

MỤC LỤC

Bài 1. Làm quen môi trường Visual Studio và ngôn ngữ C#	1
Bài 2. Lớp và đối tượng	15
Bài 3. Toán tử	26
Bài 4. Danh sách đối tượng.....	33
Bài 5. Ôn tập	42
Bài 6. Kế thừa	47
Bài 7. Đa hình	59
Bài 8. Interface.....	68
Bài 9. Bài tập tổng hợp	77
Bài 10. Ôn tập	80



Trường: **ĐH CNTP TP.HCM**
Khoa: **Công nghệ Thông tin**
Bộ môn: **Công nghệ phần mềm**
MH: **TH Lập trình hướng đối tượng**

BÀI 1. LÀM QUEN MÔI TRƯỜNG VISUAL STUDIO VÀ NGÔN NGỮ C#



A. MỤC TIÊU:

- Làm quen với IDE Visual Studio và ngôn ngữ lập trình C#.
- Thực hiện các thao tác cơ bản Visual Studio và viết chương trình bằng ngôn ngữ C#: tạo mới Solution, Project, nhập xuất dữ liệu, viết các biểu thức, câu lệnh cơ bản.
- Thực hiện một số bài tập cơ bản.

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

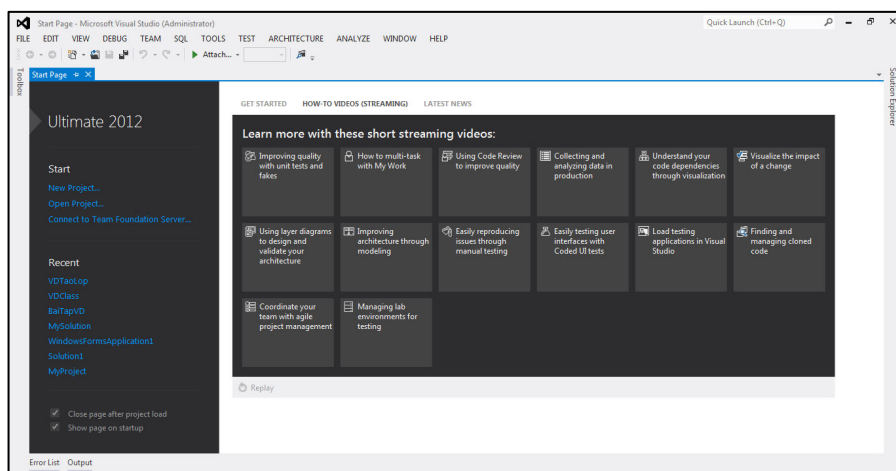
STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

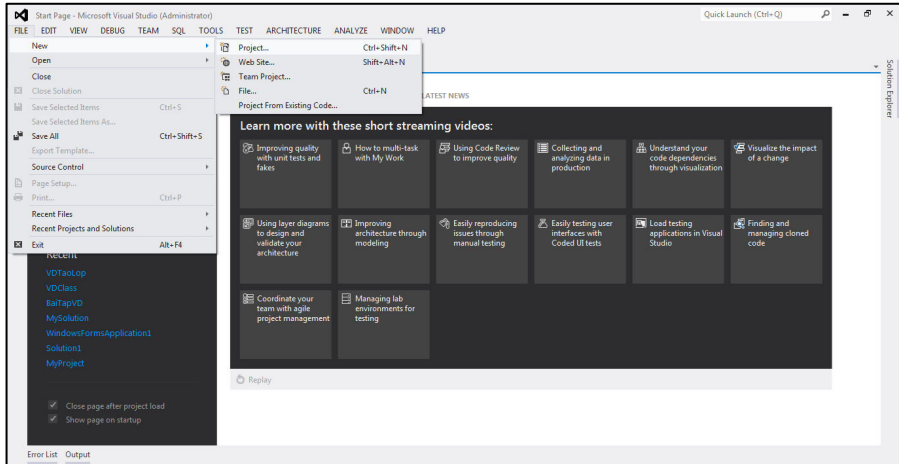
1. Tóm tắt lý thuyết

a. Tạo mới Solution và Project chạy bằng ngôn ngữ C#, biên dịch dạng Console trên Visual Studio.

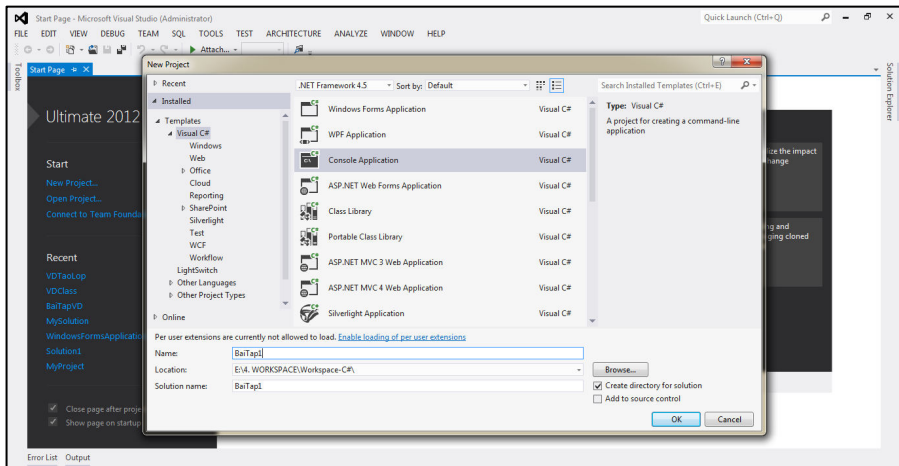
- Bước 1: Mở chương trình Visual Studio



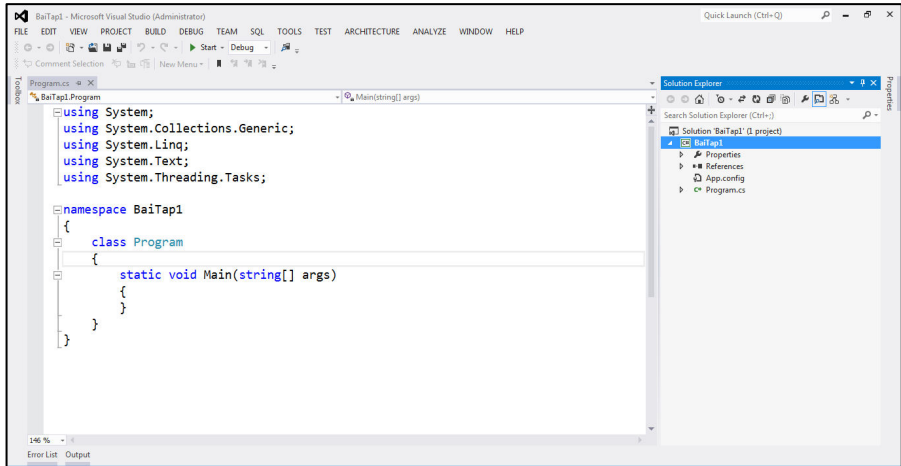
– Bước 2: Tạo mới Solution và Project



– Bước 3: Chọn loại và đặt tên cho project



– Bước 4: Viết chương trình trên project đã tạo



b. Các lệnh khai báo thư viện và nhập xuất dữ liệu trong C#

– Khai báo thư viện:

`using <Tên thư viện>;`

– Lệnh xuất dữ liệu:

`Console.WriteLine(<dữ liệu cần xuất và các định dạng>;`

– Lệnh nhập dữ liệu dạng chuỗi:

`string Console.ReadLine()`

– Lệnh nhập dữ liệu dạng khác chuỗi: cần chuyển dữ liệu nhập vào về đúng kiểu bằng lệnh Parse.

Ví dụ: Nhập giá trị cho biến int a:

`int a = int.Parse(Console.ReadLine());`

c. Ví dụ các định dạng dữ liệu xuất ra của lệnh Console.WriteLine()

```
using System;
class Sample
{
    enum Color {Yellow = 1, Blue, Green};
    static DateTime thisDate = DateTime.Now;
    public static void Main()
    {
        Console.Clear();
    }
}
```

```
// Format a negative integer or floating-point number in
various ways.

    Console.WriteLine("Standard Numeric Format
Specifiers");

    Console.WriteLine(
        "(C) Currency: . . . . . {0:C}\n" +
        "(D) Decimal:.. . . . . {0:D}\n" +
        "(E) Scientific: . . . . . {1:E}\n" +
        "(F) Fixed point:.. . . . . {1:F}\n" +
        "(G) General:.. . . . . {0:G}\n" +
        "    (default):.. . . . . {0} (default =
'G')\n" +
        "(N) Number: . . . . . {0:N}\n" +
        "(P) Percent:.. . . . . {1:P}\n" +
        "(R) Round-trip: . . . . . {1:R}\n" +
        "(X) Hexadecimal:.. . . . . {0:X}\n",
        -123, -123.45f);

// Format the current date in various ways.

    Console.WriteLine("Standard DateTime Format
Specifiers");

    Console.WriteLine(
        "(d) Short date: . . . . . {0:d}\n" +
        "(D) Long date:.. . . . . {0:D}\n" +
        "(t) Short time: . . . . . {0:t}\n" +
        "(T) Long time:.. . . . . {0:T}\n" +
        "(f) Full date/short time: . . {0:f}\n" +
        "(F) Full date/long time:.. . {0:F}\n" +
        "(g) General date/short time:.. {0:g}\n" +
        "(G) General date/long time: . {0:G}\n" +
        "    (default):.. . . . . {0} (default =
'G')\n" +
        "(M) Month:.. . . . . {0:M}\n" +
        "(R) RFC1123:.. . . . . {0:R}\n" +
        "(s) Sortable: . . . . . {0:s}\n" +
```

```

        "(u) Universal sortable: . . . {0:u}
(invariant)\n" +
        "(U) Universal full date/time: {0:U}\n" +
        "(Y) Year: . . . . . {0:Y}\n",
        thisDate);
// Format a Color enumeration value in various ways.
    Console.WriteLine("Standard Enumeration Format
Specifiers");
    Console.WriteLine(
        "(G) General:. . . . . {0:G}\n" +
        "    (default):. . . . . {0} (default =
'G')\n" +
        "(F) Flags:. . . . . {0:F} (flags or
integer)\n" +
        "(D) Decimal number: . . . . . {0:D}\n" +
        "(X) Hexadecimal:. . . . . {0:X}\n",
        Color.Green);
    }
}
/*

```

This code example produces the following results:

Standard Numeric Format Specifiers

```

(C) Currency: . . . . . ($123.00)
(D) Decimal:. . . . . -123
(E) Scientific: . . . . . -1.234500E+002
(F) Fixed point:. . . . . -123.45
(G) General:. . . . . -123
    (default):. . . . . -123 (default = 'G')
(N) Number: . . . . . -123.00
(P) Percent:. . . . . -12,345.00 %
(R) Round-trip: . . . . . -123.45
(X) Hexadecimal:. . . . . FFFFFFF85

