**Name: Nguyễn Thị Hà Giang**

**MSV: 22IT072**

src: [Bai-tap-Web-nang-cao/baitap\_tuan5/lab02\_Collection and Generic at main · zannie-hg/Bai-tap-Web-nang-cao · GitHub](https://github.com/zannie-hg/Bai-tap-Web-nang-cao/tree/main/baitap_tuan5/lab02_Collection%20and%20Generic)

**LAB 02 COLLECTION AND GENERIC**

A close up of a text

AI-generated content may be incorrect.

**Book.cs**

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace exercise1

{

internal class Book : IBook

{

private string isbn;

private string title;

private string author;

private string publisher;

private int year;

//luu tru mang chua ten chuong

private ArrayList chapter = new ArrayList();

public string this[int index]

{

get

{

if (index >= 0 && index < chapter.Count)

{

return (string)chapter[index];

}

else

{

throw new IndexOutOfRangeException();

}

}

set

{

if (index >= 0 && index < chapter.Count)

{

chapter[index] = value;

}

else if(index == chapter.Count)

chapter.Add(value);

else

{

throw new IndexOutOfRangeException();

}

}

}

public string Title { get => throw new NotImplementedException(); set => throw new NotImplementedException(); }

public string Author { get => throw new NotImplementedException(); set => throw new NotImplementedException(); }

public string Publisher { get => throw new NotImplementedException(); set => throw new NotImplementedException(); }

public string ISBN { get => throw new NotImplementedException(); set => throw new NotImplementedException(); }

public int Year { get => throw new NotImplementedException(); set => throw new NotImplementedException(); }

public void Show()

{

// xuất thông tin sách ra màn hình console

Console.WriteLine("-------------------------");

Console.WriteLine("Title: " + title);

Console.WriteLine("Author: " + author);

Console.WriteLine("Publisher: " + publisher);

Console.WriteLine("Year: " + year);

Console.WriteLine("ISBN: " + isbn);

Console.WriteLine("Chapter: ");

for (int i = 0; i < chapter.Count; i++)

Console.WriteLine("\t{0}: {1}", i + 1, chapter[i]);

Console.WriteLine("-------------------------");

}

public void Input()

{

Console.Write("Title: ");

title = Console.ReadLine();

Console.Write("Author: ");

author = Console.ReadLine();

Console.Write("Publisher: ");

publisher = Console.ReadLine();

Console.Write("ISBN: ");

isbn = Console.ReadLine();

Console.Write("Year: ");

year = int.Parse(Console.ReadLine());

Console.WriteLine("Input chapter (finished with empty string)");

string str;

do // nhập lần lượt các chương sách

{

str = Console.ReadLine();

if (str.Length > 0)

chapter.Add(str);

} while (str.Length > 0); // kết thúc khi chuỗi rỗng

}

}

}

**Ibook.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace exercise1

{

internal interface IBook

{

//mo ta indexer chuong sach

string this[int index] { get; set; }

//property title dai dien cho ten cua sach

string Title { get; set; }

//property author dai dien cho ten cua tac gia cuon sach

string Author { get; set; }

//property publisher dai dien cho ten cua NXB

string Publisher { get; set; }

//property ISBN dai dien cho so hieu

string ISBN { get; set; }

//property year dai dien cho nam xuat ban

int Year { get; set; }

//Mo ta phuong thuc hien thi thong tin cuon sach

void Show();

}

}

**BookList.cs**

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace exercise1

{

internal class BookList

{

// định nghĩa danh sách Book

private ArrayList list = new ArrayList();

/// Tạo một đối tượng sách, nhập các thông tin và thêm vào list

public void AddBook()

{

Book b = new Book();

// nhập thông tin

b.Input();

// thêm vào danh sách

list.Add(b);

}

/// Xuất danh sách

public void ShowList()

{

foreach (Book b in list)

b.Show();

}

public void InputList()

{

int n;

Console.Write("Amount of books: ");

n = int.Parse(Console.ReadLine());

while (n > 0)

{

AddBook();

n--;

}

}

}

}

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace exercise1

{

internal class Program

{

static void Main(string[] args)

{

// tạo một đối tượng BookList

BookList bl = new BookList();

bl.InputList();

bl.ShowList();

Console.ReadLine();

}

}

}

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

**A white background with black text

AI-generated content may be incorrect.**

**Book.cs**

using System;

using System.Collections.Generic;

namespace exercise2

{

internal class Book : IComparable<Book>

{

public string Title { get; set; }

public string Author { get; set; }

public string Publisher { get; set; }

public string ISBN { get; set; }

public int Year { get; set; }

public List<string> Chapters { get; set; } = new List<string>();

public void Input()

