|  |  |
| --- | --- |
|  |  |
| **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ SÀI GÒN**  **KHOA CÔNG NGHỆ THÔNG TIN**  **---------------------------** | |
| **BÁO CÁO TỔNG HỢP** | |
| **ĐỒ ÁN TIN HỌC** | |
| Đề tài nhóm: Quản lý cửa hàng bán sách (chỉ bán sách) | |
| ***Nhóm 2***  **HỌ TÊN SV : Trình Mỹ Duyên – DH51703270**  **Nguyen Ngoc Tien – DH51703554**  **Cao Lê Tuấn Vũ – DH51701727**  **Lê Văn Huy – DH51703477**  **Nguyễn Thị Thùy Nhung – DH51703896**  **LỚP : D17\_TH10**  **GVPT : GV.ThS Lương An Vinh** | |
|  | |
| **Tp.HCM – 12/2019** | |

**BÁO CÁO ĐỒ ÁN TIN HỌC**

***MỤC LỤC***

*Trang*

1. **Làm bài cá nhân**
2. Tuần 1

Mảng động 2 chiều 1

* Trình Mỹ Duyên 1
* Trương Ngọc Khánh 14
* Cao Lê Tuấn Vũ 21
* Lê Văn Huy 26
* Nguyễn Thị Thùy Nhung 34

1. Tuần 2

Danh sách liên kết đơn 40

* Trình Mỹ Duyên 40
* Trương Ngọc Khánh 53
* Cao Lê Tuấn Vũ 59
* Lê Văn Huy 66
* Nguyễn Thị Thùy Nhung 71

1. Tuần 3

Cây nhị phân tìm kiếm 75

* Trình Mỹ Duyên 76
* Trương Ngọc Khánh 84
* Cao Lê Tuấn Vũ 90
* Lê Văn Huy 96
* Nguyễn Thị Thùy Nhung 101

1. Tuần 4 – 5

Tìm hiểu kiểu dữ liệu Dictionary trong ngôn ngữ C# và ứng dụng vào việc xây dựng một chương trình quản lý sinh viên (trên console) 105

* Trình Mỹ Duyên 105
* Trương Ngọc Khánh 111
* Cao Lê Tuấn Vũ 116
* Lê Văn Huy 120
* Nguyễn Thị Thùy Nhung 123

1. Tuần 6

Tìm hiểu sự khác biệt giữa Hashtable, Dictionary và List. Khi nào thì dùng loại dữ liệu nào? 126

* Trình Mỹ Duyên 126
* Trương Ngọc Khánh 127
* Cao Lê Tuấn Vũ 129
* Lê Văn Huy 130
* Nguyễn Thị Thùy Nhung 131

1. **Làm bài nhóm –** Đề tài: Quản lý cửa hàng bán sách (chỉ bán sách)
2. Tuần 7

Liệt kê các loại đối tượng cần quản lý của đề tài và các thuộc tính của loại đối tượng đó 132

1. Tuần 8

Hoàn thiện các đối tượng cần quản lý và tiến hành code chương trình. Gửi mail phân công công việc cho Thầy 133

1. Tuần 9 – 10

Hoàn thiện code và viết báo cáo về Đồ án 134

1. **Phân công làm bài nhóm cụ thể 233**
2. **Lời cảm ơn 234**
3. **Làm bài cá nhân**
4. Tuần 1: Mảng động 2 chiều

- Khởi tạo mảng

- Thêm, xóa dòng

- Thêm, xóa phần tử trên dòng

- Tìm kiếm vị trí đầu tiên, cuối cùng

- Tìm kiếm tất cả các vị trí

- Xuất mảng

- Xuất mảng xoắn ốc

- Thoát

*Bài làm*

* Trình Mỹ Duyên

#include <iostream>

#include <string.h>

#include <math.h>

#include <stdlib.h>

#include <ctime>

using namespace std;

int\*\* KhoiTaoMang(int \*\*p, int dong, int cot)

{

p = new int\*[dong];

for (int i = 0; i < dong; i++)

p[i] = new int[cot];

return p;

}

void GiaiPhongMang(int \*\*p, int dong)

{

for (int i = 0; i < dong; i++)

delete[]p[i];

delete[]p;

}

void NhapMang(int \*\*p, int dong, int cot)

{

for (int i = 0; i < dong; i++)

{

for (int j = 0; j < cot; j++)

{

cout << "a[" << i + 1 << "][" << j + 1 << "]= ";

cin >> p[i][j];

}

//cout << endl;

}

}

void XuatMang(int \*\*p, int dong, int cot)

{

for (int i = 0; i < dong; i++)

{

for (int j = 0; j < cot; j++)

cout << "\t" << p[i][j];

cout << endl;

}

cout << endl;

}

void ThemDong(int \*\*p, int dong, int cot)

{

int\*\*p = new int \*[m];

int k = 0;

do

{

cout << "\nNhap dong muon them vao: ";

cin >> k;

} while (k <= 0 || k > dong + 1);

if (k >= 0 && k < m)

{

for (int i = m; i > d; i--)

for (int j = 0; j < n; j++)

p[i][j] = a[i - 1][j];

m++;

cout << "\nNhap gia tri them vao:" << endl;

for (int j = 0; j < n; j++)

{

cout << "a[" << k << "][" << j << "]:";

cin >> a[k][j];

}

}

return p;

}

void XoaDong(int \*\*p, int dong, int cot)

{

int k = 0;

do

{

cout << "\nNhap dong muon xoa vao: ";

cin >> k;

} while (k <= 0 || k > dong + 1);

for (int i = k; i < m - 1; i++)

{

for (int j = 0; j < n; j++)

a[i][j] = a[i + 1][j];

}

m--;

XuatMang(a, n, m);

}

void ThemPTuTrenDong(int \*\*p, int dong, int cot)

{

int ViTriThem, A[MAX], DongThem, NewLength;

cout << "\nNhap dong can them phan tu: ";

cin >> DongThem;

cout << "\nNhap vi tri can them phan tu: ";

cin >> ViTriThem;

for (int j = 0; j < cot; j++)

A[j] = a[DongThem][j];

for (int j = cot; j > ViTriThem; j--)

A[j] = A[j - 1];

cout << endl;

cout << "\nNhap gia tri muon them: ";

cin >> A[ViTriThem];

NewLength = cot + 1;

a[DongThem] = new int[NewLength];

for (int j = 0; j < NewLength; j++)

a[DongThem][j]=A[j];

DongThem++;

return a;

}

void XoaPTuTrenDong(int \*\*p, int dong, int cot)

{

int ViTriXoa, DongThem, NewLength;

cout << "\nNhap dong can xoa phan tu: ";

cin >> DongXoa;

cout << "\nNhap vi tri can xoa phan tu: ";

cin >> ViTriXoa;

for (int j = ViTriXoa; j < cot; j++)

a[DongXoa][j] = a[DongXoa][j+1];

NewLength = cot-1;

a[DongXoa] = new int[NewLength];

DongXoa++;

return a;

}

bool TimKiem(int \*\*a, int dong, int cot, int gt)

{

for (int i = 0; i < dong; i++)

for (int j = 0; j < cot; j++)

if (a[i][j] == gt)

return true;

return false;

}

void TimVTDau(int \*\*a, int dong, int cot)

{

int gt, v = 0, t = 0;

cout << "\nNhap gia tri muon tim: ";

cin >> gt;

for (int i = 0; i < dong; i++)

for (int j = 0; j < cot; j++)

if (TimKiem(a, dong, cot, gt) == true)

{

v = i;

t = j;

cout << "\Gia tri " << gt << " co vi tri dau tien: a[" << v + 1 << "][" << t + 1 << "]\n";

return;

}

else cout << "\nKhong co gia tri!";

}

void TimVTCuoi(int \*\*a, int dong, int cot)

{

int gt;

int v = 0, t = 0;

cout << "\nNhap gia tri muon tim: ";

cin >> gt;

for (int i = 0; i < dong; i++)

for (int j = 0; j < cot; j++)

if (TimKiem(a, dong, cot, gt) == true)

{

v = i;

t = j;

break;

}

if (v != 0 && t != 0)

cout << "\Gia tri " << gt << " co vi tri cuoi cung: a[" << v + 1 << "][" << t + 1 << "]\n";

else cout << "\nKhong co gia tri!";

}

void TimVTAll(int \*\*a, int dong, int cot)

{

int gt;

cout << "\nNhap gia tri muon tim: ";

cin >> gt;

cout << "\Gia tri " << gt << " co vi tri:";

for (int i = 0; i < dong; i++)

for (int j = 0; j < cot; j++)

if (a[i][j] == gt)

{

cout << "\ta[" << i + 1 << "," << j + 1 << "]";

}

else cout << "\nKhong co gia tri!";

cout << endl;

}

void TaoMangXoanOc(int \*\*a, int dong, int cot)

{

int A[100];

int dem = 0, trai = 0, tren = 0, phai = cot, duoi = dong, soluong = cot \* dong;

for (int i = 0; i < dong; i++)

for (int j = 0; j < dong; j++)

{

A[dem] = a[i][j];

dem++;

}

while (dem < soluong)

{

for (int i = trai; i < phai; i++)

if (dem < soluong)

{

a[tren][i] = A[dem];

dem++;

}

tren++;

for (int i = tren; i < duoi; i++)

if (dem < soluong)

{

a[i][phai - 1] = A[dem];

dem++;

}

phai--;

for (int i = phai - 1; i >= trai; i--)

if (dem < soluong)

{

a[duoi - 1][i] = A[dem];

dem++;

}

duoi--;

for (int i = duoi - 1; i >= tren; i--)

if (dem < soluong)

{

a[i][trai] = A[dem];

dem++;

}

trai++;

}

}

int main()

{

cout <<"==========================MENU============================";

cout << "\n0.Exit";

cout << "\n1.Nhap mang";

cout << "\n2.Xuat mang";

cout << "\n3.Them dong trong mang";

cout << "\n4.Xoa dong trong mang";

cout << "\n5.Them phan tu tren 1 dong";

cout << "\n6.Xoa phan tu ren 1 dong";

cout << "\n7.Tim kiem dau tien vi tri cua gia tri";

cout << "\n8.Tim kiem cuoi cung vi tri cua gia tri";

cout << "\n9.Tim kiem vi tri cua gia tri";

cout << "\n10.Xuat mang theo chieu xoan oc\n";

cout <<"===========================================================";

nhap:

int chon = 0;

do

{

cout << "\nChon Menu: ";

cin >> chon;

} while (chon < 0 || chon > 10);

int dong, cot;

int \*\*p = NULL;

int tt;

switch (chon)

{

case 1:

dong = 0, cot = 0;

do

{

if (dong <= 0)

{

cout << "\nNhap so dong: ";

cin >> dong;

}

else if (cot <= 0)

{

cout << "\nNhap so cot: ";

cin >> cot;

}

} while (dong <= 0 || cot <= 0);

cout << "\nNhap Ma Tran:\n";

p = KhoiTaoMang(p, dong, cot);

NhapMang(p, dong, cot);

cout << "\nBan co muon tiep tuc?(1 co, 2 khong): ";

cin >> tt;

if (tt == 1)

goto nhap;

break;

case 2:

cout << "\nMa Tran vua nhap:\n";

XuatMang(p, dong, cot);

cout << "\nBan co muon tiep tuc?(1 co, 2 khong): ";

cin >> tt;

if (tt == 1)

goto nhap;

break;

case 3:

ThemDong(p, dong, cot);

cout << "\nBan co muon tiep tuc?(1 co, 2 khong): ";

cin >> tt;

if (tt == 1)

goto nhap;

break;

case 4:

XoaDong(p, dong, cot);

cout << "\nBan co muon tiep tuc?(1 co, 2 khong): ";

cin >> tt;

if (tt == 1)

goto nhap;

break;

case 5:

ThemPTuTrenDong(p, dong, cot);

cout << "\nBan co muon tiep tuc?(1 co, 2 khong): ";

cin >> tt;

if (tt == 1)

goto nhap;

break;

case 6:

cout << "\nBan co muon tiep tuc?(1 co, 2 khong): ";

cin >> tt;

if (tt == 1)

goto nhap;

break;

case 7:

TimVTDau(p, dong, cot);

cout << "\nBan co muon tiep tuc?(1 co, 2 khong): ";

cin >> tt;

if (tt == 1)

goto nhap;

break;

case 8:

TimVTCuoi(p, dong, cot);

cout << "\nBan co muon tiep tuc?(1 co, 2 khong): ";

cin >> tt;

if (tt == 1)

goto nhap;

break;

case 9:

TimVTAll(p, dong, cot);

cout << "\nBan co muon tiep tuc?(1 co, 2 khong): ";

cin >> tt;

if (tt == 1)

goto nhap;

break;

case 10:

TaoMangXoanOc(p, dong, cot);

cout << "\nXuat Ma Tran xoan oc:\n";

XuatMang(p, dong, cot);

cout << "\nBan co muon tiep tuc?(1 co, 2 khong): ";

cin >> tt;

if (tt == 1)

goto nhap;

break;

case 0:

cout << "\nThoat!!!\n";

break;

default: break;

}

system("pause");

return 0;

}

\*

\* \*

* Trương Ngọc Khánh

#include<iostream>

using namespace std;

const int Max = 50;

int\*\*KhoitaoMang(int\*\*a, int&n, int&m);

int\*\*NhapMang(int\*\*a, int n, int m);

void XuatMang(int\*\*a, int n, int m);

int Timvitri(int\*\*a, int n, int m, int &x);

int Timvitridau(int\*\*a, int n, int m, int &y);

int Timvitricuoi(int \*\*a, int n, int m, int &s);

void XoaDong(int\*\*a, int n, int m);

void ThemDong(int\*\*a, int n, int m)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Source

#include"Header.h"

int\*\*KhoitaoMang(int\*\*a, int&n, int&m)

{

cout << "So dong cua Mang:";

cin >> m;

cout << "So cot cua Mang";

cin >> n;

a = new int \*[m];

for (int i = 0; i < m; i++)

a[i] = new int[n];

return a;

}

int\*\*NhapMang(int\*\*a, int n, int m)

{

for (int i = 0; i < m; i++)

{

for (int j = 0; j < n; j++)

{

cout << "a[" << i <<"]["<< j << "]:";

cin >> a[i][j];

}

}

return a;

}

void XuatMang(int\*\*a, int n, int m)

{

for (int i = 0; i < m; i++)

{

for (int j = 0; j < n; j++)

cout << a[i][j] << " ";

cout <<endl;

}

}

void ThemDong(int\*\*a, int n, int m)

{

int\*\*p = new int \*[m];

int d;

cout << "Dong can chen:";

cin >> d;

if (d >= 0 && d < m)

{

for (int i = m; i > d; i--)

{

for (int j = 0; j < n; j++)

{

p[i][j] = a[i - 1][j];

}

}m++;

cout << "Gia tri dong chen vao:" << endl;

for (int j = 0; j < n; j++)

{

cout << "a[" << d << "][" << j << "]:";

cin >> a[d][j];

}

}

return p;

}

void XoaDong(int\*\*a, int n, int m)

{

int dong;

cout << "dong can xoa:";

cin >> dong;

for (int i = dong; i < m - 1; i++)

{

for (int j = 0; j < n; j++)

a[i][j] = a[i + 1][j];

}

m--;

XuatMang(a, n, m);

}

int Timvitri(int\*\*a, int n, int m, int &x)

{

cout << "Vi tri can tim:";

cin >> x;

for (int i = 0; i < m; i++)

for (int j = 0; j < n; j++)

{

if (x == a[i][j])

{

cout << "Vi tri la:[" << i << "][" << j << "]";

cout << endl;

}

}

return 0;

}

int Timvitridau(int\*\*a, int n, int m, int &y)

{

cout << "Vi tri can tim:";

cin >> y;

for (int i = 0; i < m; i++)

{

for (int j = 0; j < n; j++)

{

if (y == a[i][j])

{

cout << "Vi tri la:[" << i << "][" << j << "]";

cout << endl;

return a[i][j];

break;

}

}

}

return 0;

}

int Timvitricuoi(int \*\*a, int n, int m, int &s)

{

cout << "Vi tri can tim:";

cin >> s;

for (int i = m-1; i >=0 ;i--)

{

for (int j = n-1; j>=0; j--)

{

if (s == a[i][j])

{

cout << "Vi tri la:[" << i << "][" << j << "]";

cout << endl;

return a[i][j];

break;

}

}

}

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Main

#include "Header.h"

void main()

{

int m;int n;

int\*\*a = NULL;

int\*\*p = NULL;

int x;

int y;

int s;

a = KhoitaoMang(a, n, m);

NhapMang(a, n,m);

XuatMang(a, n, m);

int u = Timvitri(a, n, m, x);

int z = Timvitridau(a, n, m, y);

int c = Timvitricuoi(a, n, m, s);

ThemDong(a, n, m);

/\*XoaDong(a, n, m);\*/

for (int i = 0; i < m; i++)

{

delete[]a[i];

}delete[]a;

system("pause");

}

* Cao Lê Tuấn Vũ

#include "Header.h"

int\*\* KhoiTaoMang(int\*\* a, int& m, int& n)

{

cout << "Nhap so dong: "; cin >> m;

cout << "Nhap so cot: "; cin >> n;

a = new int\* [m];

for (int i = 0; i < m; i++)

a[i] = new int[n];

return a;

}

int\*\* NhapMang(int\*\* a, int m, int n)

{

for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++) {

cout << "a[" << i << "][" << j << "]=";

cin >> a[i][j];

cout << endl;

}

}

return a;

}

void XuatMang(int\*\* a, int m, int n) {

cout << "Ma Tran La: " << endl;

for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++)

cout << a[i][j] << "\t";

cout << endl;

}

cout << endl;

}

int\*\* ThemDong(int\*\* a, int& m, int &n)

{

int k;

cout << "Nhap dong can them vao: "; cin >> k;

int\*\* b = NULL;

b = new int\* [m + 1];

for (int i = 0; i < m + 1; i++)

b[i] = new int[n];

for (int i = k; i < m; i++)

for (int j = 0; j < n; j++)

b[i + 1][j] = a[i][j];

for (int i = 0; i < m; i++)

for (int j = 0; j < n; j++)

if (i == k) {

cout << "a[" << i << "][" << j << "]=";

cin >> a[i][j];

