

LAB 7. VẬN DỤNG TÍNH KẾ THỪA VÀ ĐA HÌNH

THỜI LƯỢNG : 8 TIẾT

A. Mục tiêu

- Giúp sinh viên biết cách vận dụng những kiến thức đã học vào việc giải một bài toán thực tế.
- Sau khi hoàn thành bài thực hành này, sinh viên cần:
 - Nắm vững cách tạo cây phân cấp kế thừa
 - Nắm vững cách đọc dữ liệu từ tập tin văn bản
 - Hiểu rõ về lớp trừu tượng, phương thức trừu tượng và cách áp dụng
 - Hiểu cách tổ chức chương trình
 - Vận dụng được các khái niệm lớp, đối tượng, đa hình, kế thừa vào các bài toán thực tế
 - Thành thạo việc cài đặt các thao tác cơ bản như thêm, xóa, tìm kiếm, sắp xếp, ...

B. Yêu cầu

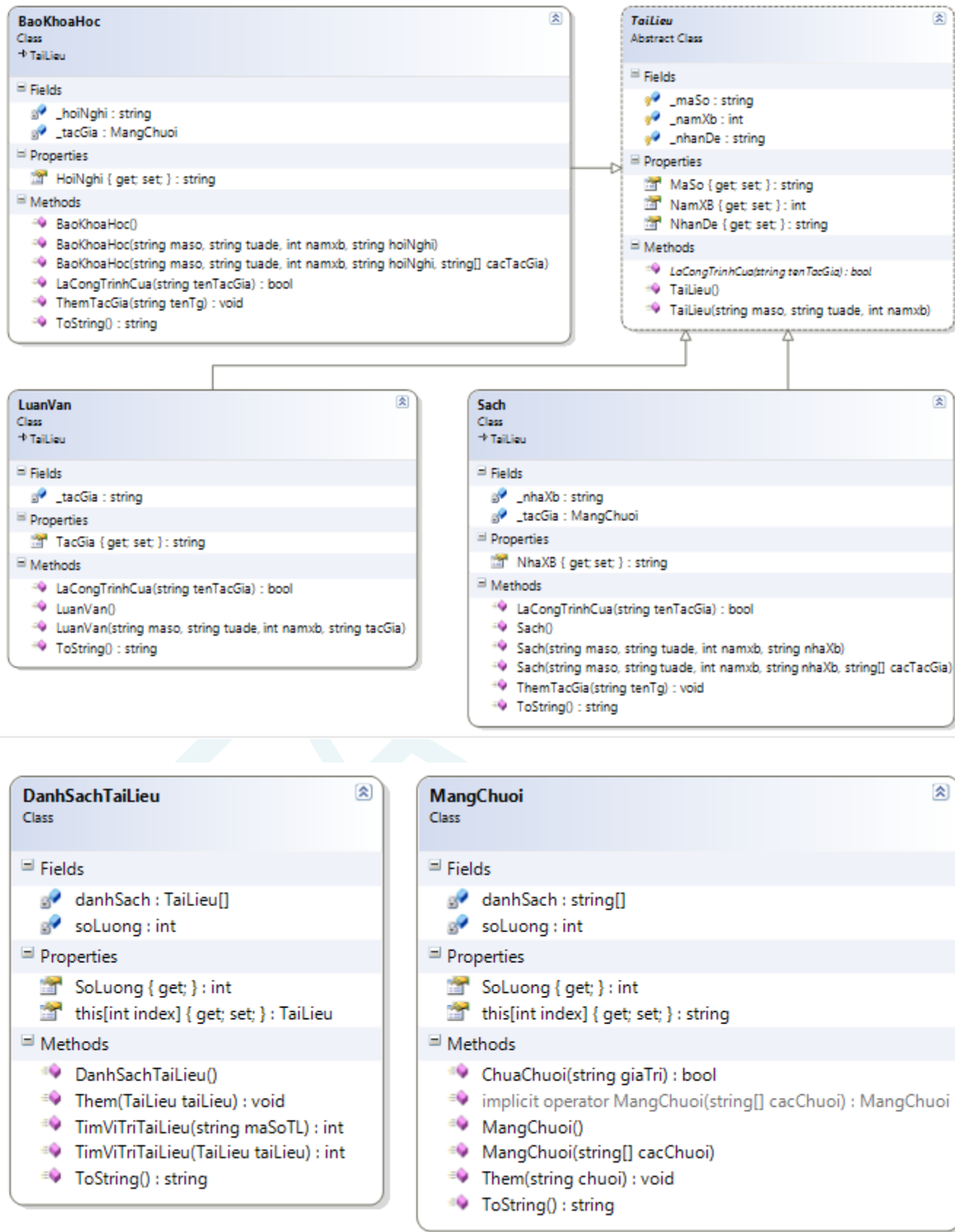
- Sinh viên phải trả lời đầy đủ các câu hỏi (nếu có) hoặc chụp màn hình kết quả, ghi lại vào tập tin Word theo yêu cầu ghi trong phần hướng dẫn thực hành.
- Đặt tên tập tin Word theo dạng: Lab07_MSSV_HoVaTen.rar.
- Tạo thư mục, đặt tên là MSSV_Lab07 để lưu tất cả bài làm trong bài thực hành này.
- Sinh viên phải hoàn thành tối thiểu 1 bài tập bắt buộc.
- Phải tạo một dự án riêng cho mỗi phần hoặc mỗi bài tập.
- Cuối buổi thực hành, nén thư mục trên & nộp bài qua email: phucnguyen5555@gmail.com

C. Hướng dẫn thực hành

Trong bài thực hành này, ta sẽ xây dựng một chương trình quản lý các loại tài liệu (bao gồm: Sách, Báo cáo khoa học, Luận văn) trong một thư viện. Ngoài ra, dữ liệu của chương trình có thể lấy từ tập tin văn bản (Text file) hoặc nhập từ bàn phím hoặc dữ liệu giả được cố định.

