UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

PROJET DES TOURS D’HANOÏ

TRAVAIL PRÉSENTÉ

COMME EXIGENCE PARTIELLE DU COURS

INF-1004 : STRUCTURES DE DONNÉES ET ALGORITHMES

PAR

JULIEN BOISVERT ET ALEXIS YOUNG

14 MARS 2023

**Listing du code**

namespace ProjectTourHanoi  
{  
 public class Anneau  
 {  
 public int Diametre { get; }  
  
 public Anneau(int diametre)  
 {  
 Diametre = diametre;  
 }  
   
 public override string ToString()  
 {  
 return Diametre.ToString();  
 }  
 }  
}

using System;  
using System.Security.Cryptography;  
  
namespace ProjectTourHanoi  
{  
 public class Tour  
 {  
  
 public char \_lettreTour;  
 private int \_nbAnneau;  
 public int \_top;  
 private Anneau[] \_tours;  
   
 public Tour(char lettreTour, int nbAnneau, int maxAnneau)  
 {  
 \_lettreTour = lettreTour;  
 \_nbAnneau = nbAnneau;  
 \_tours = new Anneau[maxAnneau];  
 \_top = nbAnneau - 1;  
 int diametre = nbAnneau;  
   
 for (int i = 0; i < nbAnneau; i++)  
 {  
 \_tours[i] = new Anneau(diametre);  
 diametre--;  
 }  
  
 }  
   
 *//Fonctions primaires de la pile Tour[]* public void push(Anneau nbAnneau)  
 {  
 \_tours[++\_top] = nbAnneau;  
 }  
  
 public Anneau pop()  
 {  
 return \_tours[\_top--];  
 }  
  
 *//Fonctions secondaires de la pile Tour[]* public Anneau peek()  
 {  
 if (\_top != -1)  
 {  
 return \_tours[\_top];  
 }  
 else  
 {  
 return (new Anneau(10));  
 }  
 }  
  
 public bool isEmpty()  
 {  
 return (\_top == -1);  
 }  
  
 public bool isFull()  
 {  
 return (\_top == \_nbAnneau - 1);  
 }  
  
 public void clear()  
 {  
 while (!isEmpty())  
 {  
 pop();  
 }  
 }  
  
 public override string ToString()  
 {  
 string affichage = \_lettreTour + ": ";  
  
 int dif = \_tours.Length - (\_top + 1);  
  
 for (int i = 0; i < (\_top + 1); i++)  
 {  
 affichage += \_tours[i].ToString();  
 }  
  
 for (int i = 0; i < dif; i++)  
 {  
 affichage += "-";  
 }  
   
   
 return affichage;  
 }  
 }  
}

namespace ProjectTourHanoi  
{  
 public class ToursHanoi  
 {  
 private Tour[] \_tours = new Tour[3];  
 private int \_nbAnneau;  
   
 *//Constructeur* public ToursHanoi(int nbAnneau)  
 {  
 \_nbAnneau = nbAnneau;  
 \_tours[0] = new Tour('A',nbAnneau,nbAnneau);  
 \_tours[1] = new Tour('B',0,nbAnneau);  
 \_tours[2] = new Tour('C',0,nbAnneau);  
 reinitialiser();  
 }  
  
 *//Réinitialise le jeu* public void reinitialiser()  
 {  
 *//Vide les tours* \_tours[0].clear();  
 \_tours[1].clear();  
 \_tours[2].clear();  
  
 *//Ajout des anneaux sur la tour A* for (int i = \_nbAnneau; i > 0; i--)  
 {  
 \_tours[0].push(new Anneau(i));  
 }  
 }  
  
 *//Déplace lun anneau si le mouvement est légal* public bool deplacer(int de, int vers)  
 {  
 Anneau nb = \_tours[de].peek();  
  
 *//Vérifie si l'anneau a déplacer est plus petit* if (nb.Diametre < \_tours[vers].peek().Diametre)  
 {  
 *//Déplacement de l'anneau* \_tours[de].pop();  
 \_tours[vers].push(nb);  
   
 *//Affichage du déplacement* Console.WriteLine("L'anneau de diamètre " + nb + " est déplacé de la tour " + de + " vers la tour " + vers);  
 return true;  
 }  
 else  
 {  
 *//Affichage du message d'erreur* Console.WriteLine("Déplacement illégal");  
 return false;  
 }  
  
 return false;  
  
 }  
  
 *//Transforme les lettres des tour en indice int (a = 0, b = 1, c = 2)* public int transform()  
 {  
 int nb = -1;  
 bool fin = false;  
 string de;  
  
 *//Boucle qui prend fin lors d'un résultat valide* while (!fin)  
 {  
   
 de = Console.ReadLine();  
   
 *//Si la tour est a, retourne 0 et sort de la boucle* if (de.Equals("a") || de.Equals("A"))  
 {  
 nb = 0;  
 fin = true;  
 }  
  
 *//Si la tour est b, retourne 1 et sort de la boucle* else if (de.Equals("b") || de.Equals("B"))  
 {  
 nb = 1;  
 fin = true;  
 }  
  