}

for (int i = 0; i < k; i++)

for (int j = 0; j < n; j++)

b[i][j] = a[i][j];

m++;

return b;

}

void XoaDong(int\*\*a, int m, int n)

{

int dong;

cout << "dong can xoa:";

cin >> dong;

for (int i = dong; i < m-1; i++)

{

for(int j=0;j<n;j++)

a[i][j] = a[i +1][j];

}

m--;

XuatMang(a, m,n);

}

int viTriBatKi(int\*\* a, int m, int n, int& x)

{

cout << "Vi tri phan tu can tim la:";

cin >> x;

for (int i = 0; i < m; i++)

{

for (int j = 0; j < n; j++)

if (a[i][j] == x)

{

cout << "vi tri la:[" << i << "][" << j << "]";

cout << endl;

}

}

return 0;

}

int vitridau(int\*\* a, int m, int n, int& y)

{

cout << "Vi tri phan tu can tim la:";

cin >> y;

for (int i = 0; i < m; i++)

{

for (int j = 0; j < n; j++)

if (a[i][j] == y)

{

cout << "vi tri dau tien la:[" << i << "][" << j << "]";

cout << endl;

return a[i][j];

break;

}

}

return 0;

}

int vitricuoi(int\*\* a, int m, int n, int& z)

{

cout << "Vi tri phan tu can tim la:";

cin >> z;

for (int i = m-1; i >=0; i--)

{

for (int j = n-1; j >=0; j--)

if (a[i][j] == z)

{

cout << "vi tri cuoi cung la:[" << i << "][" << j << "]";

cout << endl;

return a[i][j];

break;

}

}

return 0;

}

void themPhanTu(int\*\* a, int& m, int& n)

{

int vtd;

int vtc;

int x;

cout << "Moi ban nhap dong can them pt:";

cin >> vtd;

cout << "Moi ban cap nhap cot can them phan tu:";

cin >> vtc;

cout << "a[" << vtd << "][" << vtc << "]:";

cin >> a[vtd][vtc];

cout << endl;

XuatMang(a, m, n);

}

void xoaPhanTu(int\*\* a, int& m, int& n)

{

int vtd;

int vtc;

int x;

cout << "Moi ban nhap dong can them pt:";

cin >> vtd;

cout << "Moi ban cap nhap cot can them phan tu:";

cin >> vtc;

cout << "a[" << vtd << "][" << vtc << "]:";

a[vtd][vtc] = 0;

cout << endl;

XuatMang(a, m, n);

}

\*

\* \*

* Lê Văn Huy

\*Header.h

#include<iostream>

using namespce std;

int\*\* KhoiTaoMang(int\*\* a, int& m, int& n);

int\*\* NhapMang(int\*\* a, int m, int n);

void XuatMang(int\*\* a, int m, int n);

void XoaDong(int\*\*a, int m, int n);

int\*\* ThemDong(int\*\* a, int& m, int &n);

int viTriBatKi(int\*\* a, int m, int n, int& x);

int vitridau(int\*\* a, int m, int n, int& y);

int vitricuoi(int\*\* a, int m, int n, int& z);

void themPhanTu(int\*\* a, int& m, int& n);

void xoaPhanTu(int\*\* a, int& m, int& n)

-----------------------------------------------------------

\*Suorce.cpp

#include "Header.h"

int\*\* KhoiTaoMang(int\*\* a, int& m, int& n)

{

cout << "Nhap so dong: "; cin >> m;

cout << "Nhap so cot: "; cin >> n;

a = new int\* [m];

for (int i = 0; i < m; i++)

a[i] = new int[n];

return a;

}

int\*\* NhapMang(int\*\* a, int m, int n)

{

for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++) {

cout << "a[" << i << "][" << j << "]=";

cin >> a[i][j];

cout << endl;

}

}

return a;

}

void XuatMang(int\*\* a, int m, int n) {

cout << "Ma Tran La: " << endl;

for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++)

cout << a[i][j] << "\t";

cout << endl;

}

cout << endl;

}

int\*\* ThemDong(int\*\* a, int& m, int &n)

{

int k;

cout << "Nhap dong can them vao: "; cin >> k;

int\*\* b = NULL;

b = new int\* [m + 1];

for (int i = 0; i < m + 1; i++)

b[i] = new int[n];

for (int i = k; i < m; i++)

for (int j = 0; j < n; j++)

b[i + 1][j] = a[i][j];

for (int i = 0; i < m; i++)

for (int j = 0; j < n; j++)

if (i == k) {

cout << "a[" << i << "][" << j << "]=";

cin >> a[i][j];

}

for (int i = 0; i < k; i++)

for (int j = 0; j < n; j++)

b[i][j] = a[i][j];

m++;

return b;

}

void XoaDong(int\*\*a, int m, int n)

{

int dong;

cout << "dong can xoa:";

cin >> dong;

for (int i = dong; i < m-1; i++)

{

for(int j=0;j<n;j++)

a[i][j] = a[i +1][j];

}

m--;

XuatMang(a, m,n);

}

int viTriBatKi(int\*\* a, int m, int n, int& x)

{

cout << "Vi tri phan tu can tim la:";

cin >> x;

for (int i = 0; i < m; i++)

{

for (int j = 0; j < n; j++)

if (a[i][j] == x)

{

cout << "vi tri la:[" << i << "][" << j << "]";

cout << endl;

}

}

return 0;

}

int vitridau(int\*\* a, int m, int n, int& y)

{

cout << "Vi tri phan tu can tim la:";

cin >> y;

for (int i = 0; i < m; i++)

{

for (int j = 0; j < n; j++)

if (a[i][j] == y)

{

cout << "vi tri dau tien la:[" << i << "][" << j << "]";

cout << endl;

return a[i][j];

break;

}

}

return 0;

}

int vitricuoi(int\*\* a, int m, int n, int& z)

{

cout << "Vi tri phan tu can tim la:";

cin >> z;

for (int i = m-1; i >=0; i--)

{

for (int j = n-1; j >=0; j--)

if (a[i][j] == z)

{

cout << "vi tri cuoi cung la:[" << i << "][" << j << "]";

cout << endl;

return a[i][j];

break;

}

}

return 0;

}

void themPhanTu(int\*\* a, int& m, int& n)

{

int vtd;

int vtc;

int x;

cout << "Moi ban nhap dong can them pt:";

cin >> vtd;

cout << "Moi ban cap nhap cot can them phan tu:";

cin >> vtc;

cout << "a[" << vtd << "][" << vtc << "]:";

cin >> a[vtd][vtc];

cout << endl;

XuatMang(a, m, n);

}

void xoaPhanTu(int\*\* a, int& m, int& n)

{

int vtd;

int vtc;

int x;

cout << "Moi ban nhap dong can them pt:";

cin >> vtd;

cout << "Moi ban cap nhap cot can them phan tu:";

cin >> vtc;

cout << "a[" << vtd << "][" << vtc << "]:";

a[vtd][vtc] = 0;

cout << endl;

XuatMang(a, m, n);

}

----------------------------------------------------------------------------

\*Main.cpp

#include "Header.h"

void main()

{

int m, n;

int x;

int y;

int z;

int\*\* a = NULL;

a = KhoiTaoMang(a, m, n);

NhapMang(a, m, n);

XuatMang(a, m, n);

ThemDong(a, m, n);

XoaDong(a, m, n);

int u = viTriBatKi(a, m, n, x);

int o = vitridau(a, m,n, y);

int p = vitricuoi(a, n, m, z);

themPhanTu(a, m, n);

xoaPhanTu(a, m, n);

for (int i = 0; i < m; i++)

delete[]a[i];

delete[]a;

system("pause");

}

* Nguyễn Thị Thùy Nhung

#include<iostream>

using namespce std;

int\*\* KhoiTaoMang(int\*\* a, int& m, int& n)

{

cout << "Nhap so dong: "; cin >> m;

cout << "Nhap so cot: "; cin >> n;

a = new int\* [m];

for (int i = 0; i < m; i++)

a[i] = new int[n];

return a;

}

int\*\* NhapMang(int\*\* a, int m, int n)

{

for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++) {

cout << "a[" << i << "][" << j << "]=";

cin >> a[i][j];

cout << endl;

}

}

return a;

}

void XuatMang(int\*\* a, int m, int n) {

cout << "Ma Tran La: " << endl;

for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++)

cout << a[i][j] << "\t";

cout << endl;

}

cout << endl;

}

int\*\* ThemDong(int\*\* a, int& m, int &n)

{

int k;

cout << "Nhap dong can them vao: "; cin >> k;

int\*\* b = NULL;

b = new int\* [m + 1];

for (int i = 0; i < m + 1; i++)

b[i] = new int[n];

for (int i = k; i < m; i++)

for (int j = 0; j < n; j++)

b[i + 1][j] = a[i][j];

for (int i = 0; i < m; i++)

for (int j = 0; j < n; j++)

if (i == k) {

cout << "a[" << i << "][" << j << "]=";

cin >> a[i][j];

}

for (int i = 0; i < k; i++)

for (int j = 0; j < n; j++)

b[i][j] = a[i][j];

m++;

return b;

}

void XoaDong(int\*\*a, int m, int n)

{

int dong;

cout << "dong can xoa:";

cin >> dong;

for (int i = dong; i < m-1; i++)

{

for(int j=0;j<n;j++)

a[i][j] = a[i +1][j];

}

m--;

XuatMang(a, m,n);

}

int viTriBatKi(int\*\* a, int m, int n, int& x)

{

cout << "Vi tri phan tu can tim la:";

cin >> x;

for (int i = 0; i < m; i++)

{

for (int j = 0; j < n; j++)

if (a[i][j] == x)

{

cout << "vi tri la:[" << i << "][" << j << "]";

cout << endl;

}

}

return 0;

}

int vitridau(int\*\* a, int m, int n, int& y)

{

cout << "Vi tri phan tu can tim la:";

cin >> y;

for (int i = 0; i < m; i++)

{

for (int j = 0; j < n; j++)

if (a[i][j] == y)

{

cout << "vi tri dau tien la:[" << i << "][" << j << "]";

cout << endl;

return a[i][j];

break;

}

}

return 0;

}

int vitricuoi(int\*\* a, int m, int n, int& z)

{

cout << "Vi tri phan tu can tim la:";

cin >> z;

for (int i = m-1; i >=0; i--)

{

for (int j = n-1; j >=0; j--)

if (a[i][j] == z)

{

cout << "vi tri cuoi cung la:[" << i << "][" << j << "]";

cout << endl;

return a[i][j];

break;

}

}

return 0;

}

void themPhanTu(int\*\* a, int& m, int& n)

{

int vtd;

int vtc;

int x;

cout << "Moi ban nhap dong can them pt:";

cin >> vtd;

cout << "Moi ban cap nhap cot can them phan tu:";

cin >> vtc;

cout << "a[" << vtd << "][" << vtc << "]:";

cin >> a[vtd][vtc];

cout << endl;

XuatMang(a, m, n);

}

void xoaPhanTu(int\*\* a, int& m, int& n)

{

int vtd;

int vtc;

int x;

cout << "Moi ban nhap dong can them pt:";

cin >> vtd;

cout << "Moi ban cap nhap cot can them phan tu:";

cin >> vtc;

cout << "a[" << vtd << "][" << vtc << "]:";

a[vtd][vtc] = 0;

cout << endl;

XuatMang(a, m, n);

}

void main()

{

int m, n;

int x;

int y;

int z;

int\*\* a = NULL;

a = KhoiTaoMang(a, m, n);

NhapMang(a, m, n);

XuatMang(a, m, n);

ThemDong(a, m, n);

XoaDong(a, m, n);

int u = viTriBatKi(a, m, n, x);

int o = vitridau(a, m,n, y);

int p = vitricuoi(a, n, m, z);

themPhanTu(a, m, n);

xoaPhanTu(a, m, n);

for (int i = 0; i < m; i++)

delete[]a[i];

delete[]a;

system("pause");

}

1. Tuần 2: Danh sách liên kết đơn

- Khởi tạo danh sách (nhập số phần tử rồi nhập các phần tử)

- Xuất danh sách

- Chèn đầu

- Chèn giữa

- Chèn cuối

- Xóa đầu

- Xóa giữa

- Xóa cuối

- Đảo vị trí 2 node chỉ định

*Bài làm*

* Trình Mỹ Duyên

#include <iostream>

#include <stdio.h>

#include <conio.h>

#include <iomanip>

#include <cstdlib>

using namespace std;

typedef int TYPEINFO;

struct NODE

{

TYPEINFO data;

NODE \*link;

};

typedef NODE \*NODEPTR;

//Khoi tao DSLK

void Init(NODEPTR &list)

{

list = NULL;

}

//Kiem tra rong?

int Empty(NODEPTR list)

{

return (list == NULL ? 1 : 0);

}

//Tao mot nut (phan tu)

NODEPTR TaoNODE(TYPEINFO x)

{

NODEPTR p;

p = new NODE;

p->data = x;

p->link = NULL;

return p;

}

//Giai phong bo nho

void GiaiPhong(NODEPTR &list)

{

NODEPTR p = list;

while (p != NULL)

{

list = list->link;

delete p;

p = list;

}

}

//Dem phan tu trong DSLK

int DemPT(NODEPTR list)

{

int dem = 0;

while (list != NULL)

{

dem++;

list = list->link;

}

return dem;

}

//Them 1 phan tu vao dau DSLK

NODEPTR ThemDau(NODEPTR &list, TYPEINFO x)

{

NODEPTR p = TaoNODE(x);

if (list == NULL)

list = p;

else

{

p->link = list;

list = p;

}

return p;

}

//Them 1 phan tu vao giua DSLK

NODEPTR ThemGiua(NODEPTR &list, TYPEINFO x, int vt)

{

NODEPTR p = TaoNODE(x);

int dem = DemPT(list);

if (vt<1 || vt>dem)

cout << "\nVi tri khong thoa!";

else

if (vt == 1)

ThemDau(list, x);

else

{

NODEPTR q = list;

int dem = 1;

while (dem != vt - 1 && p != NULL)

{

dem++;

q = q->link;

}

p->link = q->link;

q->link = p;

}

return p;

}

//Them 1 phan tu vao cuoi DSLK

NODEPTR ThemCuoi(NODEPTR &list, TYPEINFO x)

{

NODEPTR p = TaoNODE(x);

if (list == NULL)

list = p;

else

{

NODEPTR q = list;

while (q->link != NULL)

q = q->link;

q->link = p;

}

return p;

}

void NhapDS(NODEPTR &list, int n)

{

Init(list);

int x;

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

{

cout << "\nNhap x" << i + 1 << ": ";

cin >> x;

//ThemDau(list, x);

ThemCuoi(list, x);

}

}

void XuatDS(NODEPTR list)

{

NODEPTR p = list;

while (p != NULL)

{

cout << "\t" << p->data;

p = p->link;

}

cout << endl;

}

//Tim phan tu co gia tri x

NODEPTR TimNODEX(NODEPTR &list, TYPEINFO x)

{

NODEPTR p = list;

while (p != NULL)

{

if (p->data == x)

return p;

p = p->link;

}

return NULL;

}

//Xoa 1 phan tu o dau DSLK

NODEPTR XoaDau(NODEPTR &list)

{

NODEPTR p = list;

if (p->link == NULL)

{

delete p;

list = NULL;

}

else

{

p = list;

list = p->link;

delete p;

}

return p;

}

//Xoa phan tu co gia tri x

NODEPTR XoaGiua(NODEPTR &list, TYPEINFO x)

{

NODEPTR p, q;

if (TimNODEX(list, x) == NULL)

cout << "\nKhong co gia tri can xoa!" << endl;

else

if (list->data == x)

XoaDau(list);

else

{

p = list;

q = list;

while (p != NULL && p->data != x)

{

q = p;

p = p->link;

}

if (p->data != NULL && p->data == x)

{

q->link = p->link;

delete p;

}

}

return 0;

}

//Xoa 1 phan tu o cuoi DSLK

NODEPTR XoaCuoi(NODEPTR &list)

{

NODEPTR p = list;

if (p->link == NULL)

{

delete p;

list = NULL;

}

else

{

while (p->link->link != NULL)

p = p->link;

p->link = NULL;

}

return p;

}

//Dao nguoc 2 node co gia tri chi dinh

void DaoNguocGiaTri(NODEPTR &list, int x, int y)

{

if (TimNODEX(list, x) == NULL || TimNODEX(list, y) == NULL)

cout << "\nGia tri khong hop le!" << endl;

else

{

NODEPTR p = NULL;

NODEPTR q = NULL;

int tam;

while (list != NULL)

{

if (TimNODEX(list, x) != NULL)

p->data = x;

else

if (TimNODEX(list, y) != NULL)

q->data = y;

if (p != NULL && q != NULL)

{

tam = p->data;

p->data = q->data;

q->data = tam;

}

list = list->link;

}

}

cout << "\nDSLK sau khi dao phan tu chi dinh: ";

XuatDS(list);

}

//Dao nguoc 2 node co vi tri chi dinh

void DaoNguocViTri(NODEPTR &list, int x, int y)

{

if (x > DemPT(list) || y > DemPT(list))

cout << "\nVi tri khong hop le!" << endl;

else

{

NODEPTR p = list;

int dem = 0, tam, tam1, tam2;

while (p != NULL)

{

dem++;

if (dem == x)

{

tam1 = p->data;

}

else if (dem == y)

{

tam2 = p->data;

}

tam = tam1;

tam1 = tam2;

tam2 = tam;

p = p->link;

}

}

cout << "\nDSLK sau khi dao phan tu chi dinh: ";

XuatDS(list);

}

int main()

{

NODEPTR list;

int n;

cout << "\nNhap so phan tu: ";

cin >> n;

Init(list);

NhapDS(list, n);

system("cls");

cout << "\nDSLK vua nhap:";

XuatDS(list);

TYPEINFO x;

//int vt;

if (Empty(list))

cout << "\nDanh sach rong!" << endl;

else

{

cout << "\nNhap gia tri muon chen: ";

cin >> x;

ThemDau(list, x);

cout << "\nDSLK sau khi chen dau: ";