```

Standard DateTime Format Specifiers

```

(d) Short date: . . . . . 6/26/2004
(D) Long date: . . . . . Saturday, June 26, 2004
(t) Short time: . . . . . 8:11 PM
(T) Long time: . . . . . 8:11:04 PM
(f) Full date/short time: . . Saturday, June 26, 2004
8:11 PM
(F) Full date/long time: . . . Saturday, June 26, 2004
8:11:04 PM
(g) General date/short time: . 6/26/2004 8:11 PM
(G) General date/long time: . 6/26/2004 8:11:04 PM
    (default): . . . . . 6/26/2004 8:11:04 PM
(default = 'G')
(M) Month: . . . . . June 26
(R) RFC1123: . . . . . Sat, 26 Jun 2004 20:11:04
GMT
(s) Sortable: . . . . . 2004-06-26T20:11:04
(u) Universal sortable: . . . 2004-06-26 20:11:04Z
(invariant)
(U) Universal full date/time: Sunday, June 27, 2004
3:11:04 AM
(Y) Year: . . . . . June, 2004
Standard Enumeration Format Specifiers
(G) General: . . . . . Green
    (default): . . . . . Green (default = 'G')
(F) Flags: . . . . . Green (flags or integer)
(D) Decimal number: . . . . . 3(X) Hexadecimal: . . . .
. . 00000003
*/

```

d. Các lệnh điều khiển cơ bản của C#

– Lệnh if/if-else

```
if(<biểu thức logic>)  
{  
    //khối lệnh;  
}
```

```
if( <biểu thức logic> )  
{  
    //khối lệnh 1;  
}  
else  
{  
    //khối lệnh 2;  
}
```

– Lệnh switch-case:

```
switch( <biểu_thức_lựa_chọn> ) {  
    case <biểu_thức_hằng 1>:  
    {  
        //khối lệnh;  
        break;  
    }  
    case <biểu_thức_hằng 2>:  
    {  
        //khối lệnh;  
        break;  
    }  
    ...  
    case <biểu_thức_hằng n>:  
    {  
        //khối lệnh;  
        break;  
    }  
    [default : khối lệnh; break;]  
}
```


– Lệnh for/while/do-while/ foreach

```
for(<biểu thức 1>; <biểu_thức_logic>; <biểu thức 2>)  
{  
    //khối lệnh;  
}
```

```
while( <biểu_thức_logic> )  
{  
    //khối_lệnh;  
}
```

```
do{  
    //khối_lệnh;  
}while( <biểu_thức_logic> );
```

```
foreach(<kiểu dữ liệu> tên_biến in biến_mảng)  
{  
    //khối lệnh;  
}
```

e. Mảng & danh sách liên kết trong C#

– **Mảng 1 chiều: cú pháp**

- + <kiểu dữ liệu>[] <tên mảng> = new <kiểu dữ liệu>[<kích thước mảng>];
- + <kiểu dữ liệu>[] <tên mảng> = new <kiểu dữ liệu>[<kích thước mảng>] {danh sách phần tử mảng};

Ví dụ: int[] a = new int[4] {3, 6, 8, 1};

– **Mảng 2 chiều: cú pháp**

- + <kiểu dữ liệu>[,] <tên mảng> = new <kiểu dữ liệu>[<số dòng,số cột>];
- + <kiểu dữ liệu>[,] <tên mảng> = new <kiểu dữ liệu>[<số dòng,số cột>] {danh sách phần tử mảng};

Ví dụ: int[,] a = new int[2, 3] { {1,2,3},{4,5,6} };

– **Mảng động ArrayList**

- + Là mảng động, mỗi phần tử là một object → các phần tử trong ArrayList có thể khác kiểu dữ liệu.
- + Thuộc Namespace: System.Collections
- + Khai báo:
`ArrayList <tên mảng> = new ArrayList();`
`ArrayList <tenmang> = new ArrayList (Số phần tử);`

Ví dụ:

```
ArrayList a = new ArrayList();  
    a.Add("Main");  
    a.Add('a');  
    a.Add(5);  
    a.Add(DateTime.Now);  
for (int i = 0; i < a.Count; i++)  
    Console.WriteLine("a[i]: {0}", a[i]);
```

– **Danh sách liên kết – List**

- + Là danh sách liên kết, các phần tử có cùng kiểu dữ liệu
- + Thuộc **Namespace**: System.Collections
- + Khai báo:
`List<Kiểu dữ liệu> <tên mảng> = new List<kiểu dữ liệu>();`
`List<Kiểu dữ liệu> <tên mảng> = new List<kiểu dữ liệu>(Số phần tử);`

Ví dụ:

```
List<int> t = new List<int>();  
t.Add(5); //thêm 1 phần tử  
t.AddRange(new int[] {3,9,8,2}); /* thêm nhiều phần tử là mảng 1 chiều các số int */  
foreach (int a in t)  
    Console.Write(a); //a duyệt qua các phần tử của t
```

2. Bài tập thực hành trên lớp

Lưu ý: Tổ chức các bài tập chung trong một project, có hiển thị menu cho người dùng lựa chọn.

Bài tập có hướng dẫn

Bài 1. Viết chương trình nhập vào 2 số nguyên. Xuất ra tổng và hiệu, tích, thương của 2 số đó.

Hướng dẫn: file Program.cs

```
namespace BaiTap1
{
    class Program
    {
        static void Main(string[] args)
        {
            int a, b;
            Console.WriteLine(" Nhập gia tri so thu 1: ");
            /*Nhập giá trị cho biến số nguyên a nên dùng lệnh
            int.Parse(...) để chuyển chuỗi nhập vào về dạng số int*/
            a = int.Parse(Console.ReadLine());
            Console.WriteLine(" Nhập gia tri so thu 2: ");
            b = int.Parse(Console.ReadLine());
            Console.WriteLine("Tong 2 so la: {0}", a + b);
            /*{0}: định dạng qui định vị trí và thứ tự của giá
            trị biến được xuất ra*/
            Console.WriteLine("Hiệu 2 so la: {0}", a - b);
            //Chúng ta có thể xuất ra nhiều biến trong 1 lệnh
            Console.WriteLine("Tich 2 so la: {0} \n Thuong cua
            2 so la: {1}", a * b, a/b );
            /*{0}: định dạng qui định vị trí và thứ tự của giá
            trịa*b, {1}: tương tự dành cho a/b */
            Console.ReadLine(); //Dừng màn hình
        }
    }
}
```

Bài 2. Viết chương trình nhập vào thông tin của một sinh viên gồm:

- + Mã số sinh viên (string)
- + Họ tên (string)
- + Điểm trung bình - ĐTB (float)

Sau đó tính kết quả xếp loại của sinh viên, biết rằng:

- + ĐTB \geq 8.0: loại giỏi
- + $8 > \text{ĐTB} \geq 6.5$: loại khá
- + $6.5 > \text{ĐTB} \geq 5.0$: loại trung bình
- + ĐTB < 5 : loại yếu kém.

Xuất đầy đủ thông tin của sinh viên.

Hướng dẫn:

```
namespace BaiTap2
{
    class Program
    {
        static void Main(string[] args)
        {
            string mssv, hten;
            float dtb;
            string xeploai;
            Console.WriteLine("Nhập thông tin của sinh viên: ");
            Console.WriteLine("Mã số sv: ");
            mssv = Console.ReadLine(); /* mssv kiểu chuỗi nên không
cần gọi lệnh Parse */
            Console.WriteLine("Họ tên sv: ");
            hten = Console.ReadLine(); /* mssv kiểu chuỗi nên không
cần gọi lệnh Parse */
            Console.WriteLine("Điểm trung bình: ");
            /* dtb có kiểu số nên cần dùng lệnh Parse để chuyển
string về đúng kiểu */
            dtb = float.Parse(Console.ReadLine());
            if (dtb >= 8)
```

```

        xeploai = "gioi";
    else if(dtb>=6.5)
        xeploai = "kha";
    else if (dtb >= 5)
        xeploai = "trung binh"
    else
        xeploai = "yeu kem";
    Console.WriteLine("Thông tin sv: \n MSSV:{0} - Ho
ten:{1} - ĐTB: {2:0.00} - Xep loai: {3}", mssv, hten, dtb,
xeploai);
    Console.ReadLine();
    }
    }
}

```

Bài 3: Tạo dãy các số chứa n số nguyên bằng cách sinh ra số ngẫu nhiên, với n do người dùng nhập vào. Sau đó xuất dãy số ra màn hình.

Hướng dẫn:

```

namespace BaiTap3
{
    class Program
    {
        static void Main(string[] args)
        {
            int n;
            Console.Write("Nhập số phần tử dãy số: ");
            n = int.Parse(Console.ReadLine());
            List<int> dso = new List<int>(n);
            Random a = new Random(); // sinh số nguyên ngẫu nhiên
            /* Sinh số ngẫu nhiên và thêm vào dãy số, dùng lệnh
for để duyệt List */
            for (int i = 0; i < dso.Capacity; i++)
            {
                int k = a.Next(100); //sinh số nguyên dương ∈[0;100)
            }
        }
    }
}

```

```

        dso.Add(k); //them số vừa sinh vào dãy số
    }
    //In dãy số, dùng foreach để duyệt List
    foreach (int t in dso)
        Console.Write(t + " ");
    Console.ReadLine();
}
}
}

```

Bài tập sinh viên tự làm

Bài 4. Viết chương trình biện luận và giải phương trình bậc 1.

Bài 5. Viết chương trình biện luận và giải phương trình bậc 2.

Bài 6. Viết chương trình nhập vào giá ngày/tháng/năm của một ngày trong một năm bất kỳ. Cho biết ngày đó thứ mấy. Biết rằng công thức tính thứ của một ngày/tháng/năm như sau:

Điều kiện:

Tháng < 3: tháng = tháng + 12; năm = năm - 1

Tháng >= 3

$$n = (\text{ngày} + 2 * \text{tháng} + (3 * (\text{tháng} + 1))) / 5 + \text{năm} + (\text{năm} / 4) \% 7$$

với n là kết quả thứ theo thứ tự: 0 là chủ nhật, 1 là thứ 2, ..., 6 là thứ 7.

Bài 7. Viết chương trình nhập vào 1 số nguyên n.

- Phân tích n ra thừa số nguyên tố.
- Cho biết n có bao nhiêu chữ số.

3. Bài tập về nhà

Bài 8. Viết chương trình cho nhập vào số phải là số chính phương, xuất số vừa nhập ra màn hình.


Bài 9. Nhập vào 1 ngày tháng năm. Cho biết ngày trước đó và ngày hôm sau là ngày mấy.

Bài 10. Tạo dãy các số nguyên ngẫu nhiên có 20 số (dùng ArrayList hoặc List). Mỗi phần tử là số nguyên dương và nhỏ hơn 1000. Thực hiện

các yêu cầu sau trên dãy số:

- Xuất dãy số.
- Đảo dãy số
- Tìm trong dãy số có chứa số x không? Với x nhập từ bàn phím
- Xuất các phân tử có 2 chữ số.
- Xuất các số có các chữ số đều là số chẵn.
- Xuất các số nguyên tố trong dãy số (nếu có).
- Xóa khỏi dãy số các số lẻ và là bội của 3.
- Tăng giá trị lên 1 đơn vị cho các số mà nhỏ hơn 2 số liền kề (trước/và sau) nó.
- Sắp xếp dãy số tăng dần, giảm dần.

--HẾT--

Trường: ĐH CNTP TP.HCM Khoa: Công nghệ Thông tin Bộ môn: Công nghệ phần mềm MH: TH Lập trình hướng đối tượng	BÀI 2. LỚP VÀ ĐỐI TƯỢNG	
---	--------------------------------	--

A. MỤC TIÊU:

- Cài đặt được lớp có các thuộc tính, phương thức khởi tạo và các hàm xử lý cơ bản, **phương thức static**.
- **Gọi được các đối tượng** trong phương thức chương trình.

