

TP S rialisation

Soit la classe **Pi ce** qui g n ralise les classes **Pi ceEl mentaire** et **Pi ceCompos e**. Une pi ce peut  tre compos e de plusieurs autres pi ces, alors que les pi ces  l mentaires repr sentent les feuilles d'une telle arborescence de composition.

1. D finir la classe abstraite **Pi ce** qui a comme attributs :
 - numP Num ro de la pi ce
 - descP Description de la pi ce
2. D finir la classe **Pi ceCompos e** qui h rite de la classe **Pi ce** et qui a comme attributs :
 - pi cesComp la liste des pi ces composantes (elles peuvent  tre des pi ces  l mentaires ou des pi ces compos es)
3. d finir la **Pi ceEl mentaire**.
4. Ajouter les constructeurs ad quats   chaque classe.
5. Instancier dans une classe de test les pi ces suivantes :
 - (1,PA)[]
 - (2,PB)[(3,PC),(1,PA)]
 - (4,PD)[]
 - (5,PE)[(4,PD),(2,PB)][(3,PC),(1,PA)]

Le mod le de repr sentation d'une pi ce  tant :

- **Pi ce : (numP,descP)[Composition]**
 - **Composition : {Pi ce,*}**
6. S rialiser dans un fichier "*test.ser*" les quatre pi ces une par une.
 7. En d -s rialisant le contenu du fichier, afficher l'ensemble de pi ces en respectant le mod le de repr sentation.
 8. Ajouter les quatre pi ces   une collection et refaire le test en s rialisant cette derni re.
 9. Refaire la question 8 en appliquant la s rialisation XML.