XuatDS(list);

cout << "\nNhap gia tri muon chen: ";

cin >> x;

cout << "\nNhap vi tri muon chen: ";

cin >> vt;

ThemGiua(list, x, vt);

cout << "\nDSLK sau khi chen giua: ";

XuatDS(list);

cout << "\nNhap gia tri muon chen: ";

cin >> x;

ThemCuoi(list, x);

cout << "\nDSLK sau khi chen cuoi: ";

XuatDS(list);

XoaDau(list);

cout << "\nDSLK sau khi xoa phan tu dau: ";

XuatDS(list);

cout << "\nNhap gia tri muon xoa: ";

cin >> x;

XoaGiua(list, x);

cout << "\nDSLK sau khi xoa " << x << " : ";

XuatDS(list);

XoaCuoi(list);

cout << "\nDSLK sau khi xoa phan tu cuoi: ";

XuatDS(list);

int a, b;

cout << "\nNhap gia tri dau muon dao: ";

cin >> a;

cout << "\nNhap gia tri sau: ";

cin >> b;

DaoNguocGiaTri(list, a, b);

cout << "\nNhap vi tri dau muon dao: ";

cin >> a;

cout << "\nNhap vi tri sau: ";

cin >> b;

DaoNguocViTri(list, a, b);

}

GiaiPhong(list);

system("pause");

return 0;

}

\*

\* \*

* Trương Ngọc Khánh

include"Header.h"

void Init(NODEPTR& list)

{

list = NULL;

}

int empty(NODEPTR list)

{

return (list == NULL ? 1 : 0);

}

int ktgiua(NODEPTR list)

{

int dem = 0;

for (NODEPTR p = list; p != NULL; p = p->link)

{

dem++;

}

return dem;

}

NODEPTR tao\_node(TYPEINFO x)

{

NODEPTR p;

p = new NODE;

p->data = x;

p->link = NULL;

return p;

}

NODEPTR Them\_dau(NODEPTR& list, TYPEINFO x)

{

NODEPTR p;

p = tao\_node(x);

p->link = list;

list = p;

return p;

}

NODEPTR Them\_giua(NODEPTR& list, int x, int n)

{

int m;

//int dem = ktgiua(list);

NODEPTR p = list;

//cout << "\nNhap data can them:";

//cin >> p->data;

//if (dem == n / 2)

//{

// p->link = q->link;

// q->link = p;

//}

//else

//{

// q = q->link;

//}

cout << endl;

cout << "Chon vi tri can them: "; cin >> m;

for (int i = 0; i < m; i++) {

p = p->link;

}

p = p->link;

p = tao\_node(x);

p = list;

return p;

}

void Nhap\_ds(NODEPTR& list)

{

Init(list);

int n;

int x;

cout << "Nhap so ptu:";

cin >> n;

for (int i = 1; i <=n; i++)

{

cout << "\nNhap ptu can them:";

cin >> x;

Them\_cuoi(list, x);

}

}

void Xuat\_ds(NODEPTR list)

{

NODEPTR p = list;

while (p!=NULL)

{

cout << p->data << " ";

p = p->link;

}

}

NODEPTR Them\_cuoi(NODEPTR& list, TYPEINFO x)

{

NODEPTR p;

p = tao\_node(x);

if (list == NULL)

list = p;

else

{

NODEPTR q = list;

while (q->link != NULL)

q = q->link;

q->link = p;

}

return p;

Xuat\_ds(list);

}

void Xoa\_Dau(NODEPTR& list)

{

NODEPTR p;

if (list != NULL)

{

p = list;

list = p->link;

delete p;

}

}

void Xoa\_Cuoi(NODEPTR& list)

{

NODEPTR p, before;

p = list;

before = list;

if (p->link != NULL)

{

delete p;

list = NULL;

}

else

while (p->link != NULL) {

}

}

+++++++++++++++++++++++++++++++++++++++

struct NODE

{

TYPEINFO data;

NODE\* link;

};

++++++++++++++++++++++++++++++++++++++++

#include"Header.h"

void main()

{

NODEPTR list;

int n=0;

int x=0;

Init(list);

Nhap\_ds(list);

cout << "\nDanh sach phan tu can nhap:";

Xuat\_ds(list);

Them\_giua(list, 2, n);

Xuat\_ds(list);

system("pause");

}

\*

\* \*

* Cao Lê Tuấn Vũ

#include "Header.h"

void main()

{

int m, n;

int x;

int y;

int z;

int\*\* a = NULL;

a = KhoiTaoMang(a, m, n);

NhapMang(a, m, n);

XuatMang(a, m, n);

ThemDong(a, m, n);

XoaDong(a, m, n);

int u = viTriBatKi(a, m, n, x);

int o = vitridau(a, m,n, y);

int p = vitricuoi(a, n, m, z);

themPhanTu(a, m, n);

xoaPhanTu(a, m, n);

for (int i = 0; i < m; i++)

delete[]a[i];

delete[]a;

system("pause");

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include"Header.h"

void Init(NODEPTR& list)

{

list = NULL;

}

int empty(NODEPTR list)

{

return (list == NULL ? 1 : 0);

}

int ktgiua(NODEPTR list)

{

int dem = 0;

for (NODEPTR p = list; p != NULL; p = p->link)

{

dem++;

}

return dem;

}

NODEPTR tao\_node(TYPEINFO x)

{

NODEPTR p;

p = new NODE;

p->data = x;

p->link = NULL;

return p;

}

NODEPTR Them\_dau(NODEPTR& list, TYPEINFO x)

{

NODEPTR p;

p = tao\_node(x);

p->link = list;

list = p;

return p;

}

NODEPTR Them\_giua(NODEPTR& list, int x, int n)

{

int m;

//int dem = ktgiua(list);

NODEPTR p = list;

//cout << "\nNhap data can them:";

//cin >> p->data;

//if (dem == n / 2)

//{

// p->link = q->link;

// q->link = p;

//}

//else

//{

// q = q->link;

//}

cout << endl;

cout << "Chon vi tri can them: "; cin >> m;

for (int i = 0; i < m; i++) {

p = p->link;

}

p = p->link;

p = tao\_node(x);

p = list;

return p;

}

void Nhap\_ds(NODEPTR& list)

{

Init(list);

int n;

int x;

cout << "Nhap so ptu:";

cin >> n;

for (int i = 1; i <=n; i++)

{

cout << "\nNhap ptu can them:";

cin >> x;

Them\_cuoi(list, x);

}

}

void Xuat\_ds(NODEPTR list)

{

NODEPTR p = list;

while (p!=NULL)

{

cout << p->data << " ";

p = p->link;

}

}

NODEPTR Them\_cuoi(NODEPTR& list, TYPEINFO x)

{

NODEPTR p;

p = tao\_node(x);

if (list == NULL)

list = p;

else

{

NODEPTR q = list;

while (q->link != NULL)

q = q->link;

q->link = p;

}

return p;

Xuat\_ds(list);

}

void Xoa\_Dau(NODEPTR& list)

{

NODEPTR p;

if (list != NULL)

{

p = list;

list = p->link;

delete p;

}

}

void Xoa\_Cuoi(NODEPTR& list)

{

NODEPTR p, before;

p = list;

before = list;

if (p->link != NULL)

{

delete p;

list = NULL;

}

else

while (p->link != NULL) {

}

}

+++++++++++++++++++++++++++++++++++++++

struct NODE

{

TYPEINFO data;

NODE\* link;

};

++++++++++++++++++++++++++++++++++++++++

#include"Header.h"

void main()

{

NODEPTR list;

int n=0;

int x=0;

Init(list);

Nhap\_ds(list);

cout << "\nDanh sach phan tu can nhap:";

Xuat\_ds(list);

Them\_giua(list, 2, n);

Xuat\_ds(list);

system("pause");

}

\*

\* \*

* Lê Văn Huy

\*Header.h

#include<iostream>

using namespace std;

struct Node

{

int data;

struct Node\* pNext;

};

typedef struct Node NODE;

struct List

{

NODE\* pDau, \* pCuoi;

};

typedef struct List LIST;

void KhoiTao(LIST& l);

NODE\* TaoNode(int DATA);

void InPut(LIST& l);

void OutPut(LIST l);

void ThemDau(LIST& l, NODE\* p);

void ThemCuoi(LIST& l, NODE\* p);

void XoaDau(LIST& l);

void XoaCuoi(LIST& l);

-----------------------------------------------------

\*Source.cpp

#include"Header.h"

void KhoiTao(LIST& l)

{

l.pDau = l.pCuoi = NULL;

}

NODE\* TaoNode(int DATA)

{

NODE\* p = new NODE;

if (p == NULL)

{

cout << "\nKhong du bo nho de cap phat con tro";

return NULL;

}

p->data = DATA;

p->pNext = NULL;

return p;

}

void InPut(LIST & l)

{

KhoiTao(l);

int n;

cout << "Nhap so luong phan tu vao danh sach:";

cin >> n;

for (int i = 1; i <= n; i++)

{

int data;

cout << "Nhap phan tu:";

cin >> data;

NODE\* p;

p = TaoNode(data);

ThemDau(l, p);

ThemCuoi(l, p);

}

}

void OutPut(LIST l)

{

for (NODE\* p = l.pDau; p != NULL; p = p->pNext)

{

cout << " ";

p->data;

}

}

void ThemDau(LIST& l, NODE\* p)

{

if (l.pDau == NULL)

{

l.pDau = l.pCuoi= p;

}

else

{

p->pNext = l.pDau;

l.pDau = p;

}

}

void ThemCuoi(LIST& l, NODE\* p)

{

if (l.pDau == NULL)

{

l.pDau = l.pCuoi = p;

}

else

{

l.pCuoi->pNext = p;

l.pCuoi = p;

}

}

void XoaDau(LIST& l)

{

if (l.pDau != NULL)

{

NODE\* p = l.pDau;

l.pDau = l.pDau->pNext;

delete p;

}

}

void XoaCuoi(LIST &l)

{

NODE\* g;

for (NODE\* p = l.pDau; p != NULL; p = p->pNext)

{

if (p == l.pCuoi)

{

g->pNext = NULL;

l.pCuoi = g;

delete p;

return;

}

g = p;

}

}

----------------------------------------------

\*Main.cpp

#include"Header.h"

int main()

{

LIST l;

NODE\* p;

InPut(l);

OutPut(l);

ThemDau(l,p);

ThemCuoi(l,p);

cout << "Xoa dau danh sach";

XoaDau(l);

cout << "Xoa cuoi danh sach:";

XoaCuoi(l);

system("pause");

}

\*

\* \*

* Nguyễn Thị Thùy Nhung

#include<iostream>

using namespace std;

void KhoiTao(LIST& l)

{

l.pDau = l.pCuoi = NULL;

}

NODE\* TaoNode(int DATA)

{

NODE\* p = new NODE;

if (p == NULL)

{

cout << "\nKhong du bo nho de cap phat con tro";

return NULL;

}

p->data = DATA;

p->pNext = NULL;

return p;

}

void InPut(LIST & l)

{

KhoiTao(l);

int n;

cout << "Nhap so luong phan tu vao danh sach:";

cin >> n;

for (int i = 1; i <= n; i++)

{

int data;

cout << "Nhap phan tu:";

cin >> data;

NODE\* p;

p = TaoNode(data);

ThemDau(l, p);

ThemCuoi(l, p);

}

}

void OutPut(LIST l)

{

for (NODE\* p = l.pDau; p != NULL; p = p->pNext)

{

cout << " ";

p->data;

}

}

void ThemDau(LIST& l, NODE\* p)

{

if (l.pDau == NULL)

{

l.pDau = l.pCuoi= p;

}

else

{

p->pNext = l.pDau;

l.pDau = p;

}

}

void ThemCuoi(LIST& l, NODE\* p)

{

if (l.pDau == NULL)

{

l.pDau = l.pCuoi = p;

}

else

{

l.pCuoi->pNext = p;

l.pCuoi = p;

}

}

void XoaDau(LIST& l)

{

if (l.pDau != NULL)

{

NODE\* p = l.pDau;

l.pDau = l.pDau->pNext;

delete p;

}

}

void XoaCuoi(LIST &l)

{

NODE\* g;

for (NODE\* p = l.pDau; p != NULL; p = p->pNext)

{

if (p == l.pCuoi)

{

g->pNext = NULL;

l.pCuoi = g;

delete p;

return;

}

g = p;

}

}

int main()

{

LIST l;

NODE\* p;

InPut(l);

OutPut(l);

cout<<"them dau danh sach";

ThemDau(l,p);

cout<<"them cuoi danh sach";

ThemCuoi(l,p);

cout << "Xoa dau danh sach";

XoaDau(l);

cout << "Xoa cuoi danh sach:";

XoaCuoi(l);

system("pause");

}

1. Tuần 3: Cây nhị phân tìm kiếm

- Đọc cây từ file có dạng 1 dòng ghi lần lượt giá trị của các node, cách nhau bằng dấu khoảng trắng (space). Số đầu tiên chính là node gốc. Thầy không yêu cầu vẽ cây.

- Xuất cây theo thứ tự trái (LNR)

- Xuất cây theo thứ tự giữa (NLR)

- Xuất cây theo thứ tự phải (RNL)

- Tìm đường đi tới 1 node

- Xuất độ cao cây

*Bài làm*

* Trình Mỹ Duyên

#include <iostream>

#include <fstream>

#include <array>

#include <vector>

#include <conio.h>

using namespace std;

typedef int TYPEINFO;

struct NODE

{

TYPEINFO data;

NODE\* left;

NODE\* right;

};

typedef NODE\* NODEPTR;

typedef NODEPTR BST;

//Ham khoi tao cay

void Init(BST &r)

{

r = NULL;

}

//Ham kiem tra rong

int Empty(BST r)

{

if (r == NULL)

return 1;

else

return 0;

}

//Ham tao node

NODEPTR CreateNode(TYPEINFO a)

{

NODEPTR p = new NODE;

p->data = a;

p->left = NULL;

p->right = NULL;

return p;

}

//Ham chen them node

//void InsertNode(BST &r, TYPEINFO a)

//{

// if (a < r->data)

// Chen(r->left, a);

// else

// Chen(r->right, a);

//}

//Doc cay tu mot file van ban

void Read( BST &r)

{

FILE \*f;

fopen\_s(&f, "textcay.txt", "r");

if (f == NULL)

{

cout << "\nFile rong!!!";

fclose(f);

}

else

{

fscanf\_s(f, "%n", &r->data);

}

fclose(f);

}

//Ham chen them node

void Chen(BST &r, TYPEINFO a)

{

if (r == NULL)

{

NODEPTR p = CreateNode(a);

r = p;

}

else if (r->data == a)

cout << "Gia tri can chen da co trong cay." << endl;

else

{

if (a < r->data)

Chen(r->left, a);

else

Chen(r->right, a);

}

}

//Doc file, so luong phan tu khong biet truoc

void Load\_1(BST &root)

{

int \*A;

int n = 0;

int number;

FILE \*f = fopen("textcay.txt", "r");

// Dem so luong phan tu co trong file

while (fscanf(f, "%d", &number) > 0)

{

n++;

}

// Cap phat bo nho cho mang A

A = new int[n];

// Quay tro lai file de doc du lieu vao mang A

fseek(f, 0, SEEK\_SET);

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

fscanf(f, "%d", &A[i]);

// Xuat mang da doc

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

Chen(root, A[i]);

}

//Doc file bang Fstream (so luong phan tu bi gioi han)

void Load\_fstream(BST &root)

{

ifstream file;

file.open("textcay.txt", ios::in);

if (file.fail() == true)

{

cout << "\nDoc file bi loi!!!";

}

int arr[100];

int dem = 0, i = 0;

while (!file.eof())

{

file >> arr[i];

i++;

dem++;

}

cout << "\nTong so phan tu: " << dem << endl;

for (int j = 0; j < dem; j++)

{

Chen(root, arr[j]);

}

file.close();

}

//Ham Duyet LNR

void DuyetLNR(BST r)

{

if (r != NULL)

{

DuyetLNR(r->left);

cout << r->data << " ";

DuyetLNR(r->right);

}

}

//Ham Duyet NLR

void DuyetNLR(BST r)

{

if (r != NULL)

{

cout << r->data << " ";

DuyetNLR(r->left);

DuyetNLR(r->right);

}

}

//Ham Duyet LRN

void DuyetLRN(BST r)

{

if (r != NULL)

{

DuyetLRN(r->left);

DuyetLRN(r->right);

cout << r->data << " ";

}

}

//Ham tim kiem

NODEPTR SearchNode(BST r, TYPEINFO a)

{

if (r == NULL)

{

return NULL;

}

NODEPTR p = r;

while (p != NULL)

{

if (p->data < a)

{

p = p->right;

cout << "\t->" << p->data;

}

else if (p->data > a)

{

p = p->left;

cout << "\t->" << p->data;

}

else

return p;

}

return NULL;

}

//Ham tinh chieu cao cay

//int Hight(BST r)

{

if (r == NULL) return 0;

int hl = 1 + ChieuCao(r->left);

int hr = 1 + ChieuCao(r->right);

if (hl >= hr) return hl;

else return hr;

}

void main()

{

BST r;

TYPEINFO a;

Init(r);

Load\_1(r);

//Load\_fstream(r);

cout << "\nDuyet LNR:";

DuyetLNR(r);

cout << "\nDuyet NLR:";

DuyetNLR(r);

cout << "\nDuyet LRN:";

DuyetLRN(r);

cout << "\nNhap node can tim: ";

cin >> a;

//cout << "\nDuong di toi node " << a << " : " << SearchNode(r, a);

//cout << "\nChieu cao cay: " << Hight(r);

cout << "\n";

system("pause");

}

\*

\* \*

* Trương Ngọc Khánh

#include "Header.h"

void TaoNode(FREE& t)

{

t = NULL;

}

void themXvaoCay(FREE& t, int x)

{

if (t == NULL)

{

NODE\* p = new NODE;

p->data = x;

p->pLeft = NULL;

p->pRight = NULL;

t = p;

}

else

{

if (t->data > x)

{

themXvaoCay(t->pLeft, x);

}

else if (t->data < x)

{

themXvaoCay(t->pRight, x);

}

}

}

void DocFile(FREE& t, ifstream& filein)

{

int n;

filein >> n;