{

Console.Write("Title: ");

Title = Console.ReadLine();

Console.Write("Author: ");

Author = Console.ReadLine();

Console.Write("Publisher: ");

Publisher = Console.ReadLine();

Console.Write("ISBN: ");

ISBN = Console.ReadLine();

Console.Write("Year: ");

Year = int.Parse(Console.ReadLine());

Console.WriteLine("Input chapters (finished with empty string):");

while (true)

{

string chapter = Console.ReadLine();

if (string.IsNullOrEmpty(chapter)) break;

Chapters.Add(chapter);

}

}

public void Show()

{

Console.WriteLine("---------------------------");

Console.WriteLine($"Title: {Title}");

Console.WriteLine($"Author: {Author}");

Console.WriteLine($"Publisher: {Publisher}");

Console.WriteLine($"Year: {Year}");

Console.WriteLine($"ISBN: {ISBN}");

Console.WriteLine("Chapters:");

for (int i = 0; i < Chapters.Count; i++)

{

Console.WriteLine($" {i + 1}: {Chapters[i]}");

}

}

// Thực hiện sắp xếp theo Title

public int CompareTo(Book other)

{

return Title.CompareTo(other.Title);

}

}

}

**BookList.cs**

using System;

using System.Collections.Generic;

namespace exercise2

{

internal class BookList

{

private List<Book> list = new List<Book>();

public void AddBook()

{

Book b = new Book();

b.Input();

list.Add(b);

}

public void ShowList()

{

foreach (Book b in list)

b.Show();

}

public void InputList()

{

Console.Write("Amount of books: ");

int n = int.Parse(Console.ReadLine());

while (n-- > 0)

{

AddBook();

}

}

public void SortBooksByTitle()

{

list.Sort(); // Vì Book đã thực hiện IComparable nên có thể sắp xếp trực tiếp

}

}

}

**IBook.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace exercise2

{

internal interface IBook

{

//mo ta indexer chuong sach

string this[int index] { get; set; }

//property title dai dien cho ten cua sach

string Title { get; set; }

//property author dai dien cho ten cua tac gia cuon sach

string Author { get; set; }

//property publisher dai dien cho ten cua NXB

string Publisher { get; set; }

//property ISBN dai dien cho so hieu

string ISBN { get; set; }

//property year dai dien cho nam xuat ban

int Year { get; set; }

//Mo ta phuong thuc hien thi thong tin cuon sach

void Show();

}

}

**Program.cs**

using System;

namespace exercise2

{

internal class Program

{

static void Main(string[] args)

{

BookList bookList = new BookList();

// Nhập danh sách sách

bookList.InputList();

// Hiển thị danh sách trước khi sắp xếp

Console.WriteLine("\nDanh sách sách trước khi sắp xếp:");

bookList.ShowList();

// Sắp xếp sách theo tiêu đề

bookList.SortBooksByTitle();

// Hiển thị danh sách sau khi sắp xếp

Console.WriteLine("\nDanh sách sách sau khi sắp xếp theo tiêu đề:");

bookList.ShowList();

Console.ReadLine();

}

}

}

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**A white background with black text

AI-generated content may be incorrect.**

**A text on a white background

AI-generated content may be incorrect.**

**Account.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace exercise3

{

internal class Account

{

public int AccountID { get; set; }

public double Balance { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public Account()

{

}

public Account(int accountID, double balance, string firstName, string lastName)

{

AccountID = accountID;

Balance = balance;

FirstName = firstName;

LastName = lastName;

}

public override string ToString()

{

return $"ID: {AccountID}, Name: {FirstName} {LastName}, Balance: {Balance}";

}

}

}

**AccountList.cs**

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace exercise3

{

internal class AccountList

{

//List<Account> AccList { get; set; }

private List<Account> AccList;

public AccountList()

{

AccList = new List<Account>();

}

public void NewAccount()

{

//AccList.Add(acc);

//Console.WriteLine("Nhập số tài khoản cần thêm vào danh sách: ");

//int n = int.Parse(Console.ReadLine());

//for (int i = 0; i < n; i++)

//{

// Console.WriteLine($"Nhập thông tin tài khoản {i+1}:");

//}

Console.WriteLine("Nhập ID: ");

int accountId = int.Parse(Console.ReadLine());

Console.WriteLine("Nhập FirstName: ");

string firstName = Console.ReadLine();

Console.WriteLine("Nhập LastName: ");

string lastName = Console.ReadLine();

Console.WriteLine("Nhập Balance: ");

double balance = double.Parse(Console.ReadLine());