1. Tạo các lớp cơ sở

- Bước 1. Tạo một thư mục, đặt tên theo dạng MSSV-Lab07. Ví dụ: 1210234-Lab07.
- Bước 2. Tạo dự án Console Application mới, đặt tên là Lab07_Bai01_QuanLySachBao.
- Bước 3. Tạo và cài đặt các lớp theo thứ tự sau: MangChuoi (mảng lưu các chuỗi), TaiLieu (tài liệu nói chung), Sach (sách), BaoKhoaHoc (Bài báo khoa học), LuanVan (luận văn), DanhSachTaiLieu. Các lớp này được cho trong sơ đồ bên dưới.



Trong đó, lớp `TaiLieu` là lớp trừu tượng (abstract) và phương thức `LaCongTrinhCua(string tacGia)` là phương thức trừu tượng.

Bước 4. Trong hàm `Main`, nhập đoạn mã sau để kiểm tra kết quả.

```
// Tạo thể hiện của DanhSachTaiLieu
DanhSachTaiLieu thuVien = new DanhSachTaiLieu();

// Thêm tài liệu
thuVien.Them(new Sach("S001", "Lap trinh Web", 2008, "Lao Dong",
    new string[] {"Nguyen An, Dao Thi Bich"}));
thuVien.Them(new Sach("S203", "Photoshop", 2011, "Lao Dong",
    new string[] {"Phan Huy Duc"}));
thuVien.Them(new LuanVan("L100", "Khai thac luat ket hop", 2010, "Nguyen Thanh Tai"));
thuVien.Them(new BaoKhoaHoc("B440", "Hop chat Canxi cacbonat", 2013, "FACON2013",
    new string[] {"Hai Au", "Trinh Toan"} ));

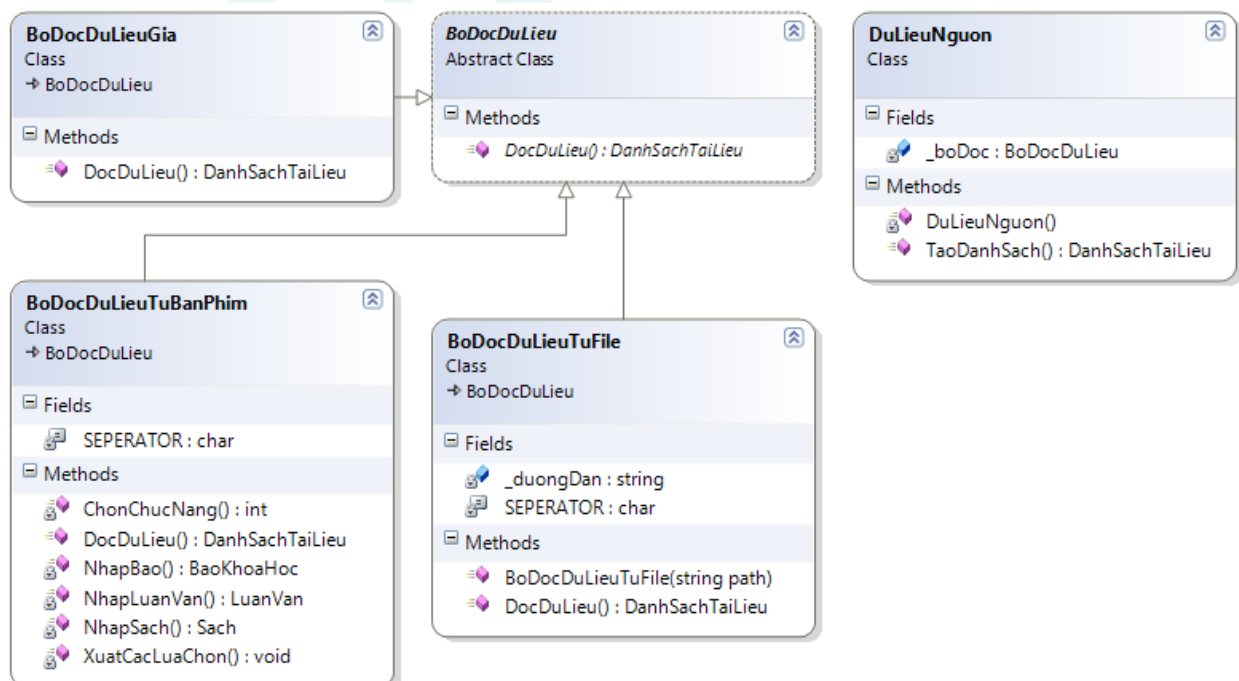
// xuất danh sách tài liệu
Console.WriteLine(thuVien);

Console.ReadKey();
```

Bước 5. Chạy chương trình và ghi nhận kết quả.