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

{

int x;

filein >> x;

themXvaoCay(t, x);

}

}

void NLR(NODE\* t)

{

if (t != NULL)

{

cout << t->data << " ";

NLR(t->pLeft);

NLR(t->pRight);

}

}

void LNR(NODE\* t)

{

if (t != NULL)

{

LNR(t->pLeft);

cout << t->data<<" ";

LNR(t->pRight);

}

}

void RNL(NODE\* t)

{

if (t != NULL)

{

RNL(t->pRight);

cout << t->data << " ";

RNL(t->pLeft);

}

}

int doCaoCay(NODE\* t)

{

int a;

int b;

if (t == NULL)

{

return 0;

}

else

{

a = doCaoCay(t->pLeft);

b = doCaoCay(t->pRight);

if (a > b)

return a + 1;

else

return b + 1;

}

return 1;

}

FREE timDuongDi(NODE\* t, int d)

{

if (t == NULL)

{

return NULL;

}

else

{

if (t->data < d)

{

cout <<"\nNODE tiep theo la:"<< t ->data << endl;

return timDuongDi(t->pRight, d);

}

else if (t->data > d)

{

cout << "\nNODE tiep theo la:" << t->data << endl;

return timDuongDi(t->pLeft, d);

}

else

{

cout << "\nNODE tiep theo la:" << t->data << endl;

return t;

}

}

}

+++++++++++++++++++++++++++++++++++

using namespace std;

struct NODE

{

int data;

struct NODE\* pLeft;

struct NODE\* pRight;

};

typedef NODE\* FREE;

++++++++++++++++++++++++++++++++++++

#include"Header.h"

void main()

{

FREE t;

TaoNode(t);

ifstream filein;

filein.open("BST.txt", ios\_base::in);

DocFile(t, filein);

cout << "\nXuat cay theo dang NLR:" << endl;

NLR(t);

cout << "\nXuat cay theo dang LNR:" << endl;

LNR(t);

cout << "\nXuat cay theo dang RNL:" << endl;

RNL(t);

int p = doCaoCay(t);

cout << "\ndo cao cua cay la:" << p << endl;

int d;

cout << " \nnhap node can tim duong toi:";

cin >> d;

timDuongDi(t, d);

filein.close();

system("pause");

}

\*

\* \*

* Cao Lê Tuấn Vũ

#include "Header.h"

void TaoNode(FREE& t)

{

t = NULL;

}

void themXvaoCay(FREE& t, int x)

{

if (t == NULL)

{

NODE\* p = new NODE;

p->data = x;

p->pLeft = NULL;

p->pRight = NULL;

t = p;

}

else

{

if (t->data > x)

{

themXvaoCay(t->pLeft, x);

}

else if (t->data < x)

{

themXvaoCay(t->pRight, x);

}

}

}

void DocFile(FREE& t, ifstream& filein)

{

int n;

filein >> n;

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

{

int x;

filein >> x;

themXvaoCay(t, x);

}

}

void NLR(NODE\* t)

{

if (t != NULL)

{

cout << t->data << " ";

NLR(t->pLeft);

NLR(t->pRight);

}

}

void LNR(NODE\* t)

{

if (t != NULL)

{

LNR(t->pLeft);

cout << t->data<<" ";

LNR(t->pRight);

}

}

void RNL(NODE\* t)

{

if (t != NULL)

{

RNL(t->pRight);

cout << t->data << " ";

RNL(t->pLeft);

}

}

int doCaoCay(NODE\* t)

{

int a;

int b;

if (t == NULL)

{

return 0;

}

else

{

a = doCaoCay(t->pLeft);

b = doCaoCay(t->pRight);

if (a > b)

return a + 1;

else

return b + 1;

}

return 1;

}

FREE timDuongDi(NODE\* t, int d)

{

if (t == NULL)

{

return NULL;

}

else

{

if (t->data < d)

{

cout <<"\nNODE tiep theo la:"<< t ->data << endl;

return timDuongDi(t->pRight, d);

}

else if (t->data > d)

{

cout << "\nNODE tiep theo la:" << t->data << endl;

return timDuongDi(t->pLeft, d);

}

else

{

cout << "\nNODE tiep theo la:" << t->data << endl;

return t;

}

}

}

+++++++++++++++++++++++++++++++++++

using namespace std;

struct NODE

{

int data;

struct NODE\* pLeft;

struct NODE\* pRight;

};

typedef NODE\* FREE;

++++++++++++++++++++++++++++++++++++

#include"Header.h"

void main()

{

FREE t;

TaoNode(t);

ifstream filein;

filein.open("BST.txt", ios\_base::in);

DocFile(t, filein);

cout << "\nXuat cay theo dang NLR:" << endl;

NLR(t);

cout << "\nXuat cay theo dang LNR:" << endl;

LNR(t);

cout << "\nXuat cay theo dang RNL:" << endl;

RNL(t);

int p = doCaoCay(t);

cout << "\ndo cao cua cay la:" << p << endl;

int d;

cout << " \nnhap node can tim duong toi:";

cin >> d;

timDuongDi(t, d);

filein.close();

system("pause");

}

\*

\* \*

* Lê Văn Huy

\*Header.h

#include<iostream>

using namespace std;

struct NODE

{

int data;

NODE\* pLeft;

NODE\* pRight;

};

NODE\* KhoiTaoCay(int x);

NODE\* TimViTriChen(NODE\* A, int x);

void ChenGiaTri(NODE\*& A, int x);

void TaoCayNhiPhan(NODE\*& A, int a[], int n);

void LNR(NODE\* A);

void NLR(NODE\* A);

void RNL(NODE\* A);

----------------------------------------------

\*Source.cpp

#include"Header.h"

NODE\* KhoiTaoCay(int x)

{

NODE\* p = new NODE();

p->data = x;

p->pLeft = p->pRight = NULL;

return p;

}

NODE\* TimViTriChen(NODE\* A, int x)

{

if (A == NULL)

{

return NULL;

}

NODE\* p = A;

NODE\* s = p;

while (p != NULL)

{

s = p;

if (p->data > x)

{

p = p->pLeft;

}

else

{

p = p->pRight;

}

}

return s;

}

void ChenGiaTri(NODE\*& A, int x)

{

NODE\* n = KhoiTaoCay(x);

if (A == NULL)

{

A = n;

return;

}

else

{

NODE\* s = TimViTriChen(A, x);

if (s != NULL)

{

if (s->data > x)

{

s->pLeft = n;

}

else

{

s->pRight = n;

}

}

}

}

void TaoCayNhiPhan(NODE\*& A, int a[], int n)

{

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

{

ChenGiaTri(A, a[i]);

}

}

void LNR(NODE\* A)

{

if (A != NULL)

{

LNR(A->pLeft);

cout<<"\t"<< A->data;

LNR(A->pRight);

}

}

void NLR(NODE\* A)

{

if (A != NULL)

{

cout<<"\t"<< A->data;

NLR(A->pLeft);

NLR(A->pRight);

}

}

void RNL(NODE\* A)

{

if (A != NULL)

{

RNL(A->pRight);

cout << "\t" << A->data;

RNL(A->pLeft);

}

}

-----------------------------------------------------------

\*Main.cpp

#include"Header.h"

int main()

{

NODE\* A = NULL;

int data[] = {12,54,10,32,17,66,85,24,33,25,6};

int n=11;

TaoCayNhiPhan(A,data,n);

LNR(A);

NLR(A);

RNL(A);

return 0;

system("pause");

}

* Nguyễn Thị Thùy Nhung

#include<iostream>

using namespace std;

struct NODE

{

int data;

NODE\* pLeft;

NODE\* pRight;

};

NODE\* KhoiTaoCay(int x)

{

NODE\* p = new NODE();

p->data = x;

p->pLeft = p->pRight = NULL;

return p;

}

NODE\* TimViTriChen(NODE\* A, int x)

{

if (A == NULL)

{

return NULL;

}

NODE\* p = A;

NODE\* s = p;

while (p != NULL)

{

s = p;

if (p->data > x)

{

p = p->pLeft;

}

else

{

p = p->pRight;

}

}

return s;

}

void ChenGiaTri(NODE\*& A, int x)

{

NODE\* n = KhoiTaoCay(x);

if (A == NULL)

{

A = n;

return;

}

else

{

NODE\* s = TimViTriChen(A, x);

if (s != NULL)

{

if (s->data > x)

{

s->pLeft = n;

}

else

{

s->pRight = n;

}

}

}

}

void TaoCayNhiPhan(NODE\*& A, int a[], int n)

{

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

{

ChenGiaTri(A, a[i]);

}

}

void LNR(NODE\* A)

{

if (A != NULL)

{

LNR(A->pLeft);

cout<<"\t"<< A->data;

LNR(A->pRight);

}

}

void NLR(NODE\* A)

{

if (A != NULL)

{

cout<<"\t"<< A->data;

NLR(A->pLeft);

NLR(A->pRight);

}

}

void RNL(NODE\* A)

{

if (A != NULL)

{

RNL(A->pRight);

cout << "\t" << A->data;

RNL(A->pLeft);

}

}

int main()

{

NODE\* A = NULL;

int data[] = {12,54,10,32,17,66,85,24,33,25,6};

int n=11;

TaoCayNhiPhan(A,data,n);

LNR(A);

NLR(A);

RNL(A);

return 0;

system("pause");

}

1. Tuần 4 – 5: Tìm hiểu kiểu dữ liệu Dictionary trong ngôn ngữ C# và ứng dụng vào việc xây dựng một chương trình quản lý sinh viên (trên console)

**Sinh viên có các thuộc tính:**

- Mã sinh viên - kiểu chuỗi

- Tên sinh viên - kiểu chuỗi

- Ngày sinh - kiểu ngày tháng

- Điểm trung bình - kiểu số thực

**Yêu cầu menu với các chức năng:**

1 - Tự động phát sinh dữ liệu 20 sinh viên

2 - Thêm 1 sinh viên

3 - Xóa 1 sinh viên (sau khi chọn menu này thì chương trình yêu cầu nhập mã sinh viên cần xóa)

4 - Tìm 1 sinh viên dựa theo mã sinh viên (sau khi chọn menu này thì chương trình yêu cầu nhập mã sinh viên cần tìm)

0 - Thoát chương trình

*Bài làm*

* Trình Mỹ Duyên

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace QuanLySinhVien

{

public class SinhVien

{

private string m\_ma;

private string m\_hoten;

private DateTime m\_ngaysinh;

private float m\_diemtb;

SinhVien sv = new SinhVien();

Dictionary<string, SinhVien> dicSV = new Dictionary<string, SinhVien>();

public string ma

{

get { return m\_ma; }

set { m\_ma = value; }

}

public string hoten

{

get { return m\_hoten; }

set { m\_hoten = value; }

}

public DateTime ngaysinh

{

get { return m\_ngaysinh; }

set { m\_ngaysinh = value; }

}

public float diemtb

{

get { return m\_diemtb; }

set { m\_diemtb = value; }

}

public SinhVien(string ma, string hoten, DateTime ngaysinh, float diemtb)

{

m\_ma = ma;

m\_hoten = hoten;

m\_ngaysinh = ngaysinh;

m\_diemtb = diemtb;

}

public SinhVien()

{

m\_ma = "";

m\_hoten = "";

m\_ngaysinh = DateTime.Now;

m\_diemtb = 0;

}

public void Xuat()

{

Console.WriteLine("MA: " + m\_ma);

Console.WriteLine("Ho ten: " + m\_hoten);

Console.WriteLine("Ngay sinh: " + m\_ngaysinh);

Console.WriteLine("Diem tb: " + m\_diemtb);

}

public void Nhap()

{

Console.WriteLine("Nhap ma sv: ");

sv.ma=Console.ReadLine();

Console.WriteLine("Nhap ho ten: ");

sv.hoten = Console.ReadLine();

Console.WriteLine("Nhap ngay sinh: ");

sv.ngaysinh = DateTime.Parse(Console.ReadLine());

Console.WriteLine("Nhap diem trung binh: ");

sv.diemtb = float.Parse(Console.ReadLine());

}

public void Them(SinhVien sv)

{

dicSV.Add(sv.ma, sv);

}

public bool Xoa()

{

Console.WriteLine("Nhap MaSV muon xoa: ");

string maxoa = Console.ReadLine();

if (dicSV.Remove(maxoa))

return true;

return false;

}

public bool TimTheoMa(SinhVien sv)

{

Console.WriteLine("Nhap MaSV can tim: ");

string ma = Console.ReadLine();

if (dicSV.ContainsKey(sv.ma))

{

return true;

sv.Xuat();

}

return false;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace QuanLySinhVien

{

public class Program: SinhVien

{

static void Main(string[] args)

{

SinhVien sv = new SinhVien();

Dictionary<string, SinhVien> dicSV = new Dictionary<string, SinhVien>();

menu:

Console.WriteLine("========================== MENU ============================");

Console.WriteLine("0 - Thoat chuong trinh");

Console.WriteLine("1 - Tu phat sinh phat sinh du lieu");

Console.WriteLine("2 - Them 1 sinh vien");

Console.WriteLine("3 - Xoa 1 sinh vien");

Console.WriteLine("4 - Tim 1 sinh vien");

Console.WriteLine("============================================================");

Console.WriteLine("Nhap lua chon: ");

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

switch(chon)

{

case 1:

{

dicSV = new Dictionary<string, SinhVien>(20);

goto menu;

}

case 2:

{

sv.Nhap();

sv.Them(sv);

goto menu;

}

case 3:

{

if (sv.Xoa() == true)

Console.WriteLine("Xoa thanh cong!");

else

Console.WriteLine("Khong tim thay SV nay!");

goto menu;

}

case 4:

{

if (sv.TimTheoMa(sv) == false)

Console.WriteLine("Khong tim thay SV nay!");

else

Console.WriteLine("Thong tin SV: ");

goto menu;

}

case 0:break;

}

Console.ReadLine();

}

}

}

\*

\* \*

* Trương Ngọc Khánh

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class HocVien

{

static void Main(string[] args)

{

HocVien hv = new HocVien();

Dictionary<string, HocVien> MyDic = new Dictionary<string, HocVien>();

MyDic.Add(hv.masv, hv);

string c;

string key = "", value = "";

Console.WriteLine("Menu");

Console.WriteLine("0 - Thoat chuong trinh");

Console.WriteLine("1 - tu phat sinh phat xin du lieu");

Console.WriteLine("2 - thêm 1 sinh viên");

Console.WriteLine("3 - Xóa 1 sinh viên");

Console.WriteLine("4 - Tìm 1 sinh viên");

do

{

Console.Write("Chon thao tac:");

c = Console.ReadLine();

switch (c)

{

case "0":

return;

case "1":

return;

case "2":

Console.Write("Nhap key:");

key = Console.ReadLine();

Console.Write("Nhap value:");

value = Console.ReadLine();

MyDic.Add(key, hv);

break;

case "3":

Console.Write("Nhap key:");

key = Console.ReadLine();

bool re =MyDic.Remove(key);

if (re == true)

Console.WriteLine("Xoa thanh cong");

else

Console.WriteLine("Khong tim thay SV nay");

break;

case "4":

Console.Write("Nhap MaSv can tim:");

string a;

if (MyDic.ContainsKey(hv.masv))

{

hv.Xuat();

}

if (value == null)

Console.WriteLine("Khong tim thay SV nay");

else

Console.WriteLine("{0} :" + " {1}", key, value);

break;

}

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

} while (c != "0");

}

private string m\_masv;

private string m\_hoten;

private DateTime m\_ngaysinh;

private float m\_diemtb;

public string masv

{

get { return m\_masv; }

set { m\_masv = value; }

}

public string hoten

{

get { return m\_hoten; }

set { m\_hoten = value; }

}

public DateTime ngaysinh

{

get { return m\_ngaysinh; }

set { m\_ngaysinh = value; }

}

public float diemtb

{

get { return m\_diemtb; }

set { m\_diemtb = value; }

}

HocVien(string masv, string hoten, DateTime ngaysinh, float diemtb)

{

m\_masv = masv;

m\_hoten = hoten;

m\_ngaysinh = ngaysinh;

m\_diemtb = diemtb;

}

HocVien()

{

m\_masv = "";

m\_hoten = "";

m\_ngaysinh = DateTime.Now;

m\_diemtb = 0;

}

public void Nhap ()

{

Console.WriteLine("ten: ");

hoten = Console.ReadLine();

Console.WriteLine("ma: ");

masv = Console.ReadLine();

Console.WriteLine("ngay sinh:");

ngaysinh = DateTime.Parse(Console.ReadLine());

Console.WriteLine("diem: ");

diemtb = float.Parse(Console.ReadLine());

}

public void Xuat()

{

Console.WriteLine("ten la: " + hoten);

Console.WriteLine("ma la" + masv);

Console.WriteLine("ngay sinh " + ngaysinh);

Console.WriteLine("diem la" + diemtb);

}

}

}

\*

\* \*

* Cao Lê Tuấn Vũ

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace bailam2

{

class Program

{

static void Main(string[] args)

{

string MSV;

Console.Write("Nhập vào MSV của bạn:");

MSV = Console.ReadLine();

string tensv;

Console.Write("Nhập vào MSV của bạn:");

tensv = Console.ReadLine();

DateTime ns;

Console.Write("Nhập vào MSV của bạn:");

ns = Console.

float dtb;

Console.Write("Nhập vào MSV của bạn:");

dtb = Console.ReadLine();

Console.ReadLine();

Dictionary<string, string> SV = new Dictionary<string, string>();

string c;

string key = "", value = "";

Console.WriteLine("Menu");

Console.WriteLine("0 - Thoat chuong trinh");

Console.WriteLine("1 - tự động phát sinh dữ liệu");

Console.WriteLine("2 - thêm 1 sinh viên");

Console.WriteLine("3 - Xóa 1 sinh viên");

Console.WriteLine("4 - Tìm 1 sinh viên");

do

{

Console.Write("Chon thao tac:");

c = Console.ReadLine();

switch (c)

{

case "0":

return;

case "1":

return;

case "2":

Console.Write("Nhap key:");

key = Console.ReadLine();

Console.Write("Nhap value:");

value = Console.ReadLine();

SV.Add(key, value);

break;

case "3":