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

1. Tóm tắt lý thuyết

1.1. Cú pháp tạo lớp

```
public class <Tên_lớp>
{
    //thành phần dữ liệu
    private <kiểu dữ liệu> <tên thuộc tính>;
    ...
    //thành phần phương thức
    public <kiểu trả về> <tên phương thức>([ds tham số])
    {
        //phần thân phương thức
    }
    ...
}
```

1.2. Khai báo đối tượng

```
<Tên lớp> <tên đối tượng> = new <Tên lớp>();
<Tên lớp> <tên đối tượng> = new <Tên lớp>(<ds tham số>);
```


1.3. Truy cập phương thức của lớp

<Tên đối tượng>.<Tên phương thức>([ds tham số]);

1.4. Property

```
public <Kiểu_trả_về> <Tên_Property>
{
    get
    {
        //các câu lệnh ...
        return <biểu thức trả về>;
    }
    set
    {
        //các câu lệnh ... xử lý value
        <tên thuộc tính> = value;
    }
}
```

1.5. Phương thức khởi tạo

a. Khởi tạo không tham số

```
public <Tên_lớp()>
{
}
}
```

☐ Tham số

b. Khởi tạo có tham số

```
public <Tên_lớp>(Danh sách tham số)>
{
}
}
```

< TS

c. Khởi tạo sao chép

```
public <Tên_lớp>(Tên_lớp <đối tượng>)>
{
}
}
```

SC

1.6. Phương thức hủy

```
~<Tên_lớp>()  
{  
  
}
```

🔗 **Lưu ý: Thêm mới một lớp trong project:** Click phải Project → Add → Class → đặt tên lớp mới → OK

2. Bài tập thực hành trên lớp

Bài tập có hướng dẫn

Bài 1. Xây dựng lớp mô tả hình tròn (HìnhTron) chứa thuộc tính bán kính (số thực). Viết các phương thức:

- Property (get/set) với ràng buộc bán kính > 0
- Khởi tạo (không tham số và có tham số)
- Nhập/xuất thông tin cho hình tròn
- Tính chu vi, diện tích

Hướng dẫn:

```
public class HìnhTron  
{  
    private double r;  
    public double R //Property với ràng buộc bán kính >0  
    {  
        get { return r; }  
        set  
        {  
            if (value < 0)  
            {  
                Console.WriteLine("Du lieu bi loi");  
                r = 0;  
            }  
            else  
                r = value;  
        }  
    }  
}
```

```

    }
}
public HìnhTron() //Khởi tạo không tham số
{
    this.r = 0;
}
public HìnhTron(double r) //Khởi tạo có tham số
{
    this.r = r;
}
public void nhap() //Phương thức nhập bán kính
{
    Console.Write("Nhap ban kinh hinh tron: ");
    this.r = double.Parse(Console.ReadLine());
}
public double tinhChuvi() //Phương thức tính chu vi
{
    return this.r * 2 * Math.PI;
}
public double tinhDienTich() //Phương thức tính diện tích
{
    return Math.Pow(this.r, 2) * Math.PI;
}
public void xuat() //Phương thức xuất
{
    Console.WriteLine("Hinh tron co ban kinh: {0:0.00},
                        chu vi: {1:0.00}, dien tich: {2:0.00}",r,
                        tinhChuvi(), tinhDienTich());
}
}

```

– Xây dựng hàm Main trong file program.cs

```

class Program
{

```

```

static void Main(string[] args)
{
    /* khai báo và khởi tạo đối tượng a thuộc lớp HìnhTron
    với phương thức khởi tạo không tham số */
    HìnhTron a = new HìnhTron();
    a.nhap(); //Gọi phương thức nhap()
    a.xuat(); //Gọi phương thức xuất()
    /* khai báo và khởi tạo đối tượng b thuộc lớp HìnhTron
    với phương thức khởi tạo có tham số */
    HìnhTron b = new HìnhTron(5.0f);
    Console.WriteLine("Chu vi hình tron b: {0:0.00}",
        b.tinhChuvi());
    Console.WriteLine("Dien tich hình tron b: {0:0.00}",
        b.tinhDienTich());
    Console.ReadLine(); //Dừng màn hình
}
}

```

Bài 2. Xây dựng lớp NhanVien chứa các thông tin:

- Mã số, họ tên (kiểu chuỗi)
- Số ngày công (số nguyên) (giá trị > 0) *get Set*
- **Xếp loại (char (A,B,C))**. Kết quả xếp loại thi đua dựa vào qui định:
 - + Số ngày công > 26 : loại A
 - + $26 \geq$ Số ngày công ≥ 22 : loại B
 - + Số ngày công < 22 : loại C
- **Lương ngày (200.000 đồng)**: Áp dụng cho **tất cả** các nhân viên

*Nhan vien
String, int*

Hãy xây dựng thêm các phương thức sau:

- Property cho từng thuộc tính có kiểm tra ràng buộc theo yêu cầu nêu trên
- 3 hàm khởi tạo
- Hàm hủy
- Nhập/xuất thông tin nhân viên
- Hàm tính lương (số ngày công * lương ngày)

- Hàm tính thưởng: nếu xếp loại A thì thưởng 5% lương, loại B thưởng 2% lương và loại C không thưởng.

Hướng dẫn:

```
class NhanVien
{
    string ms, ht; // mã số, họ tên
    public string Ms
    {
        get { return ms; }
        set { ms = value; }
    }
    public string Ht
    {
        get { return ht; }
        set { ht = value; }
    }
    int nc; // số ngày công
    public int Nc
    {
        get { return nc; }
        set
        {
            if (value < 0)
            {
                Console.WriteLine("Du lieu sai");
                nc = 0;
            }
            else nc = value;
        }
    }
    public char Xl
    {
```

```
/* Thuộc tính xếp loại không có hàm set, bởi vì giá trị xếp loại tính dựa theo ngày công không được nhập vào */
```

```
get
{
    if (nc >= 26)
        return 'A';
    else if (nc >= 22)
        return 'B';
    else
        return 'C';
}
```

```
/* Thuộc tính lương ngày áp dụng chung cho tất cả nhân viên nên phải là thành phần static */
```

```
public static int LuongNgay = 200000;
```

```
//-----Hàm khởi tạo-----
```

```
/* khi khởi tạo giá trị cho nhân viên chỉ khởi tạo cho 3 thuộc tính: ms, ht và nc */
```

```
public NhanVien()
```

```
{
```

```
    Ms = Ht = "";
```

```
    Nc = 0;
```

```
}
```

```
public NhanVien(string ht, string ms, int nc)
```

```
{
```

```
    this.Ht = ht;
```

```
    this.Ms = ms;
```

```
    this.Nc = nc;
```

```
}
```

```
public NhanVien(NhanVien a) //Khởi tạo sao chép
```

```
{
```

```
    this.Ht = a.Ht;
```

```
    this.Ms = a.Ms;
```

Khởi tạo 0
TSO
string
int

Khởi tạo
có TS

↑

```

        this.Nc = a.Nc;
    }
    //-----Hàm hủy-----
    ~NhanVien()
    {
    }

    //-----Các hàm xử lý khác-----
    public void nhapTTNV()
    {
        Console.Write("Nhap ma so: ");
        Ms = Console.ReadLine();
        Console.Write("Nhap ho ten: ");
        Ht = Console.ReadLine();
        Console.Write("Nhap so ngay cong: ");
        Nc = int.Parse(Console.ReadLine());
    }
    public void xuatTTNV()
    {
        Console.WriteLine("{0}, {1}, {2}, {3}, {4}",
                           ms, ht, nc, xl, tinhLuong());
    }
    public int tinhLuong()
    {
        return nc*LuongNgay;
    }
    public float tinhThuong()
    {
        if (xl == 'A')
            return tinhLuong()*5/100;
        else
            if (xl == 'B')
                return tinhLuong()*2/100;
    }

```

```
        else return 0;
    }
}
```

Sinh viên xây dựng hàm Main để kiểm tra các hàm đã viết trong lớp NhanVien

Bài tập tự làm trên lớp

Bài 3. Xây dựng Lớp Nước giải khát gồm các thông tin:

- Tên hàng (string)
- Đơn vị tính (string)
- Số lượng (int) (>0)
- Đơn giá (float) (>0)
- Thuế VAT (float): Áp dụng chung cho các loại nước giải khát và giá trị có thể đổi theo thời gian, hiện tại là 10%.

Yêu cầu:

- Khai báo các property cần thiết

🔗 **Lưu ý:** đơn vị tính chỉ nhận 1 trong 4 giá trị: “kết”, “thùng”, “chai”, “lon”. Nếu đơn vị tính không thuộc 1 trong 4 tham số trên thì gán đơn vị tính là “kết”.

- Hàm khởi tạo
- Các hàm xử lý khác
- + Xây dựng phương thức tính thành tiền dựa vào đơn vị tính (DVT) như sau:
 - DVT = “kết” hoặc “thùng” → Thành tiền = số lượng * đơn giá + số lượng * đơn giá * thuế VAT.
 - DVT = “chai” → Thành tiền = số lượng * (đơn giá/20) + số lượng * (đơn giá / 20)*thuế VAT.
 - DVT = “lon” → Thành tiền = số lượng * (đơn giá/24) + số lượng * (đơn giá / 24)*thuế VAT.
- + Phương thức nhập/xuất thông tin mặt hàng giải khát.

- Viết hàm Main để kiểm tra các thành phần của lớp nước giải khát.

3. Bài tập về nhà

Bài 4. Xây dựng lớp hình chữ nhật (HCN) có các thành phần sau:

- Các thuộc tính mô tả chiều dài, chiều rộng
- 3 phương thức khởi tạo: không có và có tham số
- Phương thức tính chu vi
- Phương thức tính diện tích
- Phương thức tính đường chéo
- Phương thức nhập/xuất thông tin hình chữ nhật
- Phương thức thay đổi kích thước hình chữ nhật (viết theo dạng method overload):

```
void changeSize (int tx, int ty, int kiểu);
```

//kích thước (chiều dài, chiều rộng) HCN sẽ tăng lên thêm tx và ty nếu kiểu = 1, ngược lại chúng sẽ giảm tx, ty nếu kiểu=0

```
void changeSize(HCN a, int kiểu);
```

// kích thước HCN cộng thêm kích thước HCN a nếu kiểu=1, ngược lại chúng sẽ giảm nếu kiểu=0

- Viết chương trình nhập vào kích thước 1 HCN, xuất ra chu vi, diện tích và đường chéo.
- Nhập vào các kích thước để thay đổi HCN

Bài 5. Xây dựng lớp Time chứa các thành phần sau:


- Các thuộc tính giờ, phút, giây (cần kiểm tra giá trị nhập vào của giờ, phút và giây: $0 \leq \text{giờ} \leq 23$; $0 \leq \text{phút} \leq 59$; $0 \leq \text{giây} \leq 59$).
- 3 phương thức khởi tạo.
- Nhập/xuất thời gian theo dạng 24 giờ.
- Nhập/xuất thời gian theo dạng 12 giờ (cần có thêm thông tin AM và PM).
- Phương thức kiểm tra giờ có hợp lệ không? Biết rằng giờ hợp lệ là giờ theo hệ thống đồng hồ.