2. Cài đặt các lớp đọc và tạo dữ liệu mẫu

Phần này hướng dẫn cách tạo các lớp để đọc dữ liệu từ tập tin hoặc từ bàn phím để tạo thành một danh sách các tài liệu. Các lớp cần cài đặt được cho trong sơ đồ sau:

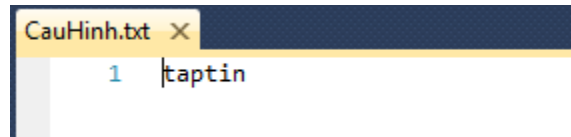


Trong đó, lớp BoDocDuLieu là lớp trừu tượng. Lớp này có một phương thức trừu tượng tên là DocDuLieu (đọc dữ liệu) và trả về một danh sách các tài liệu.

Bước 6. Tạo một thư mục mới trong dự án Lab07_Bai01_QuanLySachBao bằng cách nhấp phải chuột vào tên project, chọn Add > New Folder. Đặt tên là Data.

Bước 7. Trong thư mục này, tạo hai tập tin tên là CauHinh.txt và ThuVien.txt

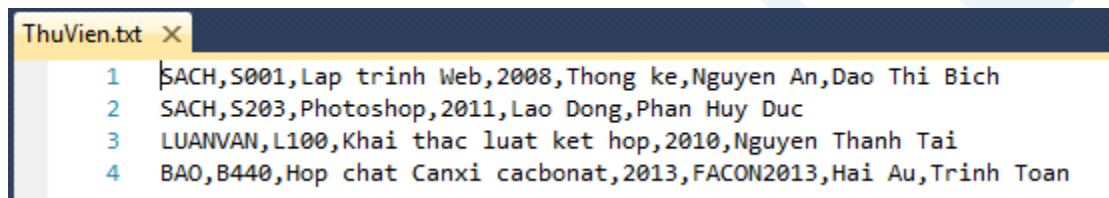
Bước 8. Trong tập tin CauHinh.txt, nhập nội dung sau



CauHinh.txt X

```
1 taptin
```

Bước 9. Trong tập tin ThuVien.txt, nhập nội dung sau



ThuVien.txt X

```
1 SACH,S001,Lap trinh Web,2008,Thong ke,Nguyen An,Dao Thi Bich
2 SACH,S203,Photoshop,2011,Lao Dong,Phan Huy Duc
3 LUANVAN,L100,Khai thac luat ket hop,2010,Nguyen Thanh Tai
4 BAO,B440,Hop chat Canxi cacbonat,2013,FACON2013,Hai Au,Trinh Toan
```

Bước 10. Trong cửa sổ Solution Explorer, chọn cả 2 tập tin. Nhấp phải chuột, chọn Properties.

Bước 11. Trong cửa sổ Properties: mục Build Action > chọn Content, mục Copy to Output Directory > chọn Copy if newer. Với thiết lập này, 2 tập tin trên sẽ tự động được chép vào thư mục Debug.

Bước 12. Tạo các lớp theo sơ đồ lớp ở trên.

Bước 13. Định nghĩa lớp BoDocDuLieu như sau:

```
// Lớp biểu diễn một bộ đọc dữ liệu tổng quát
// Ta sẽ tạo ra các lớp con để đọc dữ liệu từ
// tập tin, bàn phím hoặc tạo dữ liệu cố định
public abstract class BoDocDuLieu
{
    public abstract DanhSachTailieu DocDuLieu();
}
```

Bước 14. Định nghĩa lớp BoDocDuLieuGia như sau:

```
// Lớp tạo ra dữ liệu giả
public class BoDocDuLieuGia : BoDocDuLieu
{
    public override DanhSachTailieu DocDuLieu()
    {
        // Tạo thể hiện của DanhSachTailieu
        DanhSachTailieu thuVien = new DanhSachTailieu();

        // Thêm tài liệu
```

```

        thuVien.Them(new Sach("S001", "Lap trinh Web", 2008, "Lao Dong",
                                new string[] {"Nguyen An, Dao Thi Bich"}));
        thuVien.Them(new Sach("S203", "Photoshop", 2011, "Lao Dong",
                                new string[] {"Phan Huy Duc"}));
        thuVien.Them(new LuanVan("L100", "Khai thac luat ket hop",
                                   2010, "Nguyen Thanh Tai"));
        thuVien.Them(new BaoKhoaHoc("B440", "Hop chat Canxi cacbonat", 2013,
                                       "FACON2013", new string[] {"Hai Au", "Trinh Toan"}));

        return thuVien;
    }
}

```

Bước 15. Định nghĩa lớp BoDocDuLieuTuFile như sau:

```

public class BoDocDuLieuTuFile : BoDocDuLieu
{
    private const char SEPERATOR = ',';

    // Khai báo biến thành viên để lưu đường dẫn file
    private string _duongDan;

    // Định nghĩa phương thức khởi tạo
    public BoDocDuLieuTuFile(string path)
    {
        _duongDan = path;
    }

    public override DanhSachTaiLieu DocDuLieu()
    {
        DanhSachTaiLieu danhSach = new DanhSachTaiLieu();

        // Tạo luồng kết nối tới tập tin
        FileStream stream = new FileStream(_duongDan, FileMode.Open, FileAccess.Read);

        // Tạo đối tượng để đọc dữ liệu từ tập tin
        StreamReader reader = new StreamReader(stream);

        // đọc một dòng từ tập tin
        string line = reader.ReadLine();

        // Chừng nào còn đọc được dữ liệu thì tạo các đối tượng tương ứng
        while (!string.IsNullOrEmpty(line))
        {
            // Phân tách 1 chuỗi làm nhiều chuỗi
            string[] parts = line.Split(SEPERATOR);

            // Chuỗi đầu tiên cho biết loại tài liệu nên
            // ta kiểm tra nó để tạo đối tượng thích hợp
            switch (parts[0].ToUpper())
            {