Console.Write("Nhap key:");

key = Console.ReadLine();

bool re = SV.Remove(key);

if (re == true)

Console.WriteLine("Xoa thanh cong");

else

Console.WriteLine("Khong tim thay SV nay");

break;

case "4":

Console.Write("Nhap MaSv can tim:");

key = Console.ReadLine();

value = SV.Search(key);

if (value == null)

Console.WriteLine("Khong tim thay SV nay");

else

Console.WriteLine("{0} :" + " {1}", key, value);

break;

}

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

} while (c != "0");

}

}

}

* Lê Văn Huy

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace bailam2

{

class Program

{

static void Main(string[] args)

{

string MSV;

Console.Write("Nh?p vào MSV c?a b?n:");

MSV = Console.ReadLine();

string tensv;

Console.Write("Nh?p vào MSV c?a b?n:");

tensv = Console.ReadLine();

DateTime ns;

Console.Write("Nh?p vào MSV c?a b?n:");

ns = Console.

float dtb;

Console.Write("Nh?p vào MSV c?a b?n:");

dtb = Console.ReadLine();

Console.ReadLine();

Dictionary<string, string> SV = new Dictionary<string, string>();

string c;

string key = "", value = "";

Console.WriteLine("Menu");

Console.WriteLine("0 - Thoat chuong trinh");

Console.WriteLine("1 – Tự động phát sinh dữ liệu");

Console.WriteLine("2 - Thêm 1 sinh viên");

Console.WriteLine("3 - Xóa 1 sinh viên");

Console.WriteLine("4 - Tìm 1 sinh viên");

do

{

Console.Write("Chon thao tac:");

c = Console.ReadLine();

switch (c)

{

case "0":

return;

case "1":

return;

case "2":

Console.Write("Nhap key:");

key = Console.ReadLine();

Console.Write("Nhap value:");

value = Console.ReadLine();

SV.Add(key, value);

break;

case "3":

Console.Write("Nhap key:");

key = Console.ReadLine();

bool re = SV.Remove(key);

if (re == true)

Console.WriteLine("Xoa thanh cong");

else

Console.WriteLine("Khong tim thay SV nay");

break;

case "4":

Console.Write("Nhap MaSv can tim:");

key = Console.ReadLine();

value = SV.Search(key);

if (value == null)

Console.WriteLine("Khong tim thay SV nay");

else

Console.WriteLine("{0} :" + " {1}", key, value);

break;

}

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

} while (c != "0");

}

}

}

* Nguyễn Thị Thùy Nhung

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace bailam2

{

class Program

{

static void Main(string[] args)

{

string MSV;

Console.Write("Nhap vào MssV cua ban:");

MSV = Console.ReadLine();

string tensv;

Console.Write("Nhap vào MSSV cua ban:");

tensv = Console.ReadLine();

DateTime ns;

Console.Write("Nhap vào MSSV :");

ns = Console.

float dtb;

Console.Write("Nhap vào MSSV :");

dtb = Console.ReadLine();

Console.ReadLine();

Dictionary<string, string> SV = new Dictionary<string, string>();

string c;

string key = "", value = "";

Console.WriteLine("Menu");

Console.WriteLine("0 - Thoat chuong trinh");

Console.WriteLine("1 - tu dong phat sinh du lieu");

Console.WriteLine("2 - thêm 1 sinh viên");

Console.WriteLine("3 - Xóa 1 sinh viên");

Console.WriteLine("4 - Tìm 1 sinh viên");

do

{

Console.Write("Chon thao tac:");

c = Console.ReadLine();

switch (c)

{

case "0":

return;

case "1":

return;

case "2":

Console.Write("Nhap key:");

key = Console.ReadLine();

Console.Write("Nhap value:");

value = Console.ReadLine();

SV.Add(key, value);

break;

case "3":

Console.Write("Nhap key:");

key = Console.ReadLine();

bool re = SV.Remove(key);

if (re == true)

Console.WriteLine("Xoa thanh cong");

else

Console.WriteLine("Khong tim thay SV nay");

break;

case "4":

Console.Write("Nhap MaSv can tim:");

key = Console.ReadLine();

value = SV.Search(key);

if (value == null)

Console.WriteLine("Khong tim thay SV nay");

else

Console.WriteLine("{0} :" + " {1}", key, value);

break;

}

Console.WriteLine("\n");

} while (c != "0");

}

}

}

1. Tuần 6: Sự khác biệt giữa Hashtable, Dictionary và List, khi nào thì dùng loại dữ liệu nào

*Bài làm*

* Trình Mỹ Duyên

|  |  |  |  |
| --- | --- | --- | --- |
|  | Hashtable | Dictionary | List |
| Khái niệm | Là một Collections lưu trữ dữ liệu dưới dạng các cặp Key – value | Như Hashtable | Là một Generic Collections giúp lưu trữ và quản lý 1 danh sách các đối tượng theo kiểu mảng |
| Sử dụng | Cần phải khai báo thư viện “Collection” | Như Hashtable | Thư viện: System.Collection.Generic |
| Kiểu của Key - Value | Object | Phải xác định kiểu dữ liệu cụ thể | Không. Chỉ số Index |
| Truy xuất dữ liệu | Chỉ cần kiếm được địa chỉ khi xác định được Key hoặc Value | Như Hashtable | Biết được địa chỉ thì truy xuất được ngay vị trí trước hoặc sau địa chỉ đó |
| Sắp xếp phần tử trong | Không được | Không được | Được |
|  | Người ta thường dùng Hash hơn Dic vì nó hiệu quả cho dữ liệu lớn, lợi ích cao. Khi thêm hoặc xóa Hash sẽ tự sắp xếp lại, còn Dic sẽ theo thứ tự thêm vào | | Vì Hash và Dic tương đối giống nhau, là 1 kiểu danh sách theo từ khóa còn List thì trái ngược: nó theo kiểu mảng |

\*

\* \*

* Trương Ngọc Khánh

Hashtable | Dictionary

------------------------------------------ | -------------------------------------------

Threadsafe-Hỗ trọ multi threadiing |

Không đụng độ tài nguyên | Không hỗ trợ

------------------------------------------ | -------------------------------------------

cặp Key-value lưu kểu object | Phải xác định cụ thể kểu dữ liệu cho cặp Key-Value

------------------------------------------ | -------------------------------------------

Truy xuất phần tử không tồn tại trong | Truy xuất phần tử trong Dictinary sẽ báo lỗi

Hashtable sẽ không báo lỗi=>return |

NULL |

------------------------------------------ | -------------------------------------------

HIệu quả cho dữ liệu lớn | Không hiệu quả cho dữ liệu lớn

------------------------------------------ | -------------------------------------------

hoặc xóa phần tử trong HashTable | Các phần tử nằm theo thứ tự được thêm vào

------------------------------------------ | -------------------------------------------

tìm kiếm nhanh hơn | Tìm kiếm chậm hơn

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

List là gì?

List là 1 Generic Collections đưa ra như một sự thay thế ArrayList vì thế về khái niệm cũng như sử dụng nó hoàn toàn giống với ArrayList. (bạn có

thể tham khảo chi tiết trong bài ARRAYLIST TRONG C#)

Ở đây mình chỉ trình bày lại một số ý để những bạn nào không theo dõi những bài trước vẫn có thể hiểu được.

List trong C# là một Generic Collections giúp lưu trữ và quản lý một danh sách các đối tượng theo kiểu mảng (truy cập các phần tử bên trong thông

qua chỉ số index).

Để sử dụng các Collections trong .NET ta cần thêm thư viện System.Collections.Generic bằng câu lệnh:

using System.Collections.Generic;

\*

\* \*

* Cao Lê Tuấn Vũ

Là một Collections lưu trữ dữ liệu dưới dạng cặp Key - Value. Key đại diện cho 1 khoá giống

như chỉ số phần tử của mảng và Value chính là giá trị tương ứng của khoá đó. Ta sẽ dử dụng Key để truy cập đến Value tương ứng.

Vì Key và Value đều là kiểu object nên ta có thể lưu trữ được mọi kiểu dữ liệu từ những kiểu cơ sở đến kiểu phức tạp (class).

Nếu các Key của Hashtable là các số nguyên tăng dần từ 0 thì Hashtable

nhìn trông giống ArrayList (chỉ là giống về bề ngoài thôi chứ bên trong rất khác nhau).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Tương tự như List, Dictionary chính là sự thay thế cho Collections Hashtable đã được học.

Cho nên về khái niệm hay sử dụng thì Dictionary đều sẽ giống Hashtable.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Kết Luận(Phần ghi cho thầy)

-Dictionary chính là sự thay thế cho Collections Hashtable nên về khái niệm hay sử dụng Dictionary đều sẽ giống Hashtable.

-Chúng chỉ khác nhau ở chỗ là Hash cặp Key - value lưu kểu object, còn Dec thì phải xác định kểu giữu liệu cụ thể

-Người ta thường dùng hashh hơn là dic vì chúng hiệu quả cho dữ liệu lớn , lợi ích cao, khi thêm hoặc xóa hash sẽ tự sắp xếp lại

con Dic sẽ theo thứ tự thêm vào

-Do Dic và hash tương đối giống nhau nên Lish khác cả 2 thằng

-List có dạng như lưu kểu liên kết, nhiều biến khác nhau và biết được biến kế tiếp là gì, còn has và dic thì lại chỉ lưu được 1 cặp

biến cựu thể theo kểu Key-Value

-List có thẻ sắp xếp các phần tử trong nhưng hash và dic thì lại ko

-List biết được thằng nằm trước thì có thể kiếm được thằng nằm tiếp theo hoặc trước nó nữa nhưng Dic và hash thì chỉ tìm kiếm được

khi xác định được Key hoặc value

\*

\* \*

* Lê Văn Huy

\*Dictionary

Dictionary<string, Customer> customers = new Dictionary<string, Customer>();//Khởi tạo Dictionary

Customer customer = customers["Ali G"];//Truy xu?t Dictionary

-phải xác định cụ thể kiểu dữ liệu củaa key-value

-Truy xuất phần tử không tồn tại trong Dictionary sẽ bao lỗi

-không hiệu qủa cho dữ liệu lớn

-Các phần tử nằm theo thứ tự được thêm vào

-Tìm kiếm chậm

=======================================================================

\*Hashtable

Hashtable customers = new Hashtable();//Khởi tạo Hashtable

Customer customer = (Customer)customers["Ali G"];//Truy xuất Hashtable

-Threadsafe -Hổ trợ multi theading không đụng độ tài nguyên

-Cặp key-value luu kiểu object

-Truy xuất phần tử không tôn tại trong Hashtable sẽ return Null

-Hiệu quả cho dữ liệu

-Các phần tử được sắp xếp lại mỗi khi thêm hoặc xóa phần tử trong Hashtable

-Tìm kiếm nhanh

\*

\* \*

* Nguyễn Thị Thùy Nhung

\*Dictionary

Dictionary<string, Customer> customers = new Dictionary<string, Customer>();//Khởi tạo Dictionary

Customer customer = customers["Ali G"];//Truy xu?t Dictionary

-phải xác định cụ thể kiểu dữ liệu củaa key-value

-Truy xuất phần tử không tồn tại trong Dictionary sẽ bao lỗi

-không hiệu qủa cho dữ liệu lớn

-Các phần tử nằm theo thứ tự được thêm vào

-Tìm kiếm chậm

=======================================================================

\*Hashtable

Hashtable customers = new Hashtable();//Khởi tạo Hashtable

Customer customer = (Customer)customers["Ali G"];//Truy xuất Hashtable

-Threadsafe -Hổ trợ multi theading không đụng độ tài nguyên

-Cặp key-value luu kiểu object

-Truy xuất phần tử không tôn tại trong Hashtable sẽ return Null

-Hiệu quả cho dữ liệu

-Các phần tử được sắp xếp lại mỗi khi thêm hoặc xóa phần tử trong Hashtable

-Tìm kiếm nhanh

1. **Làm bài nhóm –** Đề tài: Quản lý cửa hàng bán sách (chỉ bán sách)
2. Tuần 7: Liệt kê các loại đối tượng cần quản lý của đề tài và các thuộc tính của loại đối tượng

*Bài làm*

- Sách

+ Mã sách

+ Tên sách

+ Tác giả

+ Ngày xuất bản

+ Giá

+ Thể loại

- Khách hàng

+ Mã KH

+ Tên KH

+ Thông tin: địa chỉ, sđt, giới tính

- Nhân viên

+ Mã NV

+ Tên NV

+ Chức vụ

+ Ngày vào làm

+ Thông tin: địa chỉ, sđt, giới tính

- Hóa đơn

+ Mã HĐ

+ Mã NV

+ Mã KH

+ Ngày xuất HĐ

+ Thành tiền

+ Hình thức thanh toán

- Nhà xuất bản

+ Mã NXB

+ Tên NXB

+ Thông tin liên hệ

- Kho

+ Mã kho

+ Tên kho

+ Số lượng nhập và tồn

1. Tuần 8: Hoàn thiện các đối tượng cần quản lý và tiến hành code chương trình. Gửi mail phân công công việc cho Thầy

*Bài làm*

- Sách (Huy)

+ Mã sách

+ Tên sách

+ DS mã TG

+ Mã NXB

+ Giá

+ DS mã TL

- Khách hàng (Khánh)

+ Mã KH

+ Tên KH

+ Thông tin: địa chỉ, sđt, giới tính

- Nhân viên (Vũ)

+ Mã NV

+ Tên NV

+ Chức vụ

+ Ngày vào làm

+ Thông tin: địa chỉ, sđt, giới tính

- Hóa đơn (Duyên)

+ Mã HĐ

+ Mã NV

+ Mã KH

+ Ngày xuất HĐ

+ Thành tiền

+ Hình thức thanh toán

- Chi tiết hóa đơn (Duyên + Vũ)

+ Mã HĐ

+ Mã sách

+ Tên Sách

+ Số lượng

+ Giá

- Nhà xuất bản (Nhung)

+ Mã NXB

+ Tên NXB

+ Thông tin liên hệ

- Kho (Khánh)

+ Mã kho

+ Tên kho

+ Số lượng nhập và tồn

- Tác giả (Nhung)

+ Mã TG

+ Tên TG

- Thể loại (Huy)

+ Mã TL

+ Tên TL

1. Tuần 9 – 10: Hoàn thiện code và viết báo cáo về Đồ án

*Bài làm*

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

using System.Runtime.Serialization.Formatters.Binary;

namespace QuanLySach

{

public class Program

{

static void Main(string[] args)

{

menu:

int chon, tt;

Console.WriteLine(" ========================== MENU ============================ ");

Console.WriteLine("0 - Thoat chuong trinh");

Console.WriteLine("1 - Sach");

Console.WriteLine("2 - Tac Gia");

Console.WriteLine("3 - The Loai");

Console.WriteLine("4 - Nha Xuat Ban");

Console.WriteLine("5 - Hoa Don");

Console.WriteLine("6 - Chi Tiet Hoa Don");

Console.WriteLine("7 - Khach Hang");

Console.WriteLine("8 - Kho");

Console.WriteLine("9 - Nhan Vien");

Console.WriteLine(" ============================================================ ");

do

{

Console.Write("Chon Menu: ");

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

switch (chon)

{

case 0:

break;

case 1:

{

Sach a = new Sach();

DanhsachSach danhsach = new DanhsachSach();

DanhsachSach n = new DanhsachSach();

List<Sach> saches = new List<Sach>();

menu1:

int sach;

Console.WriteLine(" ========================== MENU SACH ============================ ");

Console.WriteLine("0 - Thoat chuong trinh");

Console.WriteLine("1 - Nhap Sach");

Console.WriteLine("2 - Them Sach");

Console.WriteLine("3 - Ghi Sach");

Console.WriteLine("4 - Xoa Sach");

Console.WriteLine("5 - Doc Sach");

Console.WriteLine("6 - Tim Sach");

Console.WriteLine("7 - Cap Nhat Lai Sach");

Console.WriteLine(" ================================================================= ");

do

{

Console.Write("Chon thao tac: ");

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

switch (sach)

{

case 0:

break;

case 1:

{

danhsach.Them(a);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu1;

break;

}

case 2:

{

danhsach.Them(a);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu1;

break;

}

case 3:

{

danhsach.ghiflie();

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu1;

break;

}

case 4:

{

danhsach.Xoa(a);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu1;

break;

}

case 5:

{

danhsach.docfile();

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu1;

break;

}

case 6:

{

Console.WriteLine("Nhap ma sach muon tim: ");

string masach = Console.ReadLine();

danhsach.Tim(masach);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu1;

break;

}

case 7:

{

danhsach.CapNhat(a);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu1;

break;

}

}

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

} while (sach != 0);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu;

break;

}

case 2:

{

TacGia b = new TacGia();

TacGia b1 = new TacGia();

DanhSachTacGia danhsachtg = new DanhSachTacGia();

DanhSachTacGia b2 = new DanhSachTacGia();

List<TacGia> tacgias = new List<TacGia>();

menu2:

int tacgia;

Console.WriteLine(" ========================== MENU TAC GIA ============================ ");

Console.WriteLine("0 - Thoat chuong trinh");

Console.WriteLine("1 - Nhap tac gia");

Console.WriteLine("2 - Them tac gia");

Console.WriteLine("3 - Ghi tac gia");

Console.WriteLine("4 - Xoa tac gia");

Console.WriteLine("5 - Doc tac gia");

Console.WriteLine("6 - Tim tac gia");

Console.WriteLine("7 - Cap Nhat Lai tacgia");

Console.WriteLine(" ================================================================= ");

do

{

Console.Write("Chon thao tac: ");

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

switch (tacgia)

{

case 0:

break;

case 1:

{

danhsachtg.Them(b);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu2;

break;

}

case 2:

{

danhsachtg.Them(b);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu2;

break;

}

case 3:

{

danhsachtg.ghiflie();

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu2;

break;

}

case 4:

{

danhsachtg.Xoa(b1);

goto menu2;

}

case 5:

{

danhsachtg.docfile();

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu2;

break;

}

case 6:

{

Console.WriteLine("Nhap ma tac gia muon tim: ");

string matacgia = Console.ReadLine();

danhsachtg.Tim(matacgia);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu2;

break;

}

case 7:

{

danhsachtg.CapNhat(b);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu2;

break;

