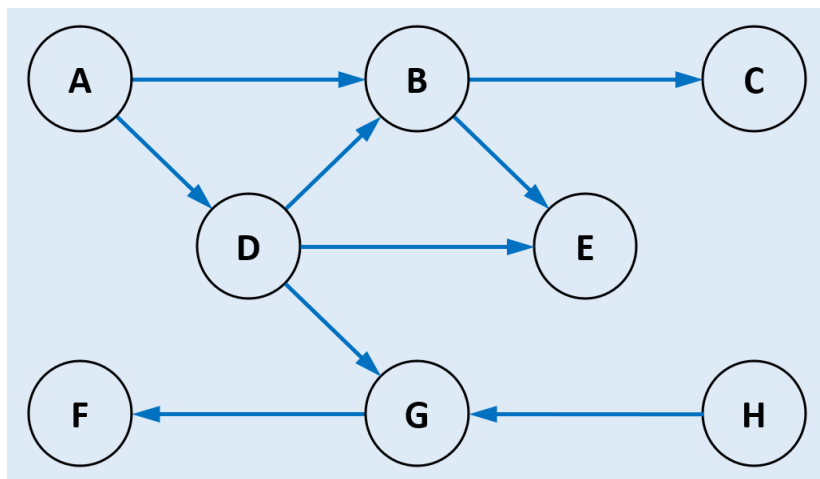


Objectif

- Maîtriser le concept de **graphe**.
- Implémenter un graphe de différentes façons.
- Définir des classes utilisant un **type générique**.

Énoncé du problème

On vous demande de fournir les classes qui stockent le graphe suivant (selon l'une ou plusieurs implémentations étudiées en cours). Conservez pour chaque nœud son nom (A,B,...H) ainsi qu'une description.



Implémentations possibles du graphe / des relations :

List<Noeud>

A B C D E F G H

List<Relation>

A,B A,D B,C B,E D,B D,E D,G G,F H,G

A	B, D
B	C, E
C	
D	B, E, G
E	
F	
G	F
H	G

	A	B	C	D	E	F	G	H
A								
B	X			X				
C								
D	X							
E		X		X				
F								
G				X				
H							X	X

Votre travail :

- Créez les classes implémentant le graphe ci-dessus, et codez les méthodes suivantes :
- void **addRelation**(source, dest) qui ajoute une relation(arête) entre les 2 nœuds(sommets) spécifiés.
- boolean **existeRelation**(source, dest) qui indique si une relation existe entre ces 2 nœuds :
une relation est un lien direct entre 2 nœuds : il existe une relation entre A et D, mais il n'y en a pas entre A et F.
- boolean **existeChemin**(source, dest) qui indique si **un chemin existe** entre ces 2 nœuds :
un chemin indique la possibilité d'arriver à un nœud destination en passant par plusieurs nœuds intermédiaires.
⇒ retournera donc true pour `existeChemin(A,F)`, mais false pour `existeChemin(A,H)` !
- Les nœuds doivent être identifiés par leur nom et contenir une description (non pas uniquement des Integer ou des lettres...)