```

```

        case "SACH":
            Sach cuonSach = new Sach(
                parts[1],           // mã số
                parts[2],           // tựa đề
                int.Parse(parts[3]), // năm xuất bản
                parts[4]            // nhà xuất bản
            );
            // Phần tử thứ 5 trở đi là tên tác giả
            for (int i = 5; i < parts.Length; i++)
            {
                cuonSach.ThemTacGia(parts[i].Trim());
            }
            danhSach.Them(cuonSach);
            break;

        case "LUANVAN":
            LuanVan baoCao = new LuanVan(
                parts[1],           // mã số
                parts[2],           // tựa đề
                int.Parse(parts[3]), // năm xuất bản
                parts[4]            // tên tác giả
            );
            danhSach.Them(baoCao);
            break;

        case "BAO":
            BaoKhoaHoc baiBao = new BaoKhoaHoc(
                parts[1],           // mã số
                parts[2],           // tựa đề
                int.Parse(parts[3]), // năm xuất bản
                parts[4]            // tên hội nghị
            );
            // Phần tử thứ 5 trở đi là tên tác giả
            for (int i = 5; i < parts.Length; i++)
            {
                baiBao.ThemTacGia(parts[i].Trim());
            }
            danhSach.Them(baiBao);
            break;
    }

    // Đọc dòng tiếp theo trong tập tin
    line = reader.ReadLine();
}
// Đóng bộ đọc dữ liệu
reader.Close();
// Đóng luồng kết nối tới tập tin
stream.Close();

return danhSach;
}
}

```

Bước 16. Định nghĩa lớp BoDocDuLieuTuBanPhim như sau:

```
// Lớp tạo danh sách tài liệu bằng cách nhập từ bàn phím
public class BoDocDuLieuTuBanPhim : BoDocDuLieu
{
    private const char SEPERATOR = ',';

    // Định nghĩa hàm hiển thị các chức năng
    private void XuatCacLuaChon()
    {
        Console.WriteLine("=====CHON CHUC NANG=====");
        Console.WriteLine("1. Nhap sach");
        Console.WriteLine("2. Nhap luan van");
        Console.WriteLine("3. Nhap bao khoa hoc");
        Console.WriteLine("0. Dung viec nhap du lieu");
        Console.WriteLine("=====");
        Console.WriteLine();
        Console.Write("Nhap mot so tu 0 den 3: ");
    }

    // định nghĩa hàm chọn chức năng nhập
    private int ChonChucNang()
    {
        int chon = 0;
        do
        {
            XuatCacLuaChon();
            chon = int.Parse(Console.ReadLine());
        } while (chon < 0 || chon > 3);
        return chon;
    }

    // Định nghĩa hàm nhập thông tin sách
    private Sach NhapSach()
    {
        Sach cuonSach = new Sach();

        Console.WriteLine();

        Console.Write("Nhap ma so sach   : ");
        cuonSach.MaSo = Console.ReadLine();

        Console.Write("Nhap tua de       : ");
        cuonSach.NhanDe = Console.ReadLine();

        Console.Write("Nhap nam xuất bản : ");
        cuonSach.NamXB = int.Parse(Console.ReadLine());

        Console.Write("Nhap nha xuất bản : ");
        cuonSach.NhaXB = Console.ReadLine();

        Console.WriteLine("Nhap danh sach tac gia (cach nhau boi dau phay) : ");
    }
}
```

```

        string line = Console.ReadLine();
        string[] dsten = line.Split(SEPERATOR);

        for (int i = 0; i < dsten.Length; i++)
            cuonSach.ThemTacGia(dsten[i]);

        Console.WriteLine();

        return cuonSach;
    }

    // Định nghĩa hàm nhập bài báo khoa học
    private BaoKhoaHoc NhapBao()
    {
        BaoKhoaHoc baiBao = new BaoKhoaHoc();

        Console.WriteLine();

        Console.Write("Nhap ma so bai bao   : ");
        baiBao.MaSo = Console.ReadLine();

        Console.Write("Nhap tua de bai bao   : ");
        baiBao.NhanDe = Console.ReadLine();

        Console.Write("Nhap nam xuất bản   : ");
        baiBao.NamXB = int.Parse(Console.ReadLine());

        Console.Write("Nhap ten hoi nghi   : ");
        baiBao.HoiNghị = Console.ReadLine();

        Console.WriteLine("Nhap danh sách tác gia (cách nhau bởi dấu phẩy) : ");
        string line = Console.ReadLine();
        string[] dsten = line.Split(SEPERATOR);

        for (int i = 0; i < dsten.Length; i++)
            baiBao.ThemTacGia(dsten[i]);

        Console.WriteLine();

        return baiBao;
    }

    // Định nghĩa hàm nhập thông tin luận văn
    private LuanVan NhapLuanVan()
    {
        LuanVan baoCao = new LuanVan();

        Console.WriteLine();

        Console.Write("Nhap ma so luan van   : ");
        baoCao.MaSo = Console.ReadLine();
    }

