

M1103 : Structures de Données & Algorithmes fondamentaux

Feuille TP n° 3 – distribuée en semaine DUT n°13

Utilisation des Files

OBJECTIFS PEDAGOGIQUES :

1.- Apprendre les manipulations de base d'un Type abstrait de Données File

2.- Découvrir le principe de gestion des exceptions

ÉVALUATION :

- L'exercice abordé dans cette feuille TP fera l'objet d'une évaluation en fin de module

RESSOURCES A VOTRE DISPOSITION POUR REALISER CE TP :

- Sur le WebCampus, dans la zone associée à ce module APL-M1103 :
 - tp3.pdf : le présent sujet de tp
 - dans une archive ressourcesTP3.zip :
 - file.h, file.cpp : les fichiers implémentant le type UneFile.
 - jeuDeCartes.h, jeuDeCartes.cpp : le paquetage de gestion de cartes
 - main.cpp : le programme principal du jeu, à compléter
 - jeu51.exe : l'exécutable du programme complet, comme modèle de programme à reproduire

DIRECTIVES PARTICULIERES POUR CETTE SEANCE DE TP

- Créer un répertoire tp3.
- Copier tous les fichiers qui vous ont été donnés dans le répertoire tp3
- À l'intérieur de ce répertoire, créer un projet JeuDu51
- Déplacer les fichiers source dans le projet

TRAVAIL A FAIRE SUR CETTE FEUILLE DE TP

- 1) Compléter le code source du fichier main.cpp conformément aux directives de la feuille de TD n°3.
Toutes les parties à compléter sont identifiées par la mention */*à compléter */*
À noter : Il ne manque que les instructions 'métiers', c'est-à-dire celles qui gèrent le 'moteur' du jeu. N'ajoutez aucune instruction de 'mise en page' (affichage, saut de ligne, espaces....), tout est déjà prévu dans le programme qui vous a été fourni.
- 2) **À noter :** Le codage de l'action piocherUneCarte dans le programme principal. Cette action *sait prendre en charge* les exceptions (erreurs) éventuellement levées par le paquetage jeuDeCartes et évite donc que le programme ne se termine en état d'erreur. Explications :
 - En appelant la primitive nouvelleCarte(), le programme *tente d'obtenir* une carte du paquet de cartes (l'instruction try enveloppe l'instruction risquant de générer une exception).
 - Si le paquet de cartes est vide, la primitive nouvelleCarte() *lèvera l'exception* 'paquetEpuise' (c'est le rôle de l'instruction throw dans le corps de la primitive nouvelleCarte()).
 - Mais le programme principal *est capable d'intercepter et de gérer une éventuelle erreur* provenant du paquetage jeuDeCartes (c'est le rôle de l'instruction catch, elle *traite* l'exception levée par le sous-programme nouvelleCarte()).
Dans ce cas, le programme sait qu'il n'y a plus de cartes et arrêtera le jeu (proprement) selon la manière prévue dans les spécifications.

Vous demandez à votre enseignant ce que vous ne comprenez pas.