- Viết phương thức tăng/giảm giờ (Phương thức overload):
 - + `void tanggio (int sogiay);` → tăng giờ thông thường theo dạng giờ 24h
 - + `void tanggio (int sogiay, string kieuGio);` → kết quả ra dạng giờ 24 tiếng nếu kiểu giờ là 24, kết quả dạng 12h kèm AM/PM nếu kiểu giờ là 12.
 - + Tương tự cho 2 phương thức giảm giờ.

Bài 6. Xây dựng lớp chứa thông tin của các vận động viên chạy đua trong 1 cuộc đua marathon, gồm những thuộc tính và thành phần sau:

- Mã số, họ tên, số áo, thời gian bắt đầu (kiểu Giờ của bài 5), thời gian kết thúc (kiểu Giờ của bài 5).
- Ngoài ra còn có thành tích chuẩn (1:30:00): thời gian qui định thành tích phải \leq thành tích chuẩn này, nếu chạy chậm hơn thì coi như không đạt trong cuộc thi. Giá trị này áp dụng cho tất cả các vận động viên.
- 3 Phương thức khởi tạo.
- Phương thức nhập/xuất thông tin vận động viên.
- Phương thức tính thành tích (= thời gian kết thúc – thời gian bắt đầu), đổi ra dạng giờ:phút:giây
- Phương thức kiểm tra thời gian nhập vào bắt đầu và kết thúc có hợp lệ không? Giờ hợp lệ là giờ thuộc hệ thống giờ trên đồng hồ (loại 24h).
- Viết chương trình nhập/xuất thông tin vận động viên. Lưu ý: không nhập thành tích vì thuộc tính này lấy kết quả từ thời gian bắt đầu và kết thúc.

--HẾT--

Trường: ĐH CNTP TP.HCM Khoa: Công nghệ Thông tin Bộ môn: Công nghệ phần mềm MH: TH Lập trình hướng đối tượng	BÀI 3. TOÁN TỬ	
---	-----------------------	--

A. MỤC TIÊU:

- Cài đặt được lớp có các thuộc tính, hàm khởi tạo và các hàm xử lý cơ bản, thành phần static, hàm chồng, operator.
- Gọi được các đối tượng trong hàm chương trình.

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

1. Tóm tắt lý thuyết

a. Cú pháp nạp chồng toán tử

```
public static <kiểu trả về> operator <toán tử>(các tham số)
{
    //thân toán tử
}
```

b. Cú pháp nạp chồng toán tử ép kiểu

```
public static [implicit|explicit] operator <Kiểu Trả Về>(các tham số)
{
    //thân toán tử
}
```

- **Implicit:** Chuyển kiểu ngầm định, chuyển từ kiểu dữ liệu nhỏ sang kiểu dữ liệu lớn → Không bị mất dữ liệu.

Ví dụ: int → float

- **Explicit:** Chuyển kiểu tường minh (chỉ định), chuyển từ kiểu dữ liệu lớn sang kiểu dữ liệu nhỏ → Có thể bị mất dữ liệu.

Ví dụ: float → int

2. Bài tập thực hành trên lớp

Bài tập có hướng dẫn

Bài 1. Xây dựng lớp SoPhuc mô tả số phức trong toán học ($a+bi$), chứa các thông tin phần thực (a – số thực) và phần ảo (b - số thực), cùng các phương thức theo yêu cầu:

- Hàm khởi tạo
- Hàm nhập/xuất
- Hàm cộng 2 số phức (dạng hàm bình thường, không phải nạp chồng toán tử).
- Hàm so sánh == 2 số phức (dạng hàm bình thường, không phải nạp chồng toán tử).
- Viết lại 2 hàm cộng và so sánh == dạng operator.
- **SV bổ sung thêm các operator: -, *, và !=. Biết rằng công thức tính các phép toán trong số phức như sau:**

Cho hai số phức $z = a + bi$ và $z' = a' + b'i$

- $z = z'$ nếu $a = a'$ và $b = b'$
- $w = z + z' = (a + a') + (b + b')i$
- $u = z - z' = (a - a') + (b - b')i$
- $t = z.z' = a.a' - b.b' + (ab' + a'b)i$

Hướng dẫn

Khai báo class

```
class SoPhuc
{
    float a; //phần thực
    public float A
    {
        get { return a; }
        set { a = value; }
    }
    float b; // phần ảo
    public float B
```

```

{
    get { return b; }
    set { b = value; }
}
public SoPhuc()
{
    a = b = 1;
}
public SoPhuc(float a, float b)
{
    this.a = a; this.b = b;
}
public SoPhuc(SoPhuc z)
{
    this.a = z.A;
    this.b = z.B;
}
public void nhapSP()
{
    Console.Write("Nhap phan thuc: ");
    a = float.Parse(Console.ReadLine());
    Console.Write("Nhap phan ao: ");
    b = float.Parse(Console.ReadLine());
}
public void xuatSP()
{
    Console.Write("z= {0:0.00} + {1:0.00}", a,b);
}
public SoPhuc cong2SP(SoPhuc x)
{
    SoPhuc tong = new SoPhuc();
    tong.a = a + x.a;
    tong.b = b + y.b;
}

```

```

        return tong;
        //viet gon: return new SoPhuc(a + x.a, b + x.b);
    }
    public bool soSanhBang(SoPhuc x)
    {
        return (a == x.a && b == x.b);
    }
    //-----operator-----
    public static SoPhuc operator+(SoPhuc x, SoPhuc y)
    {
        return new SoPhuc(x.a + y.A, x.b + y.B);
    }
    public static bool operator==(SoPhuc x, SoPhuc y)
    {
        return (x.A == y.A && x.B == y.B);
    }
}

```

Hàm Main

```

static void Main(string[] args)
{
    SoPhuc x = new SoPhuc(2.3f, 4.0f);
    SoPhuc y = new SoPhuc(3.5f, 8.8f);
    x.xuatSP(); y.xuatSP();
    SoPhuc t = x.cong2SP(y);
    Console.WriteLine("Tong 2 so phuc (ham binh thuong:");
    t.xuatSP();
    Console.WriteLine("Tong 2 so phuc (dung operator: ");
    SoPhuc t1 = x + y;
    t1.xuatSP();
    Console.WriteLine(" Ket qua so sanh bang 2 so phuc: cach 1
{0} - cach 2: {1}", x.soSanhBang(y), x==y);
    Console.ReadLine();
}

```

Bài tập sinh viên tự làm trên lớp:

Bài 1.

Xây dựng lớp PS mô tả phân số trong toán học, chứa các thông tin tử số và mẫu số (mẫu số $\neq 0$) cùng các phương thức theo yêu cầu sau:

- 3 phương thức khởi tạo.
- phương thức nhập/xuất thông tin phân số
- Rút gọn phân số.
- Các hàm: cộng, trừ, nhân, chia, so sánh 2 phân số (Viết dạng hàm bình thường, không phải nạp chồng toán tử)
- Sau đó viết lại các nạp chồng toán tử (operator) dành cho 2 phân số: $+$, $-$, $*$, $/$, $>$, $<$, $>=$, $<=$, $==$, $!=$, $++$, $--$, ép kiểu số nguyên sang phân số, phân số thành số nguyên, số thực thành phân số, phân số thành số thực.
- Operator lũy thừa ($^$) phân số lên n lần.
- Operator ép kiểu một phân số về số nguyên bằng hoặc lớn hơn nhưng gần phân số đó nhất. Ví dụ: phân số $8/2$ sẽ ép kiểu về 4, còn phân số $9/4$ sẽ ép kiểu về 3.
- Đổi phân số thành hỗn số
Ví dụ: $35/8 = 4 \frac{3}{8}$; $17/19 = 0 \frac{17}{19}$
- Viết chương trình nhập/xuất thông tin và gọi các hàm đã viết.

Bài 2.

Xây dựng lớp HonSo (hỗn số) để chứa các đối tượng hỗn số trong toán học. Hỗn số bao gồm 2 thành phần thuộc tính: phần nguyên (số nguyên) và phần phân số (thuộc lớp PS đã xây dựng ở bài 1).

Yêu cầu:

- Hàm truy cập get/set
- 3 hàm khởi tạo
- Hàm nhập/xuất
- Hàm biến đổi hỗn số thành phân số, và ngược lại
- Viết các operator: $+$, $-$, $*$, $/$, $>$, $<$, $.=$, $<=$, $==$, $!=$ giữa 2 hỗn số.

- Các operator ++, --, ép hỗn số về số thực, số thực thành hỗn số.

3. Bài tập về nhà

Bài 3. Xây dựng lớp (TinhTienDien) để tính số điện đã sử dụng và tiền điện (TĐ) cho các hộ gia đình trong một khu dân cư. Thông tin gồm họ tên chủ hộ, mã số công tơ điện, chỉ số cũ, chỉ số mới. Tiền điện của các hộ gia đình sẽ được tính theo công thức sau:

- Nếu số điện sử dụng (SĐSD) nhỏ hơn định mức (ĐM) thì $TĐ = SĐSD * \text{đơn giá trong định mức (ĐGTĐM)}$.
- Ngược lại giá tiền mỗi số điện vượt định mức sẽ được tính gấp rưỡi so với ĐGTĐM. ĐM và ĐGTĐM áp dụng chung cho tất cả các hộ và có thể thay đổi theo thời gian.

Yêu cầu: Phương thức tính số điện phải được viết dưới dạng property của lớp.

Viết chương trình cho phép người dùng nhập vào thông tin và các chỉ số điện của 1 hộ gia đình, sau đó xuất ra thông tin và tiền điện của chủ nhà.

Lưu ý: Giá trị của định mức và đơn giá định mức áp dụng cho tất cả các hộ gia đình, lấy theo giá thực tế và có thể thay đổi theo thời gian.

Bài 4. Tạo lớp Point2D để chứa các đối tượng điểm trong mặt phẳng. Point2D gồm 2 thành phần hoành độ (x) và tung độ (y). Cần viết các hàm sau:

- Hàm truy cập get/set
- 3 hàm khởi tạo & hàm hủy.
- Hàm nhập/xuất điểm.
- Hàm kiểm tra điểm có thuộc trung Ox (hoặc Oy) không?
- Hàm thay đổi vị trí điểm trên mặt phẳng:
 - + void thayDoiViTri (int tx, int ty); // cộng hoành độ/tung độ của điểm với tx/ty.
 - + void thayDoiViTri (Point2D a); // cộng hoành độ/tung độ của điểm với hoành độ/tung độ của a.


- Hàm tính khoảng cách giữa 2 điểm:
 - + `float tinhKhoangCach ();` //tính khoảng cách giữa điểm hiện hành (this) với gốc tọa độ $O(0;0)$.
 - + `float tinhKhoangCach (Point2D a);` //tính khoảng cách giữa điểm hiện hành (this) với điểm a.
- Hàm tìm điểm đối xứng của điểm hiện hành qua trục Ox.
- Hàm tìm điểm đối xứng của điểm hiện hành qua trục Oy.
- Hàm kiểm tra xem đoạn thẳng tạo bởi 2 điểm $A(x_A, y_A)$ và $B(x_B, y_B)$ có song song với trục Ox không?
- Hàm kiểm tra xem đoạn thẳng tạo bởi 2 điểm $A(x_A, y_A)$ và $B(x_B, y_B)$ có song song với trục Oy không?
- Hàm kiểm tra xem đoạn thẳng tạo bởi 2 điểm $A(x_A, y_A)$ và $B(x_B, y_B)$ có đi qua gốc tọa độ $O(0,0)$ không?