```



```

        Console.WriteLine("Nhap tua de luan van : ");
        baoCao.NhanDe = Console.ReadLine();

        Console.WriteLine("Nhap nam xuất bản : ");
        baoCao.NamXB = int.Parse(Console.ReadLine());

        Console.WriteLine("Nhap ten tac gia : ");
        baoCao.TacGia = Console.ReadLine();

        Console.WriteLine();

        return baoCao;
    }

    // Định nghĩa hàm tạo danh sách bằng cách
    // nhập dữ liệu từ bàn phím
    public override DanhSachTailieu DocDuLieu()
    {
        DanhSachTailieu danhSach = new DanhSachTailieu();

        while (true)
        {
            int chucNang = ChonChucNang();
            if (chucNang == 0) break;

            switch (chucNang)
            {
                case 1: danhSach.Them(NhapSach()); break;
                case 2: danhSach.Them(NhapLuanVan()); break;
                case 3: danhSach.Them(NhapBao()); break;
            }
        }
        return danhSach;
    }
}

```

Bước 17. Tiếp theo, cài đặt lớp DuLieuNgon như sau:

```

public class DuLieuNgon
{
    // Khai báo biến để lưu bộ đọc dữ liệu
    private static BoDocDuLieu _boDoc;

    // Định nghĩa phương thức khởi tạo tĩnh
    static DuLieuNgon()
    {
        // Đọc dữ liệu từ tập tin cấu hình
        string cauHinh = System.IO.File.ReadAllText(@"Data\CauHinh.txt");

        // Tùy theo cấu hình mà tạo bộ đọc dữ liệu tương ứng
        switch (cauHinh)

```

```

    {
        case "taptin":
            _boDoc = new BoDocDuLieuTuFile(@"Data\ThuVien.txt");
            break;
        case "banphim":
            _boDoc = new BoDocDuLieuTuBanPhim();
            break;
        case "dulieugia":
            _boDoc = new BoDocDuLieuGia();
            break;
    }
}

// Định nghĩa hàm đọc dữ liệu và tạo danh sách tài liệu
public static DanhSachTaiLieu TaoDanhSach()
{
    return _boDoc.DocDuLieu();
}
}

```

Bước 18. Trong hàm Main, nhập đoạn mã sau để kiểm tra hoạt động của các lớp đã cài đặt

```

// Muốn tạo danh sách tài liệu, trong tập tin
// Data\CauHinh.txt phải lưu một trong 3 giá trị
// taptin      : Đọc dữ liệu từ file
// banphim     : Nhập dữ liệu từ bàn phím
// dulieugia   : Lấy dữ liệu nhập cố định

// Gọi hàm TaoDanhSach để tạo danh sách tài liệu
DanhSachTaiLieu thuVien = DuLieuNgon.TaoDanhSach();

// Xuất danh sách tài liệu
Console.WriteLine(thuVien);

Console.ReadKey();

```

Bước 19. Chạy chương trình và ghi nhận kết quả. Cho biết dữ liệu được lấy từ đâu?

Bước 20. Thay đổi nội dung trong tập tin Data\CauHinh.txt thành banphim

Bước 21. Chạy chương trình và ghi nhận kết quả. Cho biết dữ liệu được lấy từ đâu?

Bước 22. Thay đổi nội dung trong tập tin Data\CauHinh.txt thành dulieugia

Bước 23. Chạy chương trình và ghi nhận kết quả. Cho biết dữ liệu được lấy từ đâu?

Bước 24. Mô tả lại một cách vắn tắt toàn bộ hoạt động của chương trình.

Bước 25. Nhập thêm dữ liệu vào tập tin Data\ThuVien.txt cho đủ 20 tài liệu.

Bước 26. Bổ sung mã vào lớp BoDocDuLieuGia để tạo đủ 20 tài liệu.

Bước 27. Thực hiện lại các bước từ 19 đến 23 và ghi nhận kết quả.

