que utilitza un buffer per llegir línies completes d'un fitxer	Més eficient que FileReader, ja que llegeix línies completes	No es pot llegir directament un tipus de dades primitiu, sinó que s'ha de convertir a través de mètodes externs
<b>Scanner</b>	Classe que proporciona una manera senzilla de llegir dades des d'un fitxer o d'entrada estàndard	Pot llegir diferents tipus de dades, com ara enters, doubles i cadenes, i també proporciona funcions de tokenització	Pot ser més lent que les altres classes a l'hora de llegir grans quantitats de dades
<b>DataInputStream</b> (trobat a internet)	Classe per llegir dades primitives des d'un fitxer	Pot llegir diferents tipus de dades primitives	No és tan fàcil d'utilitzar com Scanner i no proporciona funcions de tokenització
<b>ObjectInputStream</b> (trobat a internet)	Classe per llegir objectes serialitzats des d'un fitxer	Pot llegir objectes serialitzats, el que significa que es pot recuperar la informació que s'ha escrit en un fitxer anteriorment	No és compatible amb altres tipus de dades, com ara primitives o cadenes

En general, la classe que s'ha de triar depèn de les necessitats de lectura de cada aplicació. Si es vol llegir línies completes, **BufferedReader** és una opció adequada. Si es vol llegir diferents tipus de dades, **Scanner** és la millor opció. Si es vol llegir objectes serialitzats, **ObjectInputStream** és la millor opció, etc.

8. Prova els diferents mètodes per a escriure informació en un fitxer. Fes una graella comparativa.

Classe	Descripció	Pros	Contres
<b>FileWriter</b>	Classe per escriure caràcters en un fitxer	Fàcil d'utilitzar, escrivint caràcters	No es pot escriure directament una línia completa o un tipus de dades primitiu, sinó que s'han de convertir a través de mètodes externs
<b>BufferedWriter</b>	Classe que utilitza un buffer per escriure línies completes en un fitxer	Més eficient que FileWriter, ja que escriu línies completes	No es pot escriure directament un tipus de dades primitiu, sinó que s'ha de convertir a través de mètodes externs
<b>PrintWriter</b>	Classe que proporciona una manera senzilla d'escriure dades en un fitxer o en sortida estàndard	Pot escriure diferents tipus de dades, com ara enters, doubles i cadenes	Pot ser més lent que les altres classes a l'hora d'escriure grans quantitats de dades
<b>DataOutputStream</b> (trobat a internet)	Classe per escriure dades primitives en un fitxer	Pot escriure diferents tipus de dades primitives	No és tan fàcil d'utilitzar com PrintWriter
<b>ObjectOutputStream</b> (trobat a internet)	Classe per escriure objectes serialitzats en un fitxer	Pot escriure objectes serialitzats, el que significa que es pot desar la informació per a ser recuperada en un futur	No és compatible amb altres tipus de dades, com ara primitives o cadenes

En general, la classe que s'ha de triar depèn de les necessitats d'escriptura de cada aplicació. Si es vol escriure línies completes, **BufferedWriter** és una opció adequada. Si es vol escriure diferents tipus de dades, **PrintWriter** és la millor opció. Si es vol escriure objectes serialitzats, **ObjectOutputStream** és la millor opció, etc.



**13. Actualitza el teu repositori GitHub amb tots els exercicis de la UF3. Un cop fet, escull 3 exercicis diferents (1 de cada company) i escriu si la seva solució és correcta i per què, o no i com la resoldries.**

He posat els meus exercicis al repositori, he vist que els companys no ho tenen pujat o està molt desordenat i no els trobo.