Bài 5. Tạo lớp `DinhThucCap2` để mô tả đối tượng định thức cấp 2 trong toán học. Định thức cấp 2 chỉ chứa 1 thông tin là ma trận cấp 2.

Viết các hàm bổ sung vào lớp như sau:

- Hàm `get/set`
- Hàm khởi tạo/hủy
- Hàm nhập định thức bằng cách tạo các phần tử là các số nguyên ngẫu nhiên thuộc đoạn $(-50 \rightarrow 50)$.
- Hàm xuất định thức.
- Tính `det` của định thức.
- Tính tổng 2 định thức (viết 2 dạng bình thường và operator)
- Tính nhân định thức với 1 số nguyên (viết 2 dạng bình thường và operator).

--HẾT--

Trường: ĐH CNTP TP.HCM Khoa: Công nghệ Thông tin Bộ môn: Công nghệ phần mềm MH: TH Lập trình hướng đối tượng	BÀI 4. DANH SÁCH ĐỐI TƯỢNG	
---	---------------------------------------	--

A. MỤC TIÊU:

- Cài đặt được lớp danh sách các đối tượng sử dụng List.
- Đọc được danh sách đối tượng từ một file XML.
- Sắp xếp, trích lọc được các đối tượng trong danh sách đã có
- Gọi được các đối tượng trong hàm chương trình.

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

1. Tóm tắt lý thuyết

a. Danh sách đối tượng sử dụng List:

▪ Cú pháp khai báo List:

```
List<Kiểu DL> <tenlist> = new List<kiểu DL>();
List<Kiểu DL> <tenlist> = new List<kiểu DL>(Số pt);
```

▪ Thêm một đối tượng vào List:

```
<tenlist>.Add(<tendoituong>);
```

▪ Sắp xếp các đối tượng theo thứ tự của property A:

```
List<Kiểu DL> <tenlist>.OrderBy(t=>t.A);
```

▪ Sắp xếp các đối tượng theo thứ tự của property A, nếu property A bằng nhau thì sắp xếp theo property B:

```
List<Kiểu DL>
    <tenlist>.OrderBy(t=>t.A).ThenBy(t=>t.B);
```

▪ Sắp xếp giảm dần: **OrderByDescending** và **ThenByDescending**

▪ Lọc (filter) các đối tượng thỏa yêu cầu theo property A:

```
List<Kiểu DL> <tenlist>.Where(t.A=>điều kiện);
```

- **Duyệt List: có thể sử dụng câu lệnh for hoặc foreach:**

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

```
    //xử lý tenlist[i]
```

Hoặc dùng cấu trúc:

```
foreach(<Kiểu dữ liệu> <biến> in <tenlist>)
```

```
    //xử lý <biến>;
```

b. Đọc dữ liệu từ file XML:

Sử dụng các đối tượng: XmlDocument, XmlNodeList, XmlNode

- **Load file XML:**

```
XmlDocument <tenbien> = new XmlDocument();
```

```
<tenbien>.Load(file);
```

- **Lấy danh sách các đối tượng:**

```
XmlNodeList <tenListNode> = read.SelectNodes("duong dan");
```

```
foreach (XmlNode <bien> in <tenListNode>)
```

```
{
```

```
    //xử lý <bien>["tên dữ liệu"].InnerText;
```

```
    //xử lý trên <bien>.Attributes["..."].Value;
```

```
}
```

2. Bài tập thực hành trên lớp

Bài tập có hướng dẫn

Bài 1. Thông tin của một giáo viên hướng dẫn thực hành gồm: Tên giáo viên, Số nhóm hướng dẫn. Hãy xây dựng lớp danh sách giáo viên để lưu thông tin của tất cả các giáo viên hiện đang dạy thực hành trong học kỳ này.

Yêu cầu:

- Tính tổng số nhóm hướng dẫn thực hành của tất cả giáo viên.
- Sắp xếp tăng dần danh sách giáo viên theo họ tên.
- Sắp xếp giảm dần danh sách giáo viên theo số nhóm hướng dẫn thực hành.
- Danh sách giáo viên có số nhóm hướng dẫn thực hành lớn hơn 1.

- Đọc danh sách thông tin giáo viên từ file XML cho trước.
- Xuất danh sách giáo viên ra màn hình.

Thông tin giáo viên được lưu trong file **GV.xml** có cấu trúc như sau:

```
<?xml version="1.0" encoding="utf-8" ?>
<DanhSach>
  <GV>
    <Hoten>Lâm Thị Họa Mi</Hoten>
    <SoNhom>2</SoNhom>
  </GV>
  <GV>
    <Hoten>Dương Thị Mộng Thùy</Hoten>
    <SoNhom>1</SoNhom>
  </GV>
  <GV>
    <Hoten>Nguyễn Thị Thùy Trang</Hoten>
    <SoNhom>1</SoNhom>
  </GV>
  <GV>
    <Hoten>Nguyễn Văn Thịnh</Hoten>
    <SoNhom>1</SoNhom>
  </GV>
</DanhSach>
```

Hướng dẫn

- Xây dựng lớp thông tin giáo viên (GiaoVien):

```
public class GiaoVien
{
    string TenGV;
    public string Ten
    {
        get { return TenGV; }
        set { TenGV = value; }
    }
    int SoNhom;
    public int SNhom
    {
        get { return SoNhom; }
        set { SoNhom = value; }
    }
    public void xuat()
    {
        Console.WriteLine("{0} \t {1}", TenGV, SoNhom);
    }
}
```

- Xây dựng lớp danh sách giáo viên (DSGiaoVien) với phương thức đọc dữ liệu từ file XML:

```
public class DSGiaoVien
{
    List<GiaoVien> lst = new List<GiaoVien>();
    public List<GiaoVien> ListGV
    {
        get { return lst; }
        set { lst = value; }
    }
    public void nhap(string file)
    {
        XmlDocument read = new XmlDocument();
        read.Load(file);
        XmlNodeList nodelist = read.SelectNodes("/DanhSach/GV");
        foreach (XmlNode node in nodelist)
        {
            //Mỗi node tạo ra một giáo viên
            GiaoVien gv = new GiaoVien();
            gv.Ten = node["Hoten"].InnerText;
            gv.SNhom = int.Parse(node["SoNhom"].InnerText);
            lst.Add(gv); //đưa gv vừa tạo vào lst
        }
    }
}
```

- Cài đặt các phương thức sắp xếp và tính tổng số nhóm:

```
public int TongSonhom()
{
    int tong = 0;
    foreach (GiaoVien gv in lst)
        tong = tong + gv.SNhom;
    return tong;
}
public void SapXepTen()
{
    lst = lst.OrderBy(t => t.Ten).ToList(); ;
}
public void SapXepTheoNhom()
{
    lst = lst.OrderByDescending(t => t.SNhom).ToList();
}
```

- Cài đặt phương thức lọc danh sách nhóm theo điều kiện:

```
public List<GiaoVien> LocSoNhom1()
{
    return lst.Where(t => t.SNhom > 1).ToList();
}
```

- Cài đặt phương thức xuất danh sách giáo viên:

```
public void xuat()
{
    Console.WriteLine("Danh sách giáo viên");
    foreach (GiaoVien gv in lst)
        gv.xuat();
}
```

▪ Hàm main

```
static void Main(string[] args)
{
    DSGiaoVien ds = new DSGiaoVien();
    ds.nhap("GV.xml");
    ds.xuat();
    Console.WriteLine("Tong so nhom cua tat ca gv la {0} ", ds.TongSonhom());

    Console.WriteLine("DANH SACH GIAO VIEN SAP THEO NHOM GIAM DAN");
    ds.SapXepTheoNhom();
    ds.xuat();

    List<GiaoVien> dsloc = ds.LocSpNhom1();
    Console.WriteLine("DANH SACH GIAO VIEN DAY HON 1 NHOM LA");
    foreach (GiaoVien gv in dsloc)
        gv.xuat();
    Console.ReadLine();
}
```

Bài tập sinh viên tự làm trên lớp:

Bài 2. Thông tin mỗi thí sinh gồm: Số báo danh, tên thí sinh, năm sinh, điểm toán, điểm văn, và điểm ngoại ngữ. Xây dựng lớp thí sinh với các yêu cầu sau:

- Xây dựng 3 phương thức khởi tạo.
- Xây dựng property tính tổng điểm biết rằng tổng điểm sẽ là tổng điểm 3 môn với môn ngoại ngữ tính hệ số 2.
- Xây dựng property kết quả. Biết rằng thí sinh sẽ có kết quả “Đậu” nếu tổng điểm lớn hơn điểm chuẩn ngược lại kết quả là “Rớt”. Điểm chuẩn áp dụng cho tất cả thí sinh và do nhà trường qui định, hiện tại điểm chuẩn có giá trị là 25.
- Xây dựng phương thức xuất thí sinh (SBD, họ tên, năm sinh, điểm 3 môn, tổng điểm và kết quả)

Xây dựng lớp danh sách thí sinh để nhà trường có thể quản lý thông tin của tất cả các thí sinh dự thi vào trường với các yêu cầu sau:

- Xây dựng phương thức nhập danh sách thí sinh từ bàn phím
- Xây dựng phương thức nhập danh sách thí sinh từ file XML
- Xây dựng phương thức in ra màn hình danh sách thí sinh
- Sắp xếp danh sách thí sinh theo tổng điểm giảm dần

- e) Sắp xếp thí sinh tăng dần theo họ tên, nếu tên trùng nhau thì thí sinh nào có điểm toán lớn hơn thì đứng trên
- f) Xây dựng phương thức lấy ra danh sách các thí sinh có kết quả “Đậu”
- g) Xây dựng phương thức lấy ra danh sách sinh viên có năm sinh >1995 hoặc có điểm toán từ 9 trở lên.

SV tự tạo file xml có ít nhất 10 sinh viên với file có cấu trúc như sau:

```
<DSSV>
    <ThiSinh>
        <SBD>...</SBD>
        <HT>...</HT>
        <NamSinh>...</NamSinh>
        <Toan>...</Toan>
        <Van>...</Van>
        <NN>...</NN>
    </ThiSinh>
    .....
</DSSV>
```

3. Bài tập về nhà

Bài 3. Xây dựng lớp lưu danh sách điểm trong không gian 2 chiều Oxy với các yêu cầu:

- a) Xây dựng phương thức nhập n điểm từ bàn phím
- b) Xây dựng phương thức nhập điểm từ file xml có cấu trúc do sinh viên qui định
- c) Xây dựng phương thức sắp xếp các điểm có khoảng cách tới gốc tọa độ giảm dần. Nếu 2 điểm có khoảng cách tới gốc tọa độ bằng nhau thì điểm nào có hoành độ lớn hơn được ưu tiên
- d) Xây dựng phương thức lấy ra tất cả các điểm nằm trong đường tròn tâm O bán kính 1

- e) Xây dựng phương thức lấy ra tất cả các điểm nằm trong góc phần tư thứ nhất.
- f) Xóa khỏi danh sách các điểm có hoành độ lớn hơn 5 nhưng nhỏ hơn 8.
- g) Đếm xem danh sách có bao nhiêu điểm có hoành độ là số dương.