 *//Si la tour est c, retourne 2 et sort de la boucle* else if (de.Equals("c") || de.Equals("C"))  
 {  
 nb = 2;  
 fin = true;  
 }  
  
 *//Si la tour est non-valide, affiche le message d'erreur et reste dans la boucle* else  
 {  
 Console.WriteLine("Choix de tour non-valide");  
 }  
 }  
  
 return nb;  
 }  
  
 public void resoudre()  
 {  
 reinitialiser();  
 deplacerAuto(\_nbAnneau,0,1,2);  
 }  
   
 private void deplacerAuto(int disque,int de, int inter, int vers)  
 {  
 if (disque == 1)  
 {  
 deplacer(de, vers);  
 }  
  
 else  
 {  
 deplacerAuto(disque-1,de,vers,inter);  
 deplacer(de, vers);  
 deplacerAuto(disque - 1,inter,de,vers);  
 }  
 }  
  
 public override string ToString()  
 {  
 string affiche = "";  
 for (int i = 0; i < 3; i++)  
 {  
 affiche += \_tours[i]+"\n";  
 }  
 return affiche;  
 }  
 }  
}

namespace ProjectTourHanoi  
{  
 class Program  
 {  
 private static ToursHanoi jeu;  
   
 static void Main(string[] args)  
 {  
 string choix;  
 bool fin = false;  
   
 jeu = new ToursHanoi(3);  
  
 while (!fin)  
 {  
 *//Affichage du menu* Console.WriteLine("\n\nTours:");  
 Console.WriteLine(jeu);  
  
 Console.WriteLine("MENU\n");  
  
 Console.WriteLine("1:Déterminer le nombre d'anneaux (3 par défaut)");  
 Console.WriteLine("2: Réinitialiser les tours");  
 Console.WriteLine("3: Jouer un coup");  
 Console.WriteLine("4: Montrer la solution");  
 Console.WriteLine("5: Quitter");  
 Console.WriteLine("Faites votre choix et appuyer sur ENTER");  
   
 *//Récupération du choix de l'utilisateur* choix = Console.ReadLine();  
  
 *//Appel de la fonction selon le choix* switch (choix)  
 {  
 case "1":  
 choix1();  
 break;  
   
 case "2":  
 choix2();  
 break;  
   
 case "3":  
 choix3();  
 break;  
   
 case "4":  
 jeu.resoudre();  
 break;  
   
 *//Quitte la boucle* case "5":  
 fin = true;  
 break;  
   
 *//Message d'erreur si choix non-valide* default:  
 Console.WriteLine("Choix non-valide");  
 break;  
 }  
   
 }  
 }  
  
 static void choix1()  
 {  
 int nb;*//Variable du nombre d'anneaux* bool fin = false;*//Variable pour la boucle  
  
 //Boucle jusqu'à ce qu'un résultat valide soit entré* while (!fin)  
 {  
 *//Récupération du nombre d'anneau* Console.WriteLine("Veuiller entrer un nombre d'anneau entre 1 et 9");  
 nb = Convert.ToInt16(Console.ReadLine());  
  
 *//Validation du nombre entré* if (nb < 10 && nb > 0)  
 {  
 *//Création du nouveau jeu avec le nombre d'anneaux* jeu = new ToursHanoi(nb);  
   
 *//Termine la boucle* fin = true;  
 }  
   
 *//Si le nombre est , retour au menu* else if (nb == 0)  
 {  
 Console.WriteLine("Retour au menu principal");  
 fin = true;  
 }  
   
 *//Si le nombre est invalide, affichage du message d'erreur* else  
 {  
 Console.WriteLine("Choix non-valide");  
 }  
 }  
 }  
  
 static private void choix2()  
 {  
 jeu.reinitialiser();  
 }  
  
 static private void choix3()  
 {  
 Console.WriteLine("De quelle tour prenez-vous l'anneau? (A - B - C)");  
 int \_base = jeu.transform();  
   
 Console.WriteLine("Vers quelle tour envoyez-vous l'anneau? (A - B - C)");  
 int fin = jeu.transform();  
  
  
 jeu.deplacer(\_base, fin);  
 }  
  
   
 }  
}

**Distribution des tâches dans l’équipe**

|  |  |
| --- | --- |
| **Tâche** | **Responsable** |
| Ébauche de classe Anneau | Alexis Young |
| Ébauche de classe Tour | Julien Boisvert |
| Ébauche de classe ToursHanoi | Alexis Young |
| Classe Programme | Julien Boisvert et Alexis Young |
| Révision des classes | Julien Boisvert et Alexis Young |
| Agencement et finalisation des classes | Julien Boisvert et Alexis Young |
| Documentation et rapport final | Julien Boisvert |

**Guide utilisateur**

**Une image contenant texte

Description générée automatiquement**

**Une image contenant texte

Description générée automatiquement**

**Une image contenant texte

Description générée automatiquement**

**Une image contenant texte

Description générée automatiquement**

**Une image contenant texte

Description générée automatiquement**

**Une image contenant texte

Description générée automatiquement**

**Une image contenant texte

Description générée automatiquement**

**Une image contenant texte

Description générée automatiquement**

**Une image contenant texte

Description générée automatiquement**