}

}

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

} while (tacgia != 0);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu;

break;

}

case 3:

{

TheLoai c = new TheLoai();

TheLoai c1 = new TheLoai();

DanhSachTheLoai danhsach = new DanhSachTheLoai();

DanhSachTheLoai c2 = new DanhSachTheLoai();

List<TheLoai> theloais = new List<TheLoai>();

menu3:

int theloai;

Console.WriteLine(" ========================== MENU THE LOAI============================ ");

Console.WriteLine("0 - Thoat chuong trinh");

Console.WriteLine("1 - Nhap the loai");

Console.WriteLine("2 - Them the loai");

Console.WriteLine("3 - Ghi the loai");

Console.WriteLine("4 - Xoa the loai");

Console.WriteLine("5 - Doc the loai");

Console.WriteLine("6 - Tim the loai");

Console.WriteLine("7 - Cap Nhat Lai the loai");

Console.WriteLine(" ================================================================= ");

do

{

Console.Write("Chon thao tac: ");

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

switch (theloai)

{

case 0:

break;

case 1:

{

danhsach.Them(c);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu3;

break;

}

case 2:

{

danhsach.Them(c);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu3;

break;

}

case 3:

{

danhsach.ghiflie();

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu3;

break;

}

case 4:

{

danhsach.Xoa(c1);

goto menu3;

}

case 5:

{

danhsach.docfile();

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu3;

break;

}

case 6:

{

Console.WriteLine("Nhap ma the loai muon tim: ");

string matl = Console.ReadLine();

danhsach.Tim(matl);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu3;

break;

}

case 7:

{

danhsach.CapNhat(c);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu3;

break;

}

}

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

} while (theloai != 0);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu;

break;

}

case 4:

{

NhaXuatBan d = new NhaXuatBan();

NhaXuatBan d1 = new NhaXuatBan();

DanhsachNXB danhsachnxb = new DanhsachNXB();

DanhsachNXB d2 = new DanhsachNXB();

List<NhaXuatBan> nxbs = new List<NhaXuatBan>();

menu4:

int NXB;

Console.WriteLine(" ===================== MENU NHA XUAT BAN ========================== ");

Console.WriteLine("0 - Thoat chuong trinh");

Console.WriteLine("1 - Nhap NhaXuatBan");

Console.WriteLine("2 - Them NhaXuatBan");

Console.WriteLine("3 - Ghi NhaXuatBan");

Console.WriteLine("4 - Xoa NhaXuatBan");

Console.WriteLine("5 - Doc NhaXuatBan");

Console.WriteLine("6 - Tim NhaXuatBan");

Console.WriteLine("7 - Cap Nhat Lai NhaXuatBan");

Console.WriteLine(" ================================================================= ");

do

{

Console.Write("Chon thao tac: ");

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

switch (NXB)

{

case 0:

break;

case 1:

{

danhsachnxb.Them(d);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu4;

break;

}

case 2:

{

danhsachnxb.Them(d);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu4;

break;

}

case 3:

{

danhsachnxb.ghiflie();

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu4;

break;

}

case 4:

{

danhsachnxb.Xoa(d);

goto menu4;

}

case 5:

{

danhsachnxb.docfile();

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu4;

break;

}

case 6:

{

Console.WriteLine("Nhap ma NhaXuatBan muon tim: ");

string manxb = Console.ReadLine();

danhsachnxb.Tim(manxb);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu4;

break;

}

case 7:

{

danhsachnxb.CapNhat(d);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu4;

break;

}

}

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

} while (NXB != 0);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu;

break;

}

case 5:

HoaDon e = new HoaDon();

DanhsachHoaDon e1 = new DanhsachHoaDon();

DanhsachHoaDon danhsachhd = new DanhsachHoaDon();

DanhsachHoaDon e2 = new DanhsachHoaDon();

List<HoaDon> hoadon = new List<HoaDon>();

menu5:

int hd;

Console.WriteLine(" ===================== MENU HOA DON========================== ");

Console.WriteLine("0 - Thoat chuong trinh");

Console.WriteLine("1 - Nhap HoaDon");

Console.WriteLine("2 - Them HoaDon");

Console.WriteLine("3 - Ghi HoaDon");

Console.WriteLine("4 - Xoa HoaDon");

Console.WriteLine("5 - Doc HoaDon");

Console.WriteLine("6 - Tim HoaDon");

Console.WriteLine("7 - Cap Nhat Lai HoaDon");

Console.WriteLine(" ================================================================= ");

do

{

Console.Write("Chon thao tac: ");

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

switch (hd)

{

case 0:

break;

case 1:

{

danhsachhd.Them(e);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu5;

break;

}

case 2:

{

danhsachhd.Them(e);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu5;

break;

}

case 3:

{

danhsachhd.ghiflie();

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu5;

break;

}

case 4:

{

danhsachhd.Xoa(e);

goto menu5;

}

case 5:

{

danhsachhd.docfile();

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu5;

break;

}

case 6:

{

Console.WriteLine("Nhap ma NhaXuatBan muon tim: ");

string mahd = Console.ReadLine();

danhsachhd.Tim(mahd);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu5;

break;

}

case 7:

{

danhsachhd.CapNhat(e);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu5;

break;

}

}

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

} while (hd != 0);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu;

break;

case 6:

ChiTietHoaDon f = new ChiTietHoaDon();

ChiTietHoaDon f1 = new ChiTietHoaDon();

DSChiTietHoaDon danhsachcthd = new DSChiTietHoaDon();

DSChiTietHoaDon f2 = new DSChiTietHoaDon();

List<ChiTietHoaDon> cthds = new List<ChiTietHoaDon>();

menu6:

int cthd;

Console.WriteLine(" ===================== MENU CHI TIET HOA DON ========================== ");

Console.WriteLine("0 - Thoat chuong trinh");

Console.WriteLine("1 - Nhap ChiTietHoaDon");

Console.WriteLine("2 - Them ChiTietHoaDon");

Console.WriteLine("3 - Ghi ChiTietHoaDon");

Console.WriteLine("4 - Xoa ChiTietHoaDon");

Console.WriteLine("5 - Doc ChiTietHoaDon");

Console.WriteLine("6 - Tim ChiTietHoaDon");

Console.WriteLine("7 - Cap Nhat Lai ChiTietHoaDon");

Console.WriteLine(" ================================================================= ");

do

{

Console.Write("Chon thao tac: ");

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

switch (cthd)

{

case 0:

break;

case 1:

{

danhsachcthd.Them(f);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu6;

break;

}

case 2:

{

danhsachcthd.Them(f);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu6;

break;

}

case 3:

{

danhsachcthd.ghiflie();

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu6;

break;

}

case 4:

{

danhsachcthd.Xoa(f);

goto menu6;

}

case 5:

{

danhsachcthd.docfile();

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu6;

break;

}

case 6:

{

Console.WriteLine("Nhap ma NhaXuatBan muon tim: ");

string macthd = Console.ReadLine();

danhsachcthd.Tim(macthd);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu6;

break;

}

case 7:

{

danhsachcthd.CapNhat(f);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu6;

break;

}

}

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

} while (cthd != 0);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu;

break;

case 7:

KhachHang g = new KhachHang();

KhachHang g1 = new KhachHang();

DanhSachKhachHang danhsachKH = new DanhSachKhachHang();

DanhSachKhachHang g2 = new DanhSachKhachHang();

List<KhachHang> cKHS= new List<KhachHang>();

menu7:

int ctkh;

Console.WriteLine(" ===================== MENU KHACH HANG ========================== ");

Console.WriteLine("0 - Thoat chuong trinh");

Console.WriteLine("1 - Nhap KhachHang");

Console.WriteLine("2 - Them KhachHang");

Console.WriteLine("3 - Ghi KhachHang");

Console.WriteLine("4 - Xoa KhachHang");

Console.WriteLine("5 - Doc KhachHang");

Console.WriteLine("6 - Tim KhachHang");

Console.WriteLine("7 - Cap Nhat Lai KhachHang");

Console.WriteLine(" ================================================================= ");

do

{

Console.Write("Chon thao tac: ");

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

switch (cthd)

{

case 0:

break;

case 1:

{

danhsachKH.Them(g);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu7;

break;

}

case 2:

{

danhsachKH.Them(g);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu7;

break;

}

case 3:

{

danhsachKH.ghiflie();

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu7;

break;

}

case 4:

{

danhsachKH.Xoa(g);

goto menu7;

}

case 5:

{

danhsachKH.docfile();

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu7;

break;

}

case 6:

{

Console.WriteLine("Nhap ma NhaXuatBan muon tim: ");

string makh = Console.ReadLine();

danhsachKH.Tim(makh);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu7;

break;

}

case 7:

{

danhsachKH.CapNhat(g);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu7;

break;

}

}

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

} while (cthd != 0);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu;

break;

case 8:

Kho h = new Kho();

Kho h1 = new Kho();

DanhsachKho danhsachKHo = new DanhsachKho();

DanhsachKho h2 = new DanhsachKho();

List<Kho> kho = new List<Kho>();

menu8:

int Kho;

Console.WriteLine(" ===================== MENU KHO ========================== ");

Console.WriteLine("0 - Thoat chuong trinh");

Console.WriteLine("1 - Nhap Kho");

Console.WriteLine("2 - Them Kho");

Console.WriteLine("3 - Ghi Kho");

Console.WriteLine("4 - Xoa Kho");

Console.WriteLine("5 - Doc Kho");

Console.WriteLine("6 - Tim Kho");

Console.WriteLine("7 - Cap Nhat Lai Kho");

Console.WriteLine(" ================================================================= ");

do

{

Console.Write("Chon thao tac: ");

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

switch (Kho)

{

case 0:

break;

case 1:

{

danhsachKHo.Them(h);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu8;

break;

}

case 2:

{

danhsachKHo.Them(h);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu8;

break;

}

case 3:

{

danhsachKHo.ghiflie();

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu8;

break;

}

case 4:

{

danhsachKHo.Xoa(h);

goto menu8;

}

case 5:

{

danhsachKHo.docfile();

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu8;

break;

}

case 6:

{

Console.WriteLine("Nhap ma Kho muon tim: ");

string makho = Console.ReadLine();

danhsachKHo.Tim(makho);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu8;

break;

}

case 7:

{

danhsachKHo.CapNhat(h);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu8;

break;

}

}

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

} while (Kho != 0);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu;

break;

case 9:

NhanVien k = new NhanVien();

NhanVien k1 = new NhanVien();

DanhSachNhanVien danhsachnv = new DanhSachNhanVien();

DanhSachNhanVien k2 = new DanhSachNhanVien();

List<NhanVien> NV = new List<NhanVien>();

menu9:

int nv;

Console.WriteLine(" ===================== MENU NHAN VIEN ========================== ");

Console.WriteLine("0 - Thoat chuong trinh");

Console.WriteLine("1 - Nhap NhanVien");

Console.WriteLine("2 - Them NhanVien");

Console.WriteLine("3 - Ghi NhanVien");

Console.WriteLine("4 - Xoa NhanVien");

Console.WriteLine("5 - Doc NhanVien");

Console.WriteLine("6 - Tim NhanVien");

Console.WriteLine("7 - Cap Nhat Lai NhanVien");

Console.WriteLine(" ================================================================= ");

do

{

Console.Write("Chon thao tac: ");

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

switch (nv)

{

case 0:

break;

case 1:

{

danhsachnv.Them(k);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu9;

break;

}

case 2:

{

danhsachnv.Them(k);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu9;

break;

}

case 3:

{

danhsachnv.ghiflie();

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu9;

break;

}

case 4:

{

danhsachnv.Xoa(k);

goto menu9;

}

case 5:

{

danhsachnv.docfile();

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu9;

break;

}

case 6:

{

Console.WriteLine("Nhap ma NhanVien muon tim: ");

string manv = Console.ReadLine();

danhsachnv.Tim(manv);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu9;

break;

}

case 7:

{

danhsachnv.CapNhat(k);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu9;

break;

}

}

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

} while (nv != 0);

Console.WriteLine("Ban co muon tiep tuc?(1: co, 2: khong)");

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

if (tt == 1)

goto menu;

break;

}

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

} while (chon != 0);

}

}

}

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace QuanLySach

{

public class Sach

{

private string m\_masach;

private string m\_tensach;

private string m\_dsmatg;

private string m\_manxb;

private double m\_gia;

private string m\_dsmatl;

public string masach

{

get { return m\_masach; }

set { m\_masach = value; }

}

public string tensach

{

get { return m\_tensach; }

set { m\_tensach = value; }

}

public string dsmatg

{

get { return m\_dsmatg; }

set { m\_dsmatg = value; }

}

public string manxb

{

get { return m\_manxb; }

set { m\_manxb = value; }

}

public double gia

{

get { return m\_gia; }

set { m\_gia = value; }

}

public string dsmatl

{

get { return m\_dsmatl; }

set { m\_dsmatl = value; }

}

public Sach(string masach, string tensach, string dsmatg, string manxb, double gia, string dsmatl)

{

m\_masach = masach;

m\_tensach = tensach;

m\_dsmatg = dsmatg;

m\_manxb = manxb;

m\_gia = gia;

m\_dsmatl = dsmatl;

}

public Sach()

{

m\_masach = " ";

m\_tensach = " ";

m\_dsmatg = " ";

m\_manxb = " ";

m\_gia = 0;

m\_dsmatl = " ";

}

public void Nhap()

{

Console.Write("Ma sach: ");

masach = Console.ReadLine();

Console.Write("Ten sach: ");

tensach = Console.ReadLine();

Console.Write("Danh sach ma tac gia: ");

dsmatg = Console.ReadLine();

Console.Write("Ma ngay xuat ban:");

manxb = Console.ReadLine();

Console.Write("Gia: ");

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

Console.Write("Danh sach ma the loai: ");

dsmatl = Console.ReadLine();

}

public void Xuat()

{

DanhsachSach ds = new DanhsachSach();

Console.WriteLine("MaSach: " + masach);

Console.WriteLine("TenSach: " + tensach);

Console.WriteLine("DSMaTacGia: " + dsmatg);

Console.WriteLine("MaNgayXuatBan: " + manxb);

Console.WriteLine("Gia: " + gia);

Console.WriteLine("DSMaTheLoai " + dsmatl);

}

}

}

+++++++++++++++++++++++++++++++++++

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

using System.Runtime.Serialization.Formatters.Binary;

namespace QuanLySach

{

public class DanhsachSach

{

Sach sach = new Sach();

List<Sach> dssach = new List<Sach>();

public void Them(Sach a )

{

sach.Nhap();

dssach.Add(sach);

}

public Sach Tim(string masach)

{

foreach (Sach a in dssach)

{

if (a.masach == masach)

{

a.Xuat();

return a;

}

}

return null;

}

public int TimViTri(string masach)

{

int vitri = 0;

foreach (Sach a in dssach)

{

if (a.masach == masach)

{

vitri = int.Parse(a.masach);

}

}

return vitri;

}

public void CapNhat(Sach a)

{

DanhsachSach ds = new DanhsachSach();

Console.WriteLine("Nhap ma sach muon sua: ");

string masach = Console.ReadLine();

int vitri = TimViTri(masach);

dssach.RemoveAt(vitri);

Console.WriteLine("Nhap lai thong tin sach!");

Them(a);

ghiflie();

}

public void Xoa(Sach a)

{

DanhsachSach ds = new DanhsachSach();

Console.WriteLine("Nhap ma sach muon xoa: ");

string masach = Console.ReadLine();

int vitri = TimViTri(masach);

dssach.RemoveAt(vitri);

ghiflie();

}

public void docfile()

{

string[] allLines = File.ReadAllLines("sach.txt");

for(int i=0;i<allLines.Length;i++)

{

string line = allLines[i];

Sach sach = new Sach();

string[] temp = line.Split("\t".ToCharArray(), StringSplitOptions.RemoveEmptyEntries);

if(temp.Length >0)

{

sach.masach = temp[0];

sach.tensach = temp[1];

sach.dsmatg = temp[2];

sach.manxb = temp[3];

sach.gia = double.Parse(temp[4]);

sach.dsmatl = temp[5];

dssach.Add(sach);

}

sach.Xuat();

}

}

public void ghiflie()

{

StreamWriter swt = new StreamWriter("sach.txt", false, Encoding.UTF8);

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

{

Sach sach = new Sach();

sach = dssach[i];

swt.WriteLine(sach.masach + "\t" + sach.tensach + "\t" + sach.dsmatg + "\t" + sach.manxb+ "\t"+ sach.gia+ "\t" + sach.dsmatl);

}

swt.Close();

}

}

}

++++++++++++++++++++++++++++++++++++++++++++

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace QuanLySach

{

public class TacGia

{

private string m\_MaTG;

private string m\_TenTG;

public string matg

{

get { return m\_MaTG; }

set { m\_MaTG = value; }

}

public string tentg

{

get { return m\_TenTG; }

set { m\_TenTG = value; }

}

public TacGia(string matg, string tentg)

{

m\_MaTG = matg;

m\_TenTG = tentg;

}

public TacGia()

{

m\_MaTG = " ";

m\_TenTG = " ";

}

public void Nhap()

{

Console.WriteLine("Nhap Ma Tac Gia : ");

matg = Console.ReadLine();

Console.WriteLine("Nhap Ten Tac Gia : ");

tentg = Console.ReadLine();

}

public void Xuat()

{

Console.WriteLine("Ma Tac Gia : " + matg);

Console.WriteLine("Ten Tac Gia : " + tentg);

}

}

}

+++++++++++++++++++++++++++++++++++++++

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

using System.Runtime.Serialization.Formatters.Binary;

namespace QuanLySach

{

public class DanhSachTacGia

{

TacGia tacgia = new TacGia();

List<TacGia> dstacgia = new List<TacGia>();

public void Them(TacGia a)

{

tacgia.Nhap();

dstacgia.Add(tacgia);

}

public TacGia Tim(string matg)

{

foreach (TacGia a in dstacgia)

{

if (a.matg == matg)

{

a.Xuat();

return a;

}

}

return null;

}

public int TimViTri(string matacgia)

{

int vitri = 0;

foreach (TacGia a in dstacgia)

{

if (a.matg == matacgia)

{

vitri = int.Parse(a.matg);