Bài 4. Một nhân viên trong công ty ABC có các thông tin: Mã nhân viên, tên nhân viên, phòng ban, chức vụ, hệ số lương, thâm niên công tác và số ngày làm việc trong tháng. Chức vụ chỉ nhận 2 giá trị là “Lãnh đạo” và “Nhân viên”. Lương của nhân viên hàng tháng được tính theo công thức:

Lương = hệ số lương * Lương cơ bản * hệ số thi đua + 1100 + Phụ cấp
 Trong đó lương cơ bản do nhà nước qui định và áp dụng cho tất cả các nhân viên có giá trị tại thời điểm hiện tại là 1210.

Hệ số thi đua được xác định như sau:

- Nếu chức vụ là “Lãnh đạo” thì hệ số là 1
- Nếu chức vụ là “Nhân viên” thì hệ số thi đua được xác định như sau:
 - Nếu số ngày đi làm lớn hơn 22 thì hệ số thi đua bằng 1.0
 - Nếu số ngày đi làm lớn hơn 20 thì hệ số thi đua bằng 0.8
 - Còn lại hệ số thi đua là 0.6
- Phụ cấp được xác định như sau:
 - Nếu chức vụ là “Lãnh đạo” thì phụ cấp chức vụ là 2000
 - Nhân viên không có phụ cấp chức vụ


Xây dựng lớp Nhân viên theo mô tả như trên.

Xây dựng lớp danh sách nhân viên để lưu được các nhân viên của công ty với các yêu cầu quản lý như sau:

- a) Thông tin nhân viên được nhập từ danh sách từ file xml do công ty qui định (sinh viên tự đặt ra qui định của file xml thay công ty ABC).
- b) Xuất thông tin tất cả các nhân viên trong công ty

- c) Xuất ra được thông tin nhân viên theo từng nhóm hệ số thi đua
- d) Lập được danh sách nhân viên theo phòng cho trước.
- e) Lấy ra danh sách các nhân viên có chức vụ là “Lãnh đạo”
- f) Tính tổng lương công ty phải trả cho toàn bộ nhân viên
- g) Loại bỏ các nhân viên có số ngày làm nhỏ hơn 10 trong danh sách.
- h) Lấy ra danh sách những nhân viên không phải là lãnh đạo mà có số ngày làm việc lớn hơn 22.
- i) Lấy ra danh sách các nhân viên có hệ số lương từ 4.34 trở lên và ở phòng “Tài vụ”

--HẾT--

Trường: ĐH CNTP TP.HCM Khoa: Công nghệ Thông tin Bộ môn: Công nghệ phần mềm MH: TH Lập trình hướng đối tượng	BÀI 5. ÔN TẬP	
---	----------------------	--

A. MỤC TIÊU:

- Phân tích bài toán, xây dựng lớp đối tượng cần thiết
- Cài đặt được lớp theo yêu cầu

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

1. Tóm tắt lý thuyết

Các bước giải quyết bài toán

- Xác định các lớp cần phải xây dựng: lớp đối tượng, lớp danh sách các đối tượng
- Xác định thành phần thuộc tính, thành phần phương thức của lớp
- Cài đặt trên C#

2. Bài tập tự làm

Bài 1: Thông tin một khóa học tại một trung tâm tin học gồm: mã khóa học, tên khóa học, số tiết, giờ học, số học viên và giáo viên giảng dạy. Trong đó giờ học có chỉ có 3 giá trị là “3,5,7”, hoặc “2,4,6”, hoặc “7,CN”. Học phí của mỗi khóa học được xác định bằng số tiết học * Học phí 1 tiết * Hệ số giờ học. Các khóa học có thể trùng tên nhưng mã khác nhau.

Trong đó nếu giờ học là “3,5,7” hoặc “2,4,6” thì hệ số giờ học = 1.0. Nếu giờ học là “7,CN”, thì hệ số giờ học là 1.2. Học phí 1 tiết học là do trung tâm qui định và thay đổi theo thời gian. Hiện tại học phí một tiết học của trung tâm là 20.

Thù lao trung tâm trả cho giáo viên một tiết là 100. Số lượng tối thiểu của một khóa học là 10. Cứ mỗi sinh viên vượt qua số lượng tối thiểu của khóa học thì giáo viên giảng dạy sẽ được trung tâm trả thêm một tiết là 2.

Xây dựng lớp KhoaHoc được mô tả như trên với các yêu cầu:

- Có 3 phương thức khởi tạo. Phương thức khởi tạo mặc định sẽ tạo ra khóa học có mã là “TH023”, tên khóa học là “Android Programming”, số tiết là 48 và thời gian học là “2,4,6”. Số lượng học viên là 13, và giáo viên giảng dạy là tên của bạn. Nếu trong phương thức khởi tạo có tham số mà giờ học không phải là 3 trường hợp qui định thì lấy mặc định là “3,5,7”.
- Xây dựng property tính học phí của khóa học
- Xây dựng phương thức tính thù lao của giáo viên trong lớp học đó.
- Xây dựng toán tử + một khóa học với 1 số nguyên n để tạo ra một khóa học mới có số giờ học so với khóa học cũ n tiết học
- Xây dựng phương thức xuất khóa học (mã, tên, số tiết, buổi học, học phí)

Trong trung tâm có rất nhiều khóa học và các thông tin của các khóa học này được lưu trữ trong file xml có định dạng như sau:

```
<KhoaHocs>
```

```
  <KhoaHoc>
```

```
    <Ma>...</Ma>
```

```
    <Ten>...</Ten>
```

```
    <SoTiet>...</SoTiet>
```

```
    <GioHoc>...</GioHoc>
```

```

        <SoLuongHV>...</SoLuongHV>
        <TenGV>...</TenGV>
    </KhoaHoc>
    <KhoaHoc>
        .....
    </KhoaHoc>
</KhoaHocs>

```

Hãy xây dựng lớp DSKhoaHoc để giúp trung tâm quản lý được các khóa học hiện tại của trung tâm với các yêu cầu sau

- Đọc danh sách các khóa học từ file theo định dạng đã cho.
- Tính tổng tiền số tiền mà trung tâm thu về sau khi trả thù lao cho giáo viên.
- Tính tổng số lớp mà trung tâm đang có.
- Lấy ra danh sách các khóa học mà có lượng sinh viên từ 15 trở lên
- Cho biết hiện tại giáo viên có tên x đang dạy bao nhiêu khóa học.
- Sắp xếp theo học phí giảm dần của các khóa học trong trung tâm.
- In ra màn hình các khóa học có tên x trong trung tâm.
- In ra danh sách các khóa học của trung tâm theo từng giờ học (in riêng các khóa học có giờ là “2,4,6”, “3,5,7” và “7,CN”).

Bài 2: Thông tin của một thí sinh thi chứng chỉ tại trung tâm tin học của trường gồm có: Số báo danh, Họ và tên lót, Tên thí sinh, năm sinh và loại chứng chỉ, ca thi. Biết rằng loại chứng chỉ chỉ có thể nhận các giá trị là: CCA, CCB_Access, CCB_Autocab, CCB_Excel. Lệ phí thi được xác định tùy vào

loại chứng chỉ: Nếu loại chứng chỉ là CCA thì lệ phí là 200, CCB lệ phí là 300. Ca thi chỉ có 1 trong 6 giá trị là 1, 2, 3, 4, 5, 6.

Xây dựng lớp ThiSinh theo các thông tin được mô tả như trên với các yêu cầu:

- Xây dựng 3 phương thức khởi tạo: Phương thức khởi tạo mặc định tạo ra một thí sinh có số báo danh “BD01”, Họ và tên lót, Tên thí sinh, năm sinh lấy thông tin của SV và loại chứng chỉ là “CCA”, ca thi là 1. Phương thức khởi tạo có tham số, nếu loại chứng chỉ không nằm trong các giá trị đã nêu ở trên thì lấy giá trị là “CCA” và phương thức copy constructor.
- Xây dựng property ca thi.
- Xây dựng property tính lệ phí thi của sinh viên.
- Xây dựng phương thức xuất khóa học (SBD, Họ và tên, năm sinh, loại chứng chỉ và ca thi).

Một đợt thi có rất nhiều thí sinh và thông tin các thí sinh được lưu trong file xml có định dạng như sau:

```
<KyThi>
  <ThiSinh>
    <SBD>...</SBD>
    <Ho>...</Ho>
    <Ten>...</Ten>
    <NS>...</NS>
    <LoaiCC>...</LoaiCC>
    <Ca>...</Ca>
  </ThiSinh>
  <ThiSinh>
    .....
```


</ThiSinh>

</KyThi>

Hãy xây dựng lớp DSThiSinh để giúp trung tâm quản lý được các danh sách thí sinh thi trong đợt này

- Đọc danh sách các khóa học từ file theo định dạng đã cho.
- Nhập dữ liệu cho file xml có tối thiểu 5 thí sinh
- Tính tổng tiền số tiền lệ phí mà trung tâm thu được.
- Cho biết số lượng thí sinh của đợt thi.
- Lấy ra danh sách các thí sinh thi “CCA”.
- Sắp xếp thí sinh theo danh sách loại chứng chỉ. Nếu cùng loại chứng chỉ thì sắp xếp theo tên thí sinh.
- In ra màn hình thông tin thí sinh có số báo danh là x.

--HẾT--

Trường: ĐH CNTP TP.HCM Khoa: Công nghệ Thông tin Bộ môn: Công nghệ phần mềm MH: TH Lập trình hướng đối tượng	BÀI 6. KẾ THỪA	
---	-----------------------	--

A. MỤC TIÊU:

- Xác định được mối quan hệ cụ thể và trừu tượng.
- Xây dựng được lớp có quan hệ kế thừa
- Xây dựng được phương thức khởi tạo của lớp con trong quan hệ kế thừa
- Xây dựng được phương thức override

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

1. Tóm tắt lý thuyết

Kế thừa trong lập trình là cách một lớp có thể **thừa hưởng** lại những **thuộc tính**, **phương thức** từ **một lớp khác** và sử dụng chúng như là của bản thân mình.

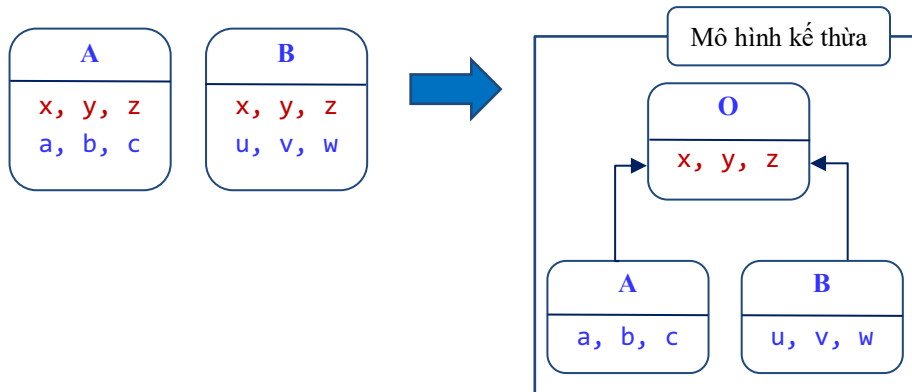
Một định nghĩa trừu tượng hơn về **kế thừa**: là một đặc điểm của ngôn ngữ hướng đối tượng dùng để biểu diễn mối quan hệ **cụ thể hóa (đặc biệt hoá) – (trừu tượng hóa) tổng quát hoá** giữa các lớp.