D. Bài tập bắt buộc

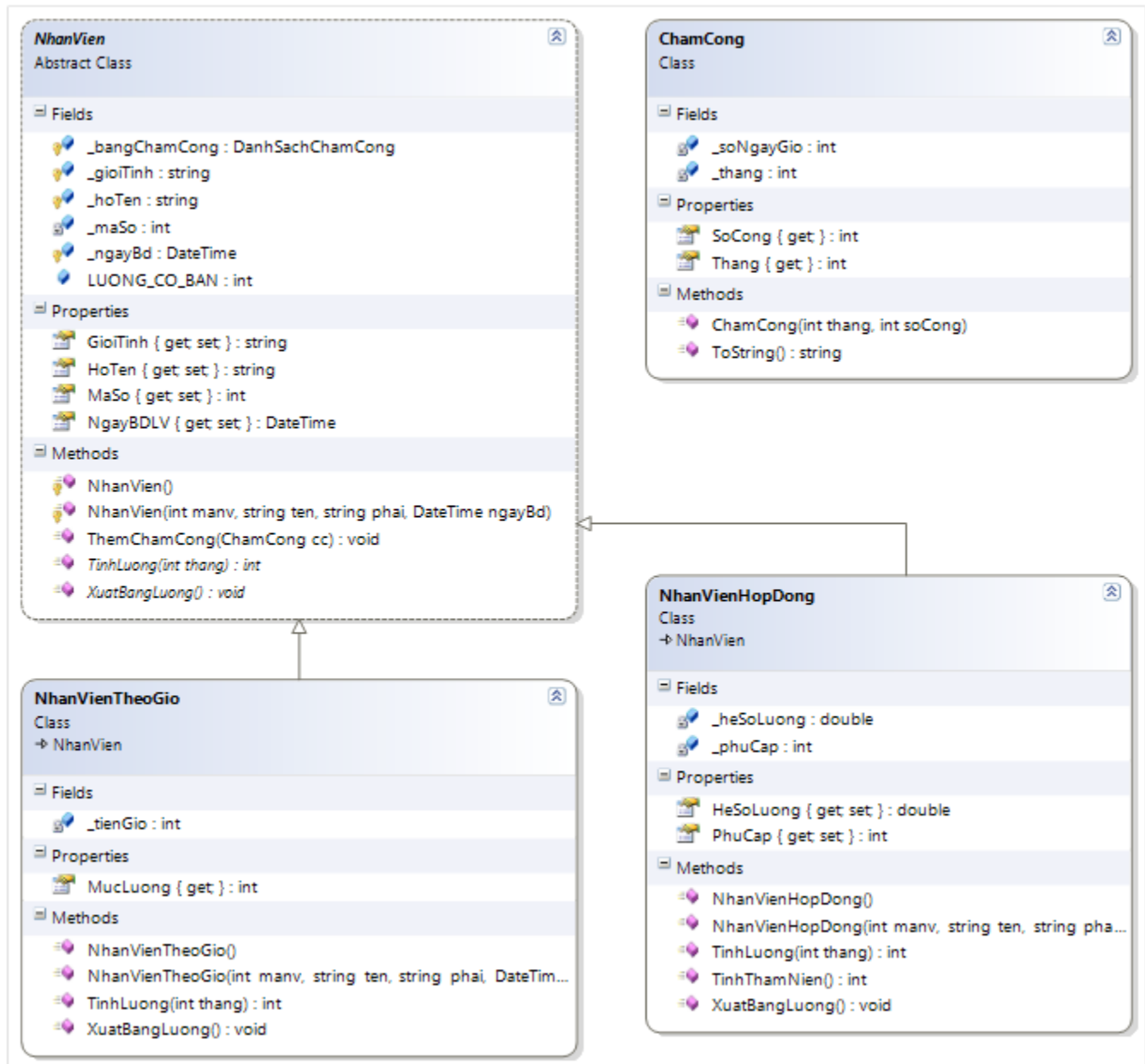
1. Tiếp tục với dự án Lab07_Bai01_QuanLySachBao ở trên. Cài đặt các phương thức sau:
 - a. Xóa một tài liệu tại vị trí (viTri) cho trước khỏi danh sách tài liệu
 - b. Xóa một tài liệu có mã số cho trước khỏi danh sách tài liệu
 - c. Xóa một tài liệu cho trước khỏi danh sách tài liệu
 - d. Tìm các tài liệu được viết bởi tác giả (tacGia) cho trước
 - e. Tìm sách được xuất bản bởi nhà xuất bản (nhaXb) cho trước
 - f. Xóa các sách được xuất bản bởi nhaXb cách đây ít nhất k năm.
 - g. Lấy danh sách tất cả các nhà xuất bản
 - h. Tìm những nhà xuất bản có nhiều sách nhất
 - i. Liệt kê danh sách các nhà xuất bản và số sách đã xuất bản
 - j. Sắp xếp sách và báo khoa học giảm dần theo năm xuất bản
 - k. Sắp danh sách tài liệu tăng dần theo nhan đề (xếp theo thứ tự abc)
 - l. Tìm những bài báo được đăng tại hội nghị (hoiNghị) và được xuất bản năm (nam)
 - m. Tìm những bài báo khoa học chỉ có đúng một tác giả
 - n. Tìm những bài báo khoa học có nhiều tác giả nhất.
 - o. Liệt kê danh sách các bài báo khoa học có sự tham gia của tác giả (tacGia)
 - p. Liệt kê danh sách các bài báo khoa học giảm dần theo năm xuất bản
 - q. Liệt kê danh sách các bài báo khoa học có sự tham gia của tác giả (tacGia), sắp theo ABC
 - r. Xóa tất cả những tài liệu có sự tham gia của tác giả (tacGia) cho trước.
 - s. Tìm những tài liệu cũ nhất (đã được xuất bản cách đây lâu nhất).
 - t. Cho biết tên của những người làm luận văn (tác giả của các luận văn).
 - u. Cho biết tên của các tác giả vừa viết sách, vừa có đăng báo khoa học và làm luận văn.
 - v. Liệt kê danh sách tất cả các tác giả của các tài liệu, nếu trùng tên thì chỉ lấy một.
 - w. Với mỗi tác giả, liệt kê số lượng tài liệu mỗi loại.
 - x. Tìm những tác giả có viết báo khoa học, làm luận văn nhưng không viết sách
 - y. Tạo lớp Menu cho phép người dùng chọn các chức năng trên.
 - z. Cài đặt hàm xử lý menu.

Sinh viên có thể bổ sung phương thức vào các lớp cơ sở hoặc tạo thêm lớp mới nếu cần.

2. Viết chương trình quản lý nhân viên theo mô tả sau:
 - a. Nhân viên được chia làm 2 loại: Nhân viên theo hợp đồng và nhân viên làm việc theo giờ. Với mỗi nhân viên, cần lưu trữ mã số, họ tên, ngày bắt đầu làm việc, giới tính.
 - b. Đối với nhân viên hợp đồng, cần lưu trữ thêm hệ số lương và phụ cấp (chẳng hạn như ăn trưa, độc hại). Hệ số lương khởi điểm là 2,34. Sau mỗi 3 năm, hệ số được tăng lên 0,33.
 - c. Chương trình phải có chức năng chấm công nhân viên. Đối với nhân viên hợp đồng, chỉ chấm công 1 tháng một lần. Với nhân viên theo giờ phải chấm công nhiều lần trong tháng (mỗi ngày) nhưng chỉ cần lưu tổng số giờ làm việc trong tháng đó.
 - d. Nhân viên hợp đồng được tính lương theo công thức: hệ số * số ngày công * lương cơ bản + phụ cấp. Lương cơ bản là 1 050 000 đồng.