// Tạo một đối tượng Account mới và thêm vào danh sách

Account acc = new Account(accountId, balance, firstName, lastName);

AccList.Add(acc);

}

public void ShowAccount()

{

Console.WriteLine("\nDanh sách tài khoản:");

foreach (var acc in AccList)

{

Console.WriteLine(acc);

}

}

public void SaveFile()

{

// nhập tên file

Console.Write("Input file name to save: ");

string filename = Console.ReadLine();

// ghi file

try

{

// tạo luồng truy cập file

FileStream output = new FileStream(filename, FileMode.CreateNew, FileAccess.Write);

// thiết lập writer

StreamWriter writer = new StreamWriter(output);

// duyệt qua từng đối tượng trong Accounts

foreach (Account acc in AccList)

{

// lưu các thông tin của một account trên dòng, phân cách nhau bằng dấu ','

writer.WriteLine("{0},{1},{2},{3}", acc.AccountID, acc.FirstName, acc.LastName, acc.Balance);

}

// đóng kết nối

writer.Close();

output.Close();

}

catch (IOException e) // phát sinh ngoại lệ nếu có

{

Console.WriteLine(e.Message);

}

}

public void LoadFile()

{

// đọc tên file chứa dữ liệu

Console.Write("Input file name to load: ");

string filename = Console.ReadLine();

// xóa danh sách hiện có

AccList.Clear();

try

{

// tạo luồng đọc file

FileStream input = new FileStream(filename, FileMode.Open, FileAccess.Read);

StreamReader reader = new StreamReader(input);

string str;

// đọc qua từng dòng, kết thúc khi chuỗi null

while ((str = reader.ReadLine()) != null)

{

// tách chuỗi đọc ra thành các phần tử con, phân cách nhau dấu ','

string[] list = str.Split(',');

// tạo đối tượng Account mới

Account acc = new Account(int.Parse(list[0]), double.Parse(list[3]), list[1], list[2]);

// đưa đối tượng mới đọc vào danh sách account

AccList.Add(acc);

}

// đóng luồng

input.Close();

reader.Close();

}

catch (IOException e)

{

Console.WriteLine(e.Message);

}

}

}

}

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace exercise3

{

internal class Program

{

static void Main(string[] args)

{

AccountList accountList = new AccountList();

//accountList.NewAccount(new Account(1, "Thanh", "Thảo", 400000));

//accountList.NewAccount(new Account(2, "Đào", "Giang", 20000));

//accountList.NewAccount(new Account(3, "Thanh", "hải", 50000));

//accountList.NewAccount(new Account(4, "Minh", "Đăng, 10000"));

Console.OutputEncoding = Encoding.UTF8;

int chon;

do

{

Console.WriteLine("\n1. Thêm nhân viên");

Console.WriteLine("2. Hiển thị danh sách");

Console.WriteLine("3. Lưu lại file");

Console.WriteLine("4. Tải file ");

Console.WriteLine("0. Thoát");

Console.Write("Chọn: ");

chon = int.Parse(Console.ReadLine());

switch (chon)

{

case 1: accountList.NewAccount(); break;

case 2: accountList.ShowAccount(); break;

case 3: accountList.SaveFile(); break;

case 4: accountList.LoadFile(); break;

case 0: Console.WriteLine("Thoát."); break;

default: Console.WriteLine("Sai lựa chọn!"); break;

}

} while (chon != 0);

}

}

}

**A screenshot of a computer

AI-generated content may be incorrect.**

A screenshot of a computer

AI-generated content may be incorrect.

**File được đọc:**

**A screenshot of a computer

AI-generated content may be incorrect.**

A close-up of a document

AI-generated content may be incorrect.

**Account.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace exercise4

{

[Serializable]

internal class Account

{

public int AccountID { get; set; }

public double Balance { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public Account()

{

}

public Account(int accountID, double balance, string firstName, string lastName)

{

AccountID = accountID;

Balance = balance;

FirstName = firstName;

LastName = lastName;

}

public override string ToString()

{

return $"ID: {AccountID}, Name: {FirstName} {LastName}, Balance: {Balance}";

}

}

}

**AccountList.cs**

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Runtime.Serialization.Formatters.Binary;

using System.Text;

using System.Threading.Tasks;

using exercise4;

namespace exercise4

{

internal class AccountList

{

//List<Account> AccList { get; set; }

private List<Account> AccList;

public AccountList()

{

AccList = new List<Account>();

}

public void NewAccount()

{

//AccList.Add(acc);

//Console.WriteLine("Nhập số tài khoản cần thêm vào danh sách: ");

//int n = int.Parse(Console.ReadLine());