Ưu điểm của kế thừa:

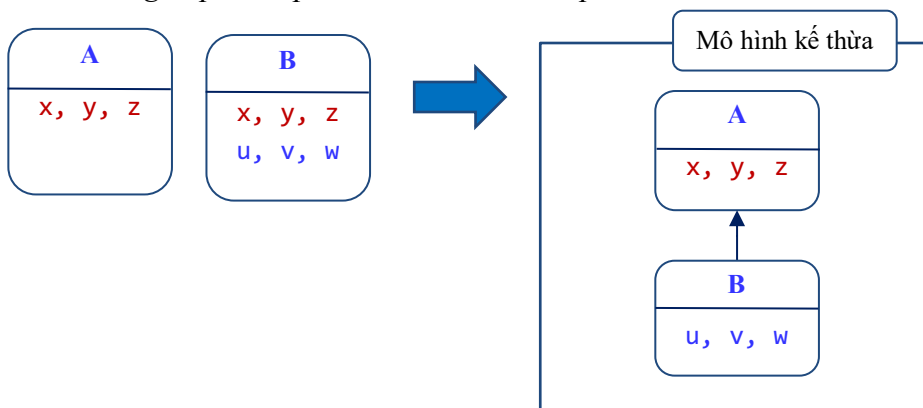
- Cho phép xây dựng một lớp mới từ lớp đã có:
 - Lớp mới gọi là **lớp con (subclass)** hay **lớp dẫn xuất (derived class)**.
 - Lớp đã có gọi là **lớp cha (superclass)** hay **lớp cơ sở (base class)**.
- Cho phép chia sẻ các thông tin chung nhằm tái sử dụng và đồng thời giúp ta dễ dàng nâng cấp, dễ dàng bảo trì.

a. Phương pháp xây dựng lớp khi tồn tại quan hệ cụ thể hóa – trừu tượng hóa

➤ Trường hợp 1: giữa 2 lớp A và B tồn tại một số đặc điểm chung



➤ Trường hợp 2: Lớp B chứa hoàn toàn lớp A



b. Cú pháp xây dựng quan hệ lớp kế thừa

```
[Từ khóa truy cập] class <Tên_Lớp_Ch>
{
    //Nội dung của lớp cha
}
[Từ khóa truy cập] class
<Tên_Lớp_Con>:<Tên_Lớp_Ch>
{
    // Nội dung phần bổ sung của lớp con
}
```

c. Phương thức khởi tạo trong quan hệ kế thừa

Trong quan hệ kế thừa, phương thức khởi tạo không được kế thừa. Phương thức khởi tạo mặc định (phương thức khởi tạo không tham số) của lớp cha luôn luôn được gọi mỗi khi có một đối tượng thuộc lớp con khởi tạo. Và được gọi trước phương thức khởi tạo của lớp con.

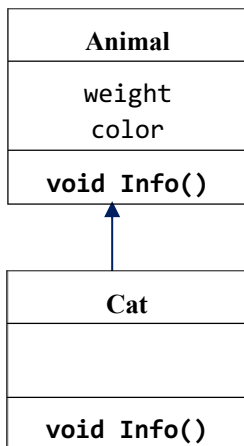
Nếu như lớp cha có phương thức khởi tạo có tham số thì đòi hỏi lớp con phải có phương thức khởi tạo tương ứng và thực hiện gọi phương thức khởi tạo của lớp cha thông qua từ khoá `base`.

Cú pháp tạo phương thức khởi tạo có tham số của lớp con gọi phương thức khởi tạo có tham số của lớp cha:

```
public <Tên_Lớp_Con>(<ds_tham_số>):base(<danh_sách_tham_số>)  
{  
    //gán giá trị cho các thành phần dữ liệu bổ sung của lớp con  
}
```

d. Phương thức trùng tên trong quan hệ kế thừa

Ví dụ:



- Phương thức của lớp cha được khai báo

```
virtual public <Kiểu trả về> <Tên_PT>(ds Tham số) {...}
```

- Phương thức của lớp con phải được khai báo

```
public override <Kiểu trả về> <Tên_PT>(ds Tham số) {...}
```

Override method (Ghi đè phương thức)

- Khái niệm ghi đè (overriding) được dùng để định nghĩa lại phương thức của lớp cơ sở (lớp cha) trong lớp dẫn xuất (lớp con kế thừa).
- Các điểm cần lưu ý khi thực hiện ghi đè phương thức:
 - Phương thức ở lớp cơ sở và lớp dẫn xuất phải có cùng dạng hàm (signature) và kiểu dữ liệu trả về.
 - Phương thức lớp cơ sở phải được khai báo với từ khóa `virtual`.
 - Phương thức lớp dẫn xuất phải được khai báo với từ khóa `override`.
 - Không thể ghi đè các phương thức `static`.

2. Bài tập thực hành trên lớp

Bài tập có hướng dẫn

Bài 1. Xây dựng lớp `NhanVien` để lưu thông tin của các nhân viên trong công ty sản xuất ABC biết rằng mỗi nhân viên đều có các thông tin sau: Mã nhân viên, tên nhân viên, năm vào làm, hệ số lương, số ngày nghỉ trong tháng.

- Xây dựng các phương thức khởi tạo: Phương thức khởi tạo không tham số; Phương thức khởi tạo 3 tham số là mã, tên và hệ số lương sẽ tạo ra nhân viên có mã, tên, hệ số lương đã đưa vào, năm vào làm là năm hiện tại và chưa nghỉ bất kỳ ngày nào; Phương thức khởi tạo có đầy đủ các thông tin của nhân viên
- Xây dựng phương thức tính phụ cấp thâm niên cho nhân viên: Biết rằng nếu nhân viên nào làm từ 5 năm trở lên thì bắt đầu được tính thâm niên (PCTN) theo công thức:

$\text{Phụ cấp thâm niên} = \text{Số năm làm việc} * \text{Lương cơ bản} / 100$

- Xây dựng phương thức xét thi đua cho nhân viên. Nếu số ngày nghỉ ≤ 1 thì xếp loại 'A', số ngày nghỉ ≤ 3 thì xếp loại 'B' còn lại là xếp loại 'C'.

- Xây dựng phương thức tính lương cho nhân viên. Biết rằng lương được tính theo công thức:

$$\text{Lương} = \text{Lương cơ bản} * \text{hệ số lương} * \text{hệ số thi đua} + \text{PCTN}$$

Trong đó hệ số thi đua được xác định thông qua xếp loại như sau:

- Nếu xếp loại A: hệ số thi đua = 1.0
 - Nếu xếp loại B: hệ số thi đua = 0.75
 - Nếu xếp loại C: hệ số thi đua = 0.5
- Xây dựng phương thức xuất và nhập thông tin nhân viên

Trong đó lương cơ bản do nhà nước qui định và áp dụng cho tất cả các nhân viên. Hiện tại lương cơ bản có giá trị là 1150.

Hướng dẫn

Xây dựng được lớp Nhanvien như sau:

```
public class NhanVien
{
    string ma, ten;
    int NamVL, SoNgayNghỉ;
    double HSL;

    //Cac property...
    public NhanVien()
    {
        ma = "NV01";
        ten = "Ho ten SV";
        NamVL = 2006;
        SoNgayNghỉ = 1;
        HSL = 3.0;
    }
}
```

Xây dựng phương thức khởi tạo có tham số

```
public NhanVien(string maso, string hoten, double hsLuong)
{
    ma = maso;
    ten = hoten;
    HSL = hsLuong;
    NamVL = DateTime.Today.Year;
    SoNgayNghỉ = 0;
}

public NhanVien(string ma, string ten, int nvl, int nghi, double hsluong)
{
    this.ma = ma;
    this.ten = ten;
    this.NamVL = nvl;
    this.SoNgayNghỉ = nghi;
    this.HSL = hsluong;
}
```

Xây dựng các phương thức lương và xếp loại thi đua

```
static int LuongCoban = 1150;
public double PhuCapThamNien()
{
    int thamnien = DateTime.Today.Year - NamVL;
    if (thamnien >= 5)
        return thamnien * NhanVien.LuongCoban / 100.0;
    return 0;
}
public char XepLoai()
{
    if (SoNgayNghỉ <= 1) return 'A';
    if (SoNgayNghỉ <= 3) return 'B';
    return 'C';
}

public double Luong()
{
    double hsThiDua = 0.5;
    char xl = XepLoai();
    if (xl == 'A') hsThiDua = 1.0;
    if (xl == 'B') hsThiDua = 0.75;
    return NhanVien.LuongCoban * HSL * hsThiDua + PhuCapThamNien();
}
```

Các phương thức property và nhập xuất thông tin, sinh viên tự cài đặt.

Bài 2: Ngoài đối tượng nhân viên lao động như đã mô tả trong bài 1. Công ty ABC còn có một đội ngũ làm công tác quản lý để điều hành

toàn bộ các hoạt động của công ty gọi tắt là cán bộ. Hãy xây dựng lớp cán bộ lãnh đạo cho công ty ABC biết rằng cán bộ lãnh đạo cũng là nhân viên của công ty. Tuy nhiên do vai trò và chức năng lãnh đạo mỗi cán bộ sẽ có một chức vụ, phòng ban làm việc và một hệ số phụ cấp lãnh đạo tương ứng.

- Xây dựng phương thức khởi tạo mặc định tạo ra một lãnh đạo giống như nhân viên nhưng có chức vụ là trưởng phòng làm việc tại phòng hành chính và có hệ số phụ cấp chức vụ là 5.0
- Xây dựng phương thức khởi tạo có 5 tham số là mã, tên, hệ số lương, chức vụ và hệ số phụ cấp chức vụ sẽ tạo ra một nhân viên có mã, tên, hệ số lương, chức vụ và hệ số chức vụ như đã đưa vào và số ngày nghỉ là 1 và vào làm việc trong công ty là năm hiện tại.
- Xây dựng phương thức khởi tạo có đầy đủ các tham số
- Xây dựng phương thức xét thi đua lại cho cán bộ biết rằng cán bộ của công ty luôn xếp loại A.
- Xây dựng phương thức tính phụ cấp lãnh đạo cho cán bộ biết rằng phụ cấp lãnh đạo được tính = hệ số phụ cấp lãnh đạo * lương cơ bản.
- Xây dựng lại phương thức tính lương của cán bộ biết rằng lương của cán bộ sẽ được tính = Lương + phụ cấp lãnh đạo.
- Xây dựng phương thức nhập, xuất thông tin một cán bộ

Ta thấy Cán bộ lãnh đạo cũng là một Nhân viên nên khi xây dựng ta sẽ cho lớp CanBo kế thừa từ lớp Nhân viên đã xây dựng ở bài 1 và bổ sung thêm các thuộc tính cần thiết: chức vụ, phòng ban và hệ số lãnh đạo.

```

public class CanBo:NhanVien
{
    string chucvu;
    string phongban;
    double hesold;
    public CanBo(): base()
    {
        chucvu = "Truong phong";
        phongban = "Phong Hanh Chinh";
        hesold = 5.0;
    }
    public CanBo(string ma, string ten, double hsl, string chucvu, double hesophucapcv)
        : base(ma, ten,DateTime.Today.Year,1,hsl)
    {
        this.chucvu = chucvu;
        this.hesold = hesophucapcv;
    }
}

```

Trong phương thức khởi tạo 5 tham số, do đề bài yêu cầu tạo ra một cán bộ có các thông số như đã truyền vào còn cần có thêm số ngày nghỉ là 1 nên ta phải gọi phương thức khởi tạo 5 tham số của lớp cha chứ không thể gọi phương thức khởi tạo 3 tham số được vì phương thức khởi tạo 3 tham số sẽ gán số ngày nghỉ bằng 0 là không đúng với yêu cầu của bài.