- e. Nhân viên theo giờ được tính lương theo công thức: số giờ * tiền công mỗi giờ + thưởng. Trong đó, nếu số giờ công lớn hơn 150 thì thưởng tương ứng với 15 giờ làm việc. Ngược lại, thưởng tương ứng với 1/15 số giờ công.

Cho sơ đồ các lớp cần thiết như sau:



Trong đó, lớp **NhanVien** là lớp trừu tượng. Lớp này chứa hai phương thức trừu tượng là `TinhLuong`, `XuatBangLuong` và một phương thức ảo là `ThemChamCong`.

Trình tự cài đặt: **ChamCong**, **DanhSachChamCong**, **NhanVien**, **NhanVienHopDong**, **NhanVienTheoGio**, **DanhSachNhanVien**.

DanhSachNhanVien Class <div> <div>Fields</div> <div> danhSach : NhanVien[] soLuong : int </div> </div> <div> <div>Properties</div> <div> SoLuong { get; } : int this[int index] { get; } : NhanVien </div> </div> <div> <div>Methods</div> <div> DanhSachNhanVien() Them(NhanVien cc) : void TimTheoMa(int maNv) : NhanVien TimViTri(int maNv) : int Xuat() : void </div> </div>	DanhSachChamCong Class <div> <div>Fields</div> <div> danhSach : ChamCong[] soLuong : int </div> </div> <div> <div>Properties</div> <div> SoLuong { get; } : int this[int index] { get; } : ChamCong </div> </div> <div> <div>Methods</div> <div> DanhSachChamCong() Them(ChamCong cc) : void TimTheoThang(int thang) : ChamCong </div> </div>
--	---

Ý nghĩa của các thuộc tính và phương thức:

Lớp	Thuộc tính / Phương thức	Ý nghĩa
NhanVien	ThemChamCong	Chấm công cho nhân viên. Ở đây là thêm một đối tượng chấm công vào bảng chấm công
	XuatBangLuong	Xuất bảng chấm công và lương theo từng tháng
	TinhLuong	Tính lương của nhân viên theo tháng cho trước
	LUONG_CO_BAN	Là một hằng số, cho biết mức lương cơ bản
NhanVien HopDong	TinhThamNien	Tính số năm công tác của các nhân viên hợp đồng
NhanVien TheoGio	MucLuong hay _tienGio	Là số tiền công mà nhân viên nhận được, tính trên mỗi giờ công.
ChamCong	SoCong hay _soNgayGio	Là số ngày công (đối với nhân viên hợp đồng) hoặc số giờ công (đối với nhân viên theo giờ).

Thực hiện các yêu cầu sau (có thể cài đặt thêm các lớp mới nếu cần):

- Cài đặt đầy đủ các lớp cơ sở ở trên.
- Tạo tập tin chứa thông tin nhân viên, tập tin chấm công và tập tin cấu hình. Cấu trúc của các tập tin như sau:
 - CauHinh.txt: dòng 1 ghi một trong 2 giá trị: taptin hoặc banphim. Dòng thứ 2 ghi đường dẫn đến tập tin NhanVien.txt, dòng thứ 3 ghi đường dẫn đến tập tin ChamCong.txt.
 - NhanVien.txt: Mỗi dòng là thông tin về một nhân viên, thuộc một trong hai dạng sau:
HD, mã số, họ và tên, giới tính, ngày bắt đầu, hệ số lương, phụ cấp
TG, mã số, họ và tên, giới tính, ngày bắt đầu, tiền công mỗi giờ
 - ChamCong.txt: Mỗi dòng là tương ứng với một mục chấm công có dạng: mã số nhân viên, tháng, số ngày/giờ công.

- c. Tạo các lớp để đọc dữ liệu từ các tập tin hoặc từ bàn phím.
- d. Xuất danh sách toàn bộ nhân viên (Yêu cầu: Canh lề các cột).
- e. Sắp xếp danh sách nhân viên tăng dần theo họ tên.
- f. Xuất danh sách nhân viên, phân loại theo nhân viên hợp đồng và nhân viên theo giờ
- g. Xuất bảng lương của nhân viên có mã (manv) cho trước. Trong đó có các cột: tháng, số ngày/giờ công, tiền lương tháng đó.
- h. Tính tổng tiền lương của nhân viên.
- i. Xuất thông tin chi tiết về nhân viên có mã (manv), kể cả bảng lương và tổng lương.
- j. Tìm các nhân viên nữ có thâm niên công tác ít nhất là 2 năm.
- k. Tìm các nhân viên theo tháng không làm việc trong tháng 7.
- l. Tìm nhân viên (hợp đồng) có hệ số lương cao nhất tại công ty
- m. Tìm nhân viên (theo giờ) có mức thưởng cao nhất trong tháng 12.
- n. Tìm nhân viên được trả lương nhiều nhất trong tháng 12.
- o. Tìm những nhân viên đến hạn được nâng lương (có thâm niên là 3, 6, 9, ... năm).
- p. Nâng hệ số lương lên 0.33 cho tất cả các nhân viên đến hạn được nâng lương.
- q. Liệt kê danh sách nhân viên có họ (honv) cho trước
- r. Tìm tháng có số tiền lương phải trả cao nhất.
- s. Tìm nhân viên có phụ cấp cao nhất.
- t. Tính tổng tiền lương phải trả cho các nhân viên hợp đồng trong tháng cho trước.
- u. Tính tổng tiền lương phải trả cho các nhân viên làm việc theo giờ trong tháng cho trước.
- v. Tính tổng tiền lương phải trả cho tất cả nhân viên trong một năm.
- w. Tính tổng số giờ công của tất cả các nhân viên làm việc theo giờ trong tháng 12
- x. Tính tổng số ngày công của các nhân viên hợp đồng trong một năm.
- y. Xóa tất cả những nhân viên làm việc theo giờ.
- z. Xây dựng lớp Menu để chọn và xử lý các chức năng trên.