}

}

return vitri;

}

public void CapNhat(TacGia a)

{

DanhSachTacGia ds = new DanhSachTacGia();

Console.WriteLine("Nhap ma tac gia muon sua: ");

string matg = Console.ReadLine();

int vitri = TimViTri(matg);

dstacgia.RemoveAt(vitri);

Console.WriteLine("Nhap lai thong tin sach!");

Them(a);

ghiflie();

}

public void Xoa(TacGia a)

{

DanhSachTacGia ds = new DanhSachTacGia();

Console.WriteLine("Nhap ma tac gia muon xoa: ");

string matg = Console.ReadLine();

int vitri = TimViTri(matg);

dstacgia.RemoveAt(vitri);

ghiflie();

}

public void docfile()

{

string[] allLines = File.ReadAllLines("TacGia.txt");

for (int i = 0; i < allLines.Length; i++)

{

string line = allLines[i];

TacGia tacgia = new TacGia();

string[] temp = line.Split("\t".ToCharArray(), StringSplitOptions.RemoveEmptyEntries);

if (temp.Length > 0)

{

tacgia.matg = temp[0];

tacgia.tentg = temp[1];

dstacgia.Add(tacgia);

}

tacgia.Xuat();

}

}

public void ghiflie()

{

StreamWriter swt = new StreamWriter("TacGia.txt", false, Encoding.UTF8);

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

{

TacGia tacgia = new TacGia();

tacgia = dstacgia[i];

swt.WriteLine(tacgia.matg + "\t" + tacgia.tentg );

}

swt.Close();

}

}

}

++++++++++++++++++++++++++++++++++++++++

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace QuanLySach

{

public class TheLoai

{

private string m\_MaTL;

private string m\_TenTL;

public string matl

{

get { return m\_MaTL; }

set { m\_MaTL = value; }

}

public string tentl

{

get { return m\_TenTL; }

set { m\_TenTL = value; }

}

public TheLoai(string matl, string tentl)

{

m\_MaTL = matl;

m\_TenTL = tentl;

}

public TheLoai()

{

m\_MaTL = " ";

m\_TenTL = " ";

}

public void Nhap()

{

Console.WriteLine("Nhap Ma The Loai : ");

matl = Console.ReadLine();

Console.WriteLine("Nhap Ten The Loai : ");

tentl = Console.ReadLine();

}

public void Xuat()

{

Console.WriteLine("Ma The Loai : " + matl);

Console.WriteLine("Ten The Loai : " + tentl);

}

public List<TheLoai> dstl = new List<TheLoai>();

public void Them(TheLoai a)

{

dstl.Add(a);

a.Nhap();

}

public void Xoa(TheLoai a)

{

Console.WriteLine("Nhap ma sach muon xoa: ");

string masach = Console.ReadLine();

dstl.Remove(a);

}

public void CapNhat(TheLoai a)

{

Console.WriteLine("Nhap ma sach muon sua: ");

string masach = Console.ReadLine();

Console.WriteLine("Nhap lai thong tin sach!");

Nhap();

dstl.Add(a);

}

public TheLoai tim(string masach)

{

foreach (TheLoai a in dstl)

{

if (a.matl == masach)

{

a.Xuat();

}

}

return null;

}

}

}

++++++++++++++++++++++++++++++++++++++++++++++++++

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

using System.Runtime.Serialization.Formatters.Binary;

namespace QuanLySach

{

public class DanhSachTheLoai

{

TheLoai theloai = new TheLoai();

List<TheLoai> dstheloai = new List<TheLoai>();

public void Them(TheLoai a)

{

theloai.Nhap();

dstheloai.Add(theloai);

}

public TheLoai Tim(string matl)

{

foreach (TheLoai a in dstheloai)

{

if (a.matl == matl)

{

a.Xuat();

return a;

}

}

return null;

}

public int TimViTri(string matl)

{

int vitri = 0;

foreach (TheLoai a in dstheloai)

{

if (a.matl == matl)

{

vitri = int.Parse(a.matl);

}

}

return vitri;

}

public void CapNhat(TheLoai a)

{

DanhSachTheLoai ds = new DanhSachTheLoai();

Console.WriteLine("Nhap ma theloai muon sua: ");

string matl = Console.ReadLine();

int vitri = TimViTri(matl);

dstheloai.RemoveAt(vitri);

Console.WriteLine("Nhap lai thong tin theloai!");

Them(a);

ghiflie();

}

public void Xoa(TheLoai a)

{

DanhSachTheLoai ds = new DanhSachTheLoai();

Console.WriteLine("Nhap ma theloai muon xoa: ");

string matl = Console.ReadLine();

int vitri = TimViTri(matl);

dstheloai.RemoveAt(vitri);

ghiflie();

}

public void docfile()

{

string[] allLines = File.ReadAllLines("theloai.txt");

for (int i = 0; i < allLines.Length; i++)

{

string line = allLines[i];

TheLoai theloai = new TheLoai();

string[] temp = line.Split("\t".ToCharArray(), StringSplitOptions.RemoveEmptyEntries);

if (temp.Length > 0)

{

theloai.matl = temp[0];

theloai.tentl = temp[1];

dstheloai.Add(theloai);

}

theloai.Xuat();

}

}

public void ghiflie()

{

StreamWriter swt = new StreamWriter("theloai.txt", false, Encoding.UTF8);

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

{

TheLoai theloai = new TheLoai();

theloai = dstheloai[i];

swt.WriteLine(theloai.matl + "\t" + theloai.tentl );

}

swt.Close();

}

}

}

++++++++++++++++++++++++++++++++++++++++++++++++

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace QuanLySach

{

public class NhaXuatBan

{

private string m\_MaNXB;

private string m\_TenNXB;

private string m\_TTLH;

public string manxb

{

get { return m\_MaNXB; }

set { m\_MaNXB = value; }

}

public string tennxb

{

get { return m\_TenNXB; }

set { m\_TenNXB = value; }

}

public string ttlh

{

get { return m\_TTLH; }

set { m\_TTLH = value; }

}

public NhaXuatBan(string manxb, string tennxb, string ttlh)

{

m\_MaNXB = manxb;

m\_TenNXB = tennxb;

m\_TTLH = ttlh;

}

public NhaXuatBan()

{

m\_MaNXB = " ";

m\_TenNXB = " ";

m\_TTLH = " ";

}

public void Nhap()

{

Console.WriteLine("Nhap Ma Nha Xuat Ban : ");

manxb = Console.ReadLine();

Console.WriteLine("Nhap Ten Nha Xuat Ban : ");

tennxb = Console.ReadLine();

Console.WriteLine("Nhap Thong Tin Lien He: ");

ttlh = Console.ReadLine();

}

public void Xuat()

{

Console.WriteLine("Ma Nha Xuat Ban : " + manxb);

Console.WriteLine("Ten Nha Xuat Ban : " + tennxb);

Console.WriteLine("Thong Tin Lien He: " + ttlh);

}

public List<NhaXuatBan> dsnxb = new List<NhaXuatBan>();

public void Them(NhaXuatBan a)

{

dsnxb.Add(a);

a.Nhap();

}

public void Xoa(NhaXuatBan a)

{

Console.WriteLine("Nhap ma sach muon xoa: ");

string masach = Console.ReadLine();

dsnxb.Remove(a);

}

public void CapNhat(NhaXuatBan a)

{

Console.WriteLine("Nhap ma sach muon sua: ");

string masach = Console.ReadLine();

Console.WriteLine("Nhap lai thong tin sach!");

Nhap();

dsnxb.Add(a);

}

public NhaXuatBan tim(string masach)

{

foreach (NhaXuatBan a in dsnxb)

{

if (a.manxb == manxb)

{

a.Xuat();

}

}

return null;

}

}

}

++++++++++++++++++++++++++++++++++++++++++++++++++++++

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

using System.Runtime.Serialization.Formatters.Binary;

namespace QuanLySach

{

public class DanhsachNXB

{

NhaXuatBan NXB = new NhaXuatBan();

List<NhaXuatBan> dsNXB = new List<NhaXuatBan>();

public void Them(NhaXuatBan a)

{

NXB.Nhap();

dsNXB.Add(NXB);

}

public NhaXuatBan Tim(string manxb)

{

foreach (NhaXuatBan a in dsNXB)

{

if (a.manxb == manxb)

{

a.Xuat();

return a;

}

}

return null;

}

public int TimViTri(string manxb)

{

int vitri = 0;

foreach (NhaXuatBan a in dsNXB)

{

if (a.manxb == manxb)

{

vitri = int.Parse(a.manxb);

}

}

return vitri;

}

public void CapNhat(NhaXuatBan a)

{

DanhsachNXB ds = new DanhsachNXB();

Console.WriteLine("Nhap ma NhaXuatBan muon sua: ");

string manxb = Console.ReadLine();

int vitri = TimViTri(manxb);

dsNXB.RemoveAt(vitri);

Console.WriteLine("Nhap lai thong tin NhaXuatBan!");

Them(a);

ghiflie();

}

public void Xoa(NhaXuatBan a)

{

DanhsachNXB ds = new DanhsachNXB();

Console.WriteLine("Nhap ma NhaXuatBan muon xoa: ");

string manxb = Console.ReadLine();

int vitri = TimViTri(manxb);

dsNXB.RemoveAt(vitri);

ghiflie();

}

public void docfile()

{

string[] allLines = File.ReadAllLines("NXB.txt");

for (int i = 0; i < allLines.Length; i++)

{

string line = allLines[i];

NhaXuatBan NXB = new NhaXuatBan();

string[] temp = line.Split("\t".ToCharArray(), StringSplitOptions.RemoveEmptyEntries);

if (temp.Length > 0)

{

NXB.manxb = temp[0];

NXB.tennxb = temp[1];

NXB.ttlh = temp[2];

dsNXB.Add(NXB);

}

NXB.Xuat();

}

}

public void ghiflie()

{

StreamWriter swt = new StreamWriter("NXB.txt", false, Encoding.UTF8);

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

{

NhaXuatBan NXB = new NhaXuatBan();

NXB = dsNXB[i];

swt.WriteLine(NXB.manxb + "\t" + NXB.tennxb + "\t" + NXB.ttlh);

}

swt.Close();

}

}

}

++++++++++++++++++++++++++++++++++++++++++++++

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace QuanLySach

{

public class HoaDon

{

private string m\_mahd;

private string m\_manv;

private string m\_makh;

private DateTime m\_ngayxuat;

private string m\_thanhtoan;

private int m\_thanhtien;

public string mahd

{

get { return m\_mahd; }

set { m\_mahd = value; }

}

public string manv

{

get { return m\_manv; }

set { m\_manv = value; }

}

public string makh

{

get { return m\_makh; }

set { m\_makh = value; }

}

public DateTime ngayxuat

{

get { return m\_ngayxuat; }

set { m\_ngayxuat = value; }

}

public string thanhtoan

{

get { return m\_thanhtoan; }

set { m\_thanhtoan = value; }

}

public int thanhtien

{

get { return m\_thanhtien; }

set { m\_thanhtien = value; }

}

public HoaDon(string mahd, string manv, string makh, DateTime ngayxuat, string thanhtoan, int thanhtien)

{

m\_mahd = mahd;

m\_manv = manv;

m\_makh = makh;

m\_ngayxuat = ngayxuat;

m\_thanhtien = thanhtien;

m\_thanhtoan = thanhtoan;

}

public HoaDon()

{

m\_mahd = " ";

m\_manv = " ";

m\_makh = " ";

m\_ngayxuat = DateTime.Now;

m\_thanhtoan = " ";

m\_thanhtien = 0;

}

public void Nhap()

{

Console.WriteLine("ma hoa don: ");

mahd = Console.ReadLine();

Console.WriteLine("ma khach hang: ");

makh = Console.ReadLine();

Console.WriteLine("ma nhhan vien: ");

manv = Console.ReadLine();

Console.WriteLine("ngay xuat hoa don:");

ngayxuat = DateTime.Parse(Console.ReadLine());

Console.WriteLine("Thanh toan: ");

thanhtoan = Console.ReadLine();

Console.WriteLine("thanh tien: ");

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

}

public void Xuat()

{

Console.WriteLine("Mahd: " + mahd);

Console.WriteLine("MaNV: " + manv);

Console.WriteLine("Makh: " + makh);

Console.WriteLine("Ngay xuat hoa don " + ngayxuat);

Console.WriteLine("HInh thuc thanh toan" + thanhtoan);

Console.WriteLine("Thanh tien " + thanhtien);

}

}

}

++++++++++++++++++++++++++++++++++++++++++

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

using System.Runtime.Serialization.Formatters.Binary;

namespace QuanLySach

{

public class DanhsachHoaDon

{

HoaDon hoadon = new HoaDon();

List<HoaDon> dshoadon = new List<HoaDon>();

public void Them(HoaDon a)

{

hoadon.Nhap();

dshoadon.Add(hoadon);

}

public HoaDon Tim(string mahd)

{

foreach (HoaDon a in dshoadon)

{

if (a.mahd == mahd)

{

a.Xuat();

return a;

}

}

return null;

}

public int TimViTri(string mahd)

{

int vitri = 0;

foreach (HoaDon a in dshoadon)

{

if (a.mahd == mahd)

{

vitri = int.Parse(a.mahd);

}

}

return vitri;

}

public void CapNhat(HoaDon a)

{

DanhsachHoaDon ds = new DanhsachHoaDon();

Console.WriteLine("Nhap ma HoaDon muon sua: ");

string mahd = Console.ReadLine();

int vitri = TimViTri(mahd);

dshoadon.RemoveAt(vitri);

Console.WriteLine("Nhap lai thong tin HoaDon!");

Them(a);

ghiflie();

}

public void Xoa(HoaDon a)

{

DanhsachHoaDon ds = new DanhsachHoaDon();

Console.WriteLine("Nhap ma HoaDon muon xoa: ");

string mahd = Console.ReadLine();

int vitri = TimViTri(mahd);

dshoadon.RemoveAt(vitri);

ghiflie();

}

public void docfile()

{

string[] allLines = File.ReadAllLines("hoadon.txt");

for (int i = 0; i < allLines.Length; i++)

{

string line = allLines[i];

HoaDon hoadon = new HoaDon();

string[] temp = line.Split("\t".ToCharArray(), StringSplitOptions.RemoveEmptyEntries);

if (temp.Length > 0)

{

hoadon.mahd = temp[0];

hoadon.manv = temp[1];

hoadon.makh = temp[2];

hoadon.ngayxuat = DateTime.Parse(temp[3]);

hoadon.thanhtoan= temp[4];

hoadon.thanhtien = int.Parse(temp[5]);

dshoadon.Add(hoadon);

}

hoadon.Xuat();

}

}

public void ghiflie()

{

StreamWriter swt = new StreamWriter("hoadon.txt", false, Encoding.UTF8);

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

{

HoaDon sach = new HoaDon();

sach = dshoadon[i];

swt.WriteLine(hoadon.mahd + "\t" + hoadon.manv + "\t" + hoadon.makh + "\t" + hoadon.ngayxuat + "\t" + hoadon.thanhtoan + "\t" + hoadon.thanhtien);

}

swt.Close();

}

}

}

+++++++++++++++++++++++++++++++++++

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace QuanLySach

{

public class ChiTietHoaDon

{

private string m\_mahd;

private string m\_masach;

private string m\_tensach;

private int m\_soluong;

private int m\_gia;

public string mahd

{

get { return m\_mahd; }

set { m\_mahd = value; }

}

public string masach

{

get { return m\_masach; }

set { m\_masach = value; }

}

public string tensach

{

get { return m\_tensach; }

set { m\_tensach = value; }

}

public int soluong

{

get { return m\_soluong; }

set { m\_soluong = value; }

}

public int gia

{

get { return m\_gia; }

set { m\_gia = value; }

}

public ChiTietHoaDon(string mahd, string masach, string tensach, int soluong, int gia)

{

m\_mahd = mahd;

m\_masach = masach;

m\_tensach = tensach;

m\_gia = gia;

m\_soluong = soluong;

}

public ChiTietHoaDon()

{

mahd = " ";

masach = " ";

tensach = " ";

gia = 0;

soluong = 0;

}

public void Nhap()

{

Console.WriteLine("ma hoa don: ");

mahd = Console.ReadLine();

Console.WriteLine("ma sach : ");

masach = Console.ReadLine();

Console.WriteLine("ten sach: ");

tensach = Console.ReadLine();

Console.WriteLine("Gia : ");

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

Console.WriteLine("So luong sach: ");

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

}

public void Xuat()

{

Console.WriteLine("Mahd: " + mahd);

Console.WriteLine("Masach: " + masach);

Console.WriteLine("ten sach: " + tensach);

Console.WriteLine("Gia: " + gia);

Console.WriteLine("So luong sach: " + soluong);

}

private List<ChiTietHoaDon> dsct;

public List<ChiTietHoaDon> DanhsachHoaDon

{

get { return dsct; }

}

public ChiTietHoaDon tim(string mahd)

{

foreach (ChiTietHoaDon a in dsct)

{

if (a.mahd == mahd) return a;

}

return null;

}

public void them(ChiTietHoaDon a)

{

dsct.Add(a);

}

public void xoa(string mshs)

{

ChiTietHoaDon a = tim(mshs);

if (a != null) dsct.Remove(a);

}

public void sua(ChiTietHoaDon a)

{

ChiTietHoaDon x = tim(a.mahd);

if (x != null)

{

x.mahd = a.mahd;

x.masach = a.masach;

x.tensach = a.tensach;

x.gia = a.gia;

x.soluong = a.soluong;

}

}

}

}

+++++++++++++++++++++++++++++++++++++++++++++++++++++

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

using System.Runtime.Serialization.Formatters.Binary;

namespace QuanLySach

{

public class DSChiTietHoaDon

{

ChiTietHoaDon cthd = new ChiTietHoaDon();

List<ChiTietHoaDon> dscthd = new List<ChiTietHoaDon>();

public void Them(ChiTietHoaDon a)

{

cthd.Nhap();

dscthd.Add(cthd);

}

public ChiTietHoaDon Tim(string mahd)

{

foreach (ChiTietHoaDon a in dscthd)

{

if (a.mahd == mahd)

{

a.Xuat();

return a;

}

}

return null;

