

Les exceptions :

Exercice 1: Ecrire un programme dans lequel vous demandez à l'utilisateur de saisir deux entiers en gérant l'exception dans le cas où l'utilisateur ne saisi pas un entier correctement en lui demandant de refaire la saisie.

Exercice 2: Créez une classe Elève caractérisée par nom, âge et moyenne, l'âge doit être entre 18 et 26 sinon l'exception InvalidAgeException (que vous devez créer avec le message « L'âge doit être entre 18 et 26 ») est générée. La note doit être entre 0 et 20 sinon l'exception InvalidNoteException est générée (InvalidNoteException doit être créée aussi avec le message « La note doit être entre 0 et 20 »). Définir les constructeurs de la classe, les accesseurs et les méthodes ToString et Equals. Définir une classe Program qui permet de tester la classe en gérant les exceptions.

Les fichiers :

Exercice 1: Ecrire un programme qui demande à l'utilisateur de saisir des phrases (des lignes) et de les enregistrer dans un fichier nommé « données.txt », la saisie s'arrête lorsque l'utilisateur entre une ligne vide. Gérer les exceptions qui peuvent être déclenchées.

Exercice 2: Ecrire un programme qui permet d'afficher à l'utilisateur le contenu du fichier « données.txt » créé dans l'exercice 1. Gérer les exceptions qui peuvent être déclenchées.

Exercice 3: Ecrire un programme qui demande à l'utilisateur un nom de fichier et qui duplique ensuite ce fichier (créer une copie) sous le nom « Copie_de_nom du fichier original ». . Gérer les exceptions qui peuvent être déclenchées.

La sérialisation :**Exercice 1:**

a. Définir une classe Personne sérialisable caractérisée par nom, prénom et âge (non sérialisable) et qui doit être un entier positif sinon une exception est générée, créer les constructeurs (par défaut, d'initialisation, et de recopie), les accesseurs des propriétés ainsi que les méthodes ToString et Equals.

b. Définir un programme dans lequel vous proposez le menu suivant à l'utilisateur :

- 1: Pour saisir les informations d'une personne et l'enregistrer
- 2: Pour charger les informations d'une personne.
- 3: Pour quitter

Le premier choix permet de saisir les informations d'un objet de type Personne et de le sérialiser dans un fichier « Personne.dat ». Le deuxième choix permet de désérialiser l'objet à partir du fichier « Personne.dat »

c. Définir un programme qui permet de gérer une collection de personnes de type ArrayList, et de proposer le menu suivant à l'utilisateur :

- 1: Pour ajouter une personne
- 2: Pour rechercher une personne par son nom
- 3: Pour modifier une personne
- 4: Pour supprimer une personne
- 5: Pour afficher la liste des personnes
- 6: Pour enregistrer la liste des personnes dans un fichier
- 7: Pour charger la liste des personnes à partir d'un fichier
- 8: Pour quitter le programme