E. Bài tập làm thêm

1. Viết chương trình quản lý tài khoản tại ngân hàng BANANA theo mô tả sau:
 - Ngân hàng cần quản lý nhiều loại tài khoản khác nhau, gồm có:
 - Tài khoản vay: Là loại tài khoản có kỳ hạn mà ngân hàng cho khách hàng vay tiền. Khách có thể trả tiền nhiều lần. Tiền lãi được tính và cộng dồn vào tiền vay vào cuối mỗi tháng.
 - Tài khoản tiết kiệm: Là loại tài khoản khách hàng gửi tiền (có thể là VNĐ hoặc ngoại tệ như USD, Bảng Anh, ...) vào để hưởng lãi theo những kỳ hạn xác định. Lãi suất phụ thuộc vào kỳ hạn, số tiền gửi và được tính vào cuối kỳ (khi tắt toán). Tài khoản này không cho phép rút tiền trước thời hạn.
 - Tài khoản thanh toán: Là các tài khoản có thực hiện các giao dịch như gửi tiền vào, rút tiền ra, chuyển khoản... để đảm bảo việc thanh toán, giao dịch giữa chủ tài khoản với các cá nhân.

- Mỗi tài khoản đều có mã số, tên chủ sở hữu, ngày lập, số tiền (vay hoặc gửi), lãi suất.
- Một khách hàng có thể sở hữu nhiều tài khoản khác nhau. Đặc biệt, khách hàng có thể dùng tài khoản thanh toán chuyển khoản vào tài khoản khác hoặc tài khoản vay để trả nợ.
- Với tài khoản vay, cần phải quản lý thời hạn cuối cùng phải trả nợ cho ngân hàng. Khách hàng có thể trả nợ vào bất cứ lúc nào.
- Với tài khoản tiết kiệm, cần lưu trữ kỳ hạn (số tháng) và đơn vị tiền tệ.
- Mỗi khi xảy ra một giao dịch (lập tài khoản, nạp tiền, rút tiền, chuyển khoản, tắt toán, tính lãi suất, ...) hệ thống đều phải ghi nhận lại các giao dịch này, kể cả ngày giờ thực hiện.
- Với giao dịch lập tài khoản hay tắt toán cần lưu trữ thêm tên của nhân viên ngân hàng.
- Giả sử mọi dữ liệu được lưu trữ ở bộ nhớ trong (trên RAM).

Hãy thực hiện các yêu cầu sau:

- Vẽ sơ đồ các lớp cần thiết theo mô tả trên đây.
- Cài đặt các lớp theo sơ đồ đã tạo ở câu a.
- Mở một tài khoản mới cho khách hàng có tên (tenKh) cho trước.
- Tìm tài khoản theo mã số (maTk) cho trước.
- Tìm tất cả tài khoản do khách hàng có mã (maKh) cho trước.
- Tìm tất cả các tài khoản vay được lập cách đây ít nhất Y năm.
- Nạp một số tiền (soTien) vào tài khoản (maTk) cho trước.
- Rút một số tiền từ tài khoản tiết kiệm hoặc tài khoản ATM
- Xem số dư (hoặc dư nợ) của mọi tài khoản
- Xem số dư (hoặc dư nợ) của mọi tài khoản do khách hàng có mã (maKh) sở hữu.
- Tính số dư (hoặc dư nợ) mới cho tất cả các tài khoản.
- Chuyển một số tiền (soTien) từ tài khoản thanh toán sang tài khoản khác.
- Tắt toán (hủy tài khoản) một tài khoản.
- Liệt kê các giao dịch chuyển khoản trong ngày
- Tìm những tài khoản không thực hiện bất kỳ giao dịch nào trong ngày.
- Hiển thị thông tin chi tiết tất cả các giao dịch.
- Cho biết thông tin những tài khoản tiết kiệm đã được tắt toán trong ngày.
- Tìm tài khoản vay với số tiền lớn nhất.
- Tìm tài khoản thực hiện nhiều giao dịch nhất.
- Tính tổng số tiền được chuyển khoản giữa các tài khoản.
- Tìm tài khoản tiết kiệm sử dụng đơn vị tiền tệ USD.
- Tính lãi suất định kỳ (vào cuối tháng) cho các tài khoản vay và tài khoản thanh toán.
- Tính tổng số tiền mà ngân hàng cho vay.
- Tính chênh lệch giữa số tiền mà ngân hàng cho vay và tổng số tiền khách hàng gửi vào.
- Cho biết loại giao dịch nào được thực hiện nhiều nhất
- Liệt kê tên các nhân viên lập tài khoản cho khách hàng có mã (maKh).