}

public int TimViTri(string mahd)

{

int vitri = 0;

foreach (ChiTietHoaDon a in dscthd)

{

if (a.mahd == mahd)

{

vitri = int.Parse(a.masach);

}

}

return vitri;

}

public void CapNhat(ChiTietHoaDon a)

{

DSChiTietHoaDon ds = new DSChiTietHoaDon();

Console.WriteLine("Nhap ma sach muon sua: ");

string mahd = Console.ReadLine();

int vitri = TimViTri(mahd);

dscthd.RemoveAt(vitri);

Console.WriteLine("Nhap lai thong tin sach!");

Them(a);

ghiflie();

}

public void Xoa(ChiTietHoaDon a)

{

DSChiTietHoaDon ds = new DSChiTietHoaDon();

Console.WriteLine("Nhap ma sach muon xoa: ");

string mahd = Console.ReadLine();

int vitri = TimViTri(mahd);

dscthd.RemoveAt(vitri);

ghiflie();

}

public void docfile()

{

string[] allLines = File.ReadAllLines("cthd.txt");

for (int i = 0; i < allLines.Length; i++)

{

string line = allLines[i];

ChiTietHoaDon cthd = new ChiTietHoaDon();

string[] temp = line.Split("\t".ToCharArray(), StringSplitOptions.RemoveEmptyEntries);

if (temp.Length > 0)

{

cthd.mahd = temp[0];

cthd.masach = temp[1];

cthd.tensach = temp[2];

cthd.soluong = int.Parse(temp[3]);

cthd.gia = int.Parse(temp[4]);

dscthd.Add(cthd);

}

cthd.Xuat();

}

}

public void ghiflie()

{

StreamWriter swt = new StreamWriter("cthd.txt", false, Encoding.UTF8);

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

{

ChiTietHoaDon sach = new ChiTietHoaDon();

sach = dscthd[i];

swt.WriteLine(cthd.mahd + "\t" + cthd.masach + "\t" + cthd.tensach + "\t" + cthd.soluong + "\t" + sach.gia);

}

swt.Close();

}

}

}

++++++++++++++++++++++++++++++++++++++++++++++++++

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace QuanLySach

{

public class KhachHang

{

private string \_makh;

private string \_tenkh;

private string \_diachi;

private string \_sdt;

private string \_gioitinh;

public string makh

{

get { return \_makh; }

set { \_makh = value; }

}

public string tenkh

{

get { return \_tenkh; }

set { \_tenkh = value; }

}

public string diachi

{

get { return \_diachi; }

set { \_diachi = value; }

}

public string sdt

{

get { return \_sdt; }

set { \_sdt = value; }

}

public string gioitinh

{

get { return \_gioitinh; }

set { \_gioitinh = value; }

}

public KhachHang(string makh, string tenkh, string diachi, string sdt, string gioitinh)

{

\_makh = makh;

\_tenkh = tenkh;

\_diachi = diachi;

\_sdt = sdt;

\_gioitinh = gioitinh;

}

public KhachHang()

{

\_makh = "";

\_tenkh = "";

\_diachi = "";

\_sdt = "";

\_gioitinh = "";

}

public void Nhap()

{

Console.WriteLine("Nhap Thông tin Khach Hang:");

Console.WriteLine("\nMa Khach Hang: ");

makh = Console.ReadLine();

Console.WriteLine("Ten Khach Hang: ");

tenkh = Console.ReadLine();

Console.WriteLine("Dia Chi: ");

diachi = Console.ReadLine();

Console.WriteLine("SDT: ");

sdt = Console.ReadLine();

Console.WriteLine("Gioi Tinh:");

gioitinh = Console.ReadLine();

}

public void Xuat()

{

Console.WriteLine("\nThông tin Khach Hang");

Console.WriteLine("MaKH: " + makh);

Console.WriteLine("TenKh:" + tenkh);

Console.WriteLine("Dia Chi: " + diachi);

Console.WriteLine("SDT:" + sdt);

Console.WriteLine("Gioi Tinh:" + gioitinh);

}

}

}

+++++++++++++++++++++++++++++++++++++++++

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

using System.Runtime.Serialization.Formatters.Binary;

namespace QuanLySach

{

class DanhSachKhachHang

{

KhachHang khach = new KhachHang();

List<KhachHang> dskhach = new List<KhachHang>();

public void Them(KhachHang a)

{

khach.Nhap();

dskhach.Add(khach);

}

public KhachHang Tim(string masach)

{

foreach (KhachHang a in dskhach)

{

if (a.makh == masach)

{

a.Xuat();

return a;

}

}

return null;

}

public int TimViTri(string makh)

{

int vitri = 0;

foreach (KhachHang a in dskhach)

{

if (a.makh == makh)

{

vitri = int.Parse(a.makh);

}

}

return vitri;

}

public void CapNhat(KhachHang a)

{

DanhsachSach ds = new DanhsachSach();

Console.WriteLine("Nhap ma sach muon sua: ");

string masach = Console.ReadLine();

int vitri = TimViTri(masach);

dskhach.RemoveAt(vitri);

Console.WriteLine("Nhap lai thong tin sach!");

Them(a);

ghiflie();

}

public void Xoa(KhachHang a)

{

DanhsachSach ds = new DanhsachSach();

Console.WriteLine("Nhap ma sach muon xoa: ");

string manv = Console.ReadLine();

int vitri = TimViTri(manv);

dskhach.RemoveAt(vitri);

ghiflie();

}

public void docfile()

{

string[] allLines = File.ReadAllLines("khachhang.txt");

for (int i = 0; i < allLines.Length; i++)

{

string line = allLines[i];

KhachHang khach = new KhachHang();

string[] temp = line.Split("\t".ToCharArray(), StringSplitOptions.RemoveEmptyEntries);

if (temp.Length > 0)

{

khach.makh = temp[0];

khach.tenkh = temp[1];

khach.gioitinh = temp[2];

khach.diachi = temp[3];

khach.sdt = temp[4];

dskhach.Add(khach);

}

khach.Xuat();

}

}

public void ghiflie()

{

StreamWriter swt = new StreamWriter("khachhang.txt", false, Encoding.UTF8);

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

{

KhachHang khach = new KhachHang();

khach = dskhach[i];

swt.WriteLine(khach.makh + "\t" + khach.tenkh + "\t" + khach.gioitinh + "\t" + khach.diachi + "\t" + khach.sdt);

}

swt.Close();

}

}

}

+++++++++++++++++++++++++++++++++++++

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace QuanLySach

{

public class Kho

{

private string \_makho;

private string \_tenkho;

private int \_sonhap;

private int \_sonton;

public string makho

{

get { return \_makho; }

set { \_makho = value; }

}

public string tenkho

{

get { return \_tenkho; }

set { \_tenkho = value; }

}

public int sonhap

{

get { return \_sonhap; }

set { \_sonhap = value; }

}

public int soton

{

get { return \_sonton; }

set { \_sonton = value; }

}

public Kho(string makho, string tenkho, int sonhap, int soton)

{

\_makho = makho;

\_tenkho = tenkho;

\_sonhap = sonhap;

\_sonton = soton;

}

public Kho()

{

\_makho = "";

\_tenkho = "";

\_sonhap = 0;

\_sonton = 0;

}

public void Nhap()

{

Console.WriteLine("Ma Kho: ");

makho = Console.ReadLine();

Console.WriteLine("Ten Kho: ");

tenkho = Console.ReadLine();

Console.WriteLine("So Luong Nhap:");

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

Console.WriteLine("So Luong Ton Kho:");

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

}

public void Xuat()

{

Console.WriteLine("Mã Kho : " + makho);

Console.WriteLine("Tên Kho:" + tenkho);

Console.WriteLine("So Luong Nhap; " + sonhap);

Console.WriteLine("So Luong Ton Kho:" + soton);

}

}

}

++++++++++++++++++++++++++++++++++++

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

using System.Runtime.Serialization.Formatters.Binary;

namespace QuanLySach

{

public class DanhsachKho

{

Kho kho = new Kho();

List<Kho> dskho = new List<Kho>();

public void Them(Kho a)

{

kho.Nhap();

dskho.Add(kho);

}

public Kho Tim(string makho)

{

foreach (Kho a in dskho)

{

if (a.makho == makho)

{

a.Xuat();

return a;

}

}

return null;

}

public int TimViTri(string makho)

{

int vitri = 0;

foreach (Kho a in dskho)

{

if (a.makho == makho)

{

vitri = int.Parse(a.makho);

}

}

return vitri;

}

public void CapNhat(Kho a)

{

DanhsachSach ds = new DanhsachSach();

Console.WriteLine("Nhap ma sach muon sua: ");

string masach = Console.ReadLine();

int vitri = TimViTri(masach);

dskho.RemoveAt(vitri);

Console.WriteLine("Nhap lai thong tin sach!");

Them(a);

ghiflie();

}

public void Xoa(Kho a)

{

DanhsachSach ds = new DanhsachSach();

Console.WriteLine("Nhap ma sach muon xoa: ");

string masach = Console.ReadLine();

int vitri = TimViTri(masach);

dskho.RemoveAt(vitri);

ghiflie();

}

public void docfile()

{

string[] allLines = File.ReadAllLines("sach.txt");

for (int i = 0; i < allLines.Length; i++)

{

string line = allLines[i];

Kho sach = new Kho();

string[] temp = line.Split("\t".ToCharArray(), StringSplitOptions.RemoveEmptyEntries);

if (temp.Length > 0)

{

kho.makho = temp[0];

kho.tenkho = temp[1];

kho.sonhap = int.Parse(temp[2]);

kho.soton = int.Parse(temp[3]);

dskho.Add(kho);

}

sach.Xuat();

}

}

public void ghiflie()

{

StreamWriter swt = new StreamWriter("sach.txt", false, Encoding.UTF8);

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

{

Kho kho = new Kho();

kho = dskho[i];

swt.WriteLine(kho.makho + "\t" + kho.tenkho + "\t" + kho.sonhap + "\t" + kho.soton);

}

swt.Close();

}

}

}

+++++++++++++++++++++++++++++++++++++++++

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

using System.Runtime.Serialization.Formatters.Binary;

namespace QuanLySach

{

public class NhanVien

{

private string m\_manv;

private string m\_hoten;

private DateTime m\_ngayvaolam;

private string m\_chucvu;

private string m\_gioitinh;

private string m\_diachi;

private string m\_sdt;

public string manv

{

get { return m\_manv; }

set { m\_manv = value; }

}

public string hoten

{

get { return m\_hoten; }

set { m\_hoten = value; }

}

public DateTime ngayvaolam

{

get { return m\_ngayvaolam; }

set { m\_ngayvaolam = value; }

}

public string chucvu

{

get { return m\_chucvu; }

set { m\_chucvu = value; }

}

public string gioitinh

{

get { return m\_gioitinh; }

set { m\_gioitinh = value; }

}

public string diachi

{

get { return m\_diachi; }

set { m\_diachi = value; }

}

public string sdt

{

get { return m\_sdt; }

set { m\_sdt = value; }

}

public NhanVien(string manv, string hoten, DateTime ngayvaolam, string chucvu, string gioitinh, string diachi, string sdt)

{

m\_manv = manv;

m\_gioitinh = gioitinh;

m\_diachi = diachi;

m\_chucvu = chucvu;

m\_ngayvaolam = ngayvaolam;

m\_sdt = sdt;

m\_hoten = hoten;

}

public NhanVien()

{

m\_manv = " ";

m\_hoten = " ";

m\_ngayvaolam = DateTime.Now;

m\_chucvu = " ";

m\_gioitinh = " ";

m\_diachi = " ";

m\_sdt = " ";

}

private List<NhanVien> dsnv;

public List<NhanVien> DanhsachNhanVien

{

get { return dsnv; }

}

public void ghiflie()

{

Console.WriteLine("Vi du minh hoa doc va ghi File trong C#");

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

string[] names = new string[] { "Tran Van A", "Nguyen Minh B" };

using (StreamWriter sw = new StreamWriter("textfile.txt"))

{

foreach (string s in names)

{

sw.WriteLine(s);

}

}

// doc va hien thi du lieu trong textfile.txt

string line = "";

using (StreamReader sr = new StreamReader("E:\\test.txt"))

{

while ((line = sr.ReadLine()) != null)

{

Console.WriteLine(line);

}

}

Console.ReadKey();

}

public void docfile()

{

FileStream fs = new FileStream("E:\\test.txt", FileMode.Open);

StreamReader rd = new StreamReader(fs, Encoding.UTF8);

String giatri = rd.ReadToEnd();// ReadLine() chỉ đọc 1 dòng đầu thoy, ReadToEnd là đọc hết

Console.WriteLine(giatri);

rd.Close();

Console.ReadLine();

}

public NhanVien tim(string manv)

{

foreach (NhanVien a in dsnv)

{

if (a.manv == manv) return a;

}

return null;

}

public void them(NhanVien a)

{

dsnv.Add(a);

}

public void xoa(string mshs)

{

NhanVien a = tim(mshs);

if (a != null) dsnv.Remove(a);

}

public void sua(NhanVien a)

{

NhanVien x = tim(a.manv);

if (x != null)

{

x.hoten = a.hoten;

x.ngayvaolam = a.ngayvaolam;

x.chucvu = a.chucvu;

x.gioitinh = a.gioitinh;

x.diachi = a.diachi;

x.diachi = a.diachi;

}

}

public void Nhap()

{

Console.WriteLine("manv: ");

manv = Console.ReadLine();

Console.WriteLine("hoten: ");

hoten = Console.ReadLine();

Console.WriteLine("ngayvaolam:");

ngayvaolam = DateTime.Parse(Console.ReadLine());

Console.WriteLine("chucvu: ");

chucvu = Console.ReadLine();

Console.WriteLine("diachi: ");

diachi = Console.ReadLine();

Console.WriteLine("sdt: ");

sdt = Console.ReadLine();

Console.WriteLine("gioitinh: ");

gioitinh = Console.ReadLine();

}

public void Xuat()

{

Console.WriteLine("MaNV: " + manv);

Console.WriteLine("THoTenNV: " + hoten);

Console.WriteLine("Ngay Vao Lam: " + ngayvaolam);

Console.WriteLine("Chuc Vu: " + chucvu);

Console.WriteLine("Dia Chi: " + diachi);

Console.WriteLine("SDT: " + sdt);

Console.WriteLine("Gioi Tinh: " + gioitinh);

}

}

}

++++++++++++++++++++++++++++++++++++++++++++

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

using System.Runtime.Serialization.Formatters.Binary;

namespace QuanLySach

{

class DanhSachNhanVien

{

NhanVien nv = new NhanVien();

List<NhanVien> dsnv = new List<NhanVien>();

public void Them(NhanVien a)

{

nv.Nhap();

dsnv.Add(nv);

}

public NhanVien Tim(string masach)

{

foreach (NhanVien a in dsnv)

{

if (a.manv == masach)

{

a.Xuat();

return a;

}

}

return null;

}

public int TimViTri(string manv)

{

int vitri = 0;

foreach (NhanVien a in dsnv)

{

if (a.manv == manv)

{

vitri = int.Parse(a.manv);

}

}

return vitri;

}

public void CapNhat(NhanVien a)

{

DanhsachSach ds = new DanhsachSach();

Console.WriteLine("Nhap ma sach muon sua: ");

string masach = Console.ReadLine();

int vitri = TimViTri(masach);

dsnv.RemoveAt(vitri);

Console.WriteLine("Nhap lai thong tin sach!");

Them(a);

ghiflie();

}

public void Xoa(NhanVien a)

{

DanhsachSach ds = new DanhsachSach();

Console.WriteLine("Nhap ma sach muon xoa: ");

string manv = Console.ReadLine();

int vitri = TimViTri(manv);

dsnv.RemoveAt(vitri);

ghiflie();

}

public void docfile()

{

string[] allLines = File.ReadAllLines("nv.txt");

for (int i = 0; i < allLines.Length; i++)

{

string line = allLines[i];

NhanVien nv = new NhanVien();

string[] temp = line.Split("\t".ToCharArray(), StringSplitOptions.RemoveEmptyEntries);

if (temp.Length > 0)

{

nv.manv = temp[0];

nv.hoten = temp[1];

nv.ngayvaolam = DateTime.Parse(temp[2]);

nv.chucvu = temp[3];

nv.gioitinh = temp[4];

nv.diachi = temp[5];

nv.sdt = temp[6];

dsnv.Add(nv);

}

nv.Xuat();

}

}

public void ghiflie()

{

StreamWriter swt = new StreamWriter("nv.txt", false, Encoding.UTF8);

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

{

NhanVien sach = new NhanVien();

nv = dsnv[i];

swt.WriteLine(nv + "\t" + nv.hoten + "\t" + nv.ngayvaolam + "\t" + nv.chucvu + "\t" + nv.gioitinh + "\t" + nv.diachi + "\t" + nv.sdt);

}

swt.Close();

}

}

}

1. **Phân công làm bài nhóm cụ thể**

* Trình Mỹ Duyên:

+ Phân công công việc cho từng người

+ Tổng hợp các nội dung: bài làm cá nhân và bài nhóm

+ Viết báo cáo cho nhóm

+ Làm nội dung tuần thứ 7 để phân công công việc

+ Code theo nội dung phân công

+ Chỉnh sửa và hoàn thiện bài làm

* Trương Ngọc Khánh:

+ Code theo nội dung phân công

+ Chỉnh sửa bài và hoàn thiện bài làm

* Cao Lê Tuấn Vũ:

+ Code theo nội dung phân công

+ Viết thêm chức năng đọc file, ghi file

* Lê Văn Huy:

+ Code theo nội dung phân công

+ In báo cáo cho nhóm

* Nguyễn Thị Thùy Nhung:

+ Code theo nội dung phân công

*\*Nội dung code được phân công:*

*+ Định nghĩa các phương thức tạo lập*

*+ Các chức năng thêm, xóa, sửa*

*+ Viết hàm nhập và xuất*

1. **Lời cảm ơn**

Nhóm em xin chân thành cảm ơn Thầy đã hướng dẫn và đồng hành cùng nhóm em trong 10 tuần vừa qua của môn Đồ án Tin Học. Nhóm em cũng mong Thầy chấm điểm nương tay một xíu ạ..