Tương tự hoàn thành phương thức khởi tạo đầy đủ tham số:

```

public CanBo(string ma, string ten, int nvl,int nghi, double hsl,
               string chucvu, string pb, double hesold)
    :base(ma, ten,nvl,nghi,hsl)
{
    this.chucvu = chucvu;
    this.phongban = pb;
    this.hesold = hesold;
}

```

Vì trong lớp Cán bộ cũng có phương thức xếp loại và tính lương nhưng cách tính có thay đổi so với lớp nhân viên nên cần phải được **override** cho phù hợp. Do đó ta cần phải điều chỉnh lại lớp nhân viên ở bài 1 bằng cách bổ sung các từ khóa **virtual** vào các phương thức mà ta cần phải override như xếp loại, lương, nhập, xuất rồi cài đặt ở lớp cán bộ như sau:

```

public override char XepLoai()
{
    return 'A';
}

public double PhuCapCV()
{
    return hesold * NhanVien.LuongCoban;
}
public override double Luong()
{
    return base.Luong() + PhuCapCV();
}

```

Để kiểm tra tính override và các phương thức của 2 lớp NhanVien và CanBo ta xây dựng hàm main như sau:

```

static void Main(string[] args)
{
    NhanVien nv = new NhanVien();
    nv.nhap(); nv.xuat();
    CanBo cb = new CanBo();
    cb.nhap(); cb.xuat();
    nv = cb; nv.xuat();
    Console.ReadLine();
}

```

Bài tập sinh viên tự làm trên lớp:

Bài 3. Một hàng hóa bao gồm các thông tin: mã hàng và tên hàng. Nước giải khát là một hàng hóa và có thêm các thuộc tính: đơn vị tính, số lượng, đơn giá. Hệ thống quản lý bao gồm các yêu cầu chi tiết như sau:

- a. Lớp hàng hóa:
 - Xây dựng lớp với các thuộc tính trên.
 - Xây dựng phương thức khởi tạo mặc định, mã hàng và tên hàng đều rỗng
 - Xây dựng phương thức khởi tạo có tham số. Mã hàng có kiểu chuỗi có chiều dài là 5 ký tự, 2 ký tự đầu là “HH”, 3 ký tự sau là số. Nếu tham số mã hàng không đúng quy tắc này thì khởi tạo mã hàng là “HH001”
 - Xây dựng phương thức xuất một hàng hóa (**void xuat()**)

b. Lớp Nước giải khát

- Xây dựng lớp Nước giải khát kế thừa lớp hàng hóa và các thuộc tính bổ sung.
- Xây dựng phương thức khởi tạo có tham số. Lưu ý: đơn vị tính chỉ nhận 1 trong 4 giá trị: “kết”, “thùng”, “chai”, “lon”. Nếu đơn vị tính không thuộc 1 trong 4 tham số trên thì gán đơn vị tính là “kết”
- Xây dựng phương thức xuất Nước giải khát (**void xuat()**)
- Xây dựng phương thức tổng tiền dựa vào đơn vị tính (DVT) như sau:
 - $DVT = \text{“kết”}$ hoặc $\text{“thùng”} \rightarrow \text{thành tiền} = \text{số lượng} * \text{đơn giá}$
 - $DVT = \text{“chai”} \rightarrow \text{thành tiền} = \text{số lượng} * \text{đơn giá} / 20$
 - $DVT = \text{“lon”} \rightarrow \text{thành tiền} = \text{số lượng} * \text{đơn giá} / 24$

Tổng tiền = thành tiền * tỉ lệ chiết khấu. Biết rằng tỉ lệ chiết khấu là chung cho các sản phẩm Nước giải khát và do công ty qui định.

3. Bài tập về nhà

Bài 4. Một nhân viên trong công ty có các thông tin sau: mã nhân viên, tên nhân viên, hệ số lương. Cán bộ lãnh đạo là một nhân viên và có thêm thuộc tính là chức vụ, thâm niên quản lý. Hệ thống quản lý bao gồm các yêu cầu chi tiết như sau:

a. Lớp nhân viên (NhanVien)

- Xây dựng lớp với các thuộc tính trên.
- Xây dựng phương thức khởi tạo mặc định, mã nhân viên là “NV001”, tên nhân viên là họ tên sinh viên, hệ số lương mặc định là 2.34.
- Xây dựng phương thức khởi tạo có tham số. Yêu cầu 2 ký tự đầu của mã nhân viên phải là “NV”, nếu không phù hợp thì lấy mã nhân viên mặc định là “NV001”


- Xây dựng phương tính thu nhập (`double thunhap()`) của nhân viên. Biết rằng thu nhập = hệ số lương * lương cơ bản. Trong đó lương cơ bản là do nhà nước qui định và được khởi tạo là 1150
- Xây dựng phương thức xuất() (mã, tên, hệ số, thu nhập)
- b. Lớp cán bộ lãnh đạo (`CBLanhDao`)
 - Xây dựng lớp cán bộ lãnh đạo kế thừa lớp Nhân viên với các thuộc tính bổ sung.
 - Xây dựng phương thức khởi tạo không tham số tạo ra một cán bộ lãnh đạo có mã là “NV009”, tên là “Dieu Hien”, hệ số lương là 4.67, chức vụ “Giám đốc”, thâm niên quản lý là 10 năm
 - Xây dựng phương thức tính thu nhập của cán bộ lãnh đạo theo tắc sau: Lương của cán bộ lãnh đạo được tính giống như lương của nhân viên nhưng có thêm phần phụ cấp lãnh đạo. Biết rằng phụ cấp lãnh đạo được tính bằng $= 1500 * \text{hệ số lãnh đạo}$
 - Nếu chức vụ là “Giám đốc” hệ số lãnh đạo số là 7.0
 - Nếu chức vụ là trưởng phòng thì hệ số lãnh đạo là 6.0
 - Nếu chức vụ là phó phòng thì hệ số lãnh đạo là 4.5
 - Các chức vụ khác hệ số lãnh đạo là 1.0
 - Xây dựng phương thức xuất thông tin của cán bộ lãnh đạo (**`void xuat()`**) (mã, tên, hệ số, thu nhập, chức vụ)

Bài 5. Một người bao gồm các thông tin: họ tên, ngày sinh, giới tính (giới tính là “nam” hoặc “nữ”). Sinh viên của một trường là một người và có thêm các thuộc tính: mã số sinh viên, hệ đào tạo, tổng số tín chỉ. Hệ thống quản lý bao gồm các yêu cầu chi tiết như sau:

- a. Lớp người:
 - Xây dựng lớp với các thuộc tính trên.
 - Xây dựng phương thức khởi tạo mặc định. Giới tính mặc định là “nam”

- Xây dựng phương thức khởi tạo có tham số, nếu tham số giới tính truyền vào không rơi vào một trong hai trường hợp “nam” hoặc “nữ” thì thuộc tính giới tính là “nam”
 - Xây dựng phương thức xuất một người
- b. Lớp sinh viên
- Xây dựng lớp sinh viên kế thừa lớp người và các thuộc tính bổ sung.
 - Xây dựng phương thức khởi tạo có tham số. Thuộc tính hệ đào tạo là một trong 3 giá trị “đại học”, “cao đẳng” và “cao đẳng nghề”. Nếu tham số hệ đào tạo không thuộc một trong ba trường hợp này thì thiết lập hệ đào tạo là “đại học”. Thuộc tính tổng số tín chỉ được tính như sau:
 - Nếu hệ đào tạo là “đại học” thì tổng số tín chỉ là 150
 - Nếu hệ đào tạo là “cao đẳng” thì tổng số tín chỉ là 100
 - Nếu hệ đào tạo là “cao đẳng nghề” thì tổng số tín chỉ là 130
 - Xây dựng phương thức xuất sinh viên
 - Xây dựng phương thức tính tổng học phí của sinh viên như sau: tổng học phí = tổng số tín chỉ * học phí tín chỉ. Trong đó, học phí tín chỉ được tính như sau:
 - Nếu hệ đào tạo là “đại học” thì học phí tín chỉ = 200.000
 - Nếu hệ đào tạo là “cao đẳng” thì học phí tín chỉ = 150.000
 - Nếu hệ đào tạo là “cao đẳng nghề” thì học phí tín chỉ = 120.000

--HẾT--

Trường: ĐH CNTP TP.HCM Khoa: Công nghệ Thông tin Bộ môn: Công nghệ phần mềm MH: TH Lập trình hướng đối tượng	BÀI 7. ĐA HÌNH	
---	-----------------------	--

A. MỤC TIÊU:

- Phân tích được mối quan hệ giữa các lớp
- Xác định được các thành phần chung để xây dựng lớp cha
- Xác định được các phương thức trừu tượng

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

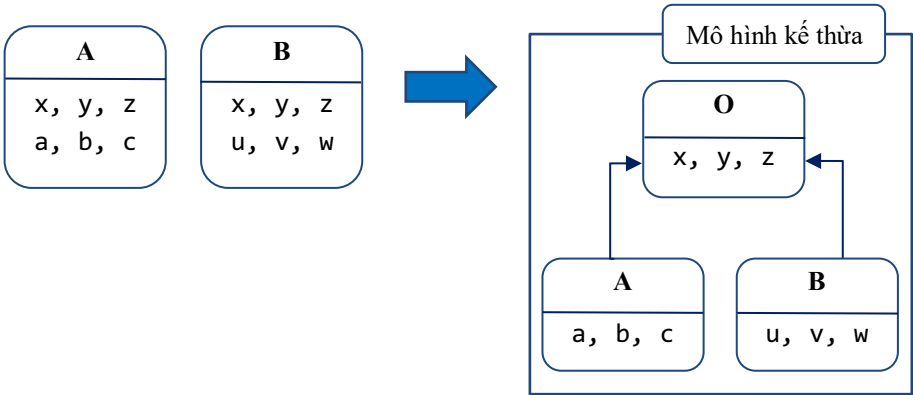
STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

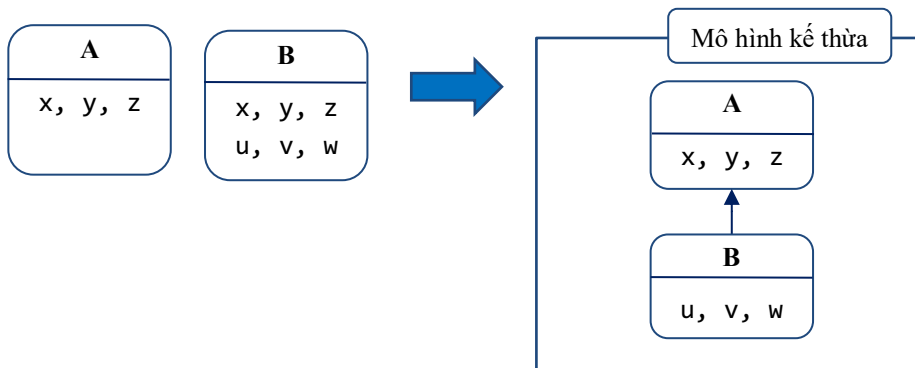
1. Tóm tắt lý thuyết

a. Phương pháp xây dựng