//for (int i = 0; i < n; i++)

//{

// Console.WriteLine($"Nhập thông tin tài khoản {i+1}:");

//}

Console.WriteLine("Nhập ID: ");

int accountId = int.Parse(Console.ReadLine());

Console.WriteLine("Nhập FirstName: ");

string firstName = Console.ReadLine();

Console.WriteLine("Nhập LastName: ");

string lastName = Console.ReadLine();

Console.WriteLine("Nhập Balance: ");

double balance = double.Parse(Console.ReadLine());

// Tạo một đối tượng Account mới và thêm vào danh sách

Account acc = new Account(accountId, balance, firstName, lastName);

AccList.Add(acc);

}

public void ShowAccount()

{

Console.WriteLine("\nDanh sách tài khoản:");

foreach (var acc in AccList)

{

Console.WriteLine(acc);

}

}

public void SaveFile()

{

Console.Write("Nhập tên file để lưu: ");

string filename = Console.ReadLine();

try

{

using (FileStream fs = new FileStream(filename, FileMode.Create))

{

BinaryFormatter formatter = new BinaryFormatter();

formatter.Serialize(fs, AccList);

}

Console.WriteLine("Lưu file thành công!");

}

catch (Exception e)

{

Console.WriteLine("Lỗi khi lưu file: " + e.Message);

}

}

public void LoadFile()

{

Console.Write("Nhập tên file để tải: ");

string filename = Console.ReadLine();

if (!File.Exists(filename))

{

Console.WriteLine("File không tồn tại!");

return;

}

try

{

using (FileStream fs = new FileStream(filename, FileMode.Open))

{

BinaryFormatter formatter = new BinaryFormatter();

AccList = (List<Account>)formatter.Deserialize(fs);

}

Console.WriteLine("Load file thành công!");

ShowAccount();

}

catch (Exception e)

{

Console.WriteLine("❌ Lỗi khi tải file: " + e.Message);

}

}

public void RemoveAccount()

{

Console.Write("Nhập ID tài khoản cần xóa: ");

int id = int.Parse(Console.ReadLine());

// Sắp xếp danh sách trước khi BinarySearch

AccList.Sort(new AccountComparer());

// Tạo một đối tượng giả để tìm kiếm

Account temp = new Account(id, 0, "", "");

int index = AccList.BinarySearch(temp, new AccountComparer());

if (index >= 0)

{

AccList.RemoveAt(index);

Console.WriteLine($"Tài khoản {id} đã bị xóa.");

}

else

{

Console.WriteLine("Không tìm thấy tài khoản!");

}

}

public void SortAccounts()

{

AccList = AccList.OrderBy(a => a.AccountID)

.ThenBy(a => a.FirstName)

.ThenBy(a => a.Balance)

.ToList();

Console.WriteLine("Danh sách đã được sắp xếp!");

ShowAccount();

}

}

}

**AccountComparer.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace exercise4

{

internal class AccountComparer : IComparer<Account>

{

public int Compare(Account x, Account y)

{

return x.AccountID.CompareTo(y.AccountID);

}

}

}

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using exercise4;

namespace exercise4

{

internal class Program

{

static void Main(string[] args)

{

AccountList accountList = new AccountList();

//accountList.NewAccount(new Account(1, "Thanh", "Thảo", 400000));

//accountList.NewAccount(new Account(2, "Đào", "Giang", 20000));

//accountList.NewAccount(new Account(3, "Thanh", "hải", 50000));

//accountList.NewAccount(new Account(4, "Minh", "Đăng, 10000"));

Console.OutputEncoding = Encoding.UTF8;

int chon;

do

{

Console.WriteLine("\n1. Thêm nhân viên");

Console.WriteLine("2. Hiển thị danh sách");

Console.WriteLine("3. Lưu lại file");

Console.WriteLine("4. Tải file ");

Console.WriteLine("5. Xóa tài khoản");

Console.WriteLine("6. Sắp xếp danh sách");

Console.WriteLine("0. Thoát");

Console.Write("Chọn: ");

chon = int.Parse(Console.ReadLine());

switch (chon)

{

case 1: accountList.NewAccount(); break;

case 2: accountList.ShowAccount(); break;

case 3: accountList.SaveFile(); break;

case 4: accountList.LoadFile(); break;

case 5: accountList.RemoveAccount(); break;

case 6: accountList.SortAccounts(); break;

case 0: Console.WriteLine("Thoát."); break;

default: Console.WriteLine("Sai lựa chọn!"); break;

}

} while (chon != 0);

}

}

}

**A screenshot of a computer

AI-generated content may be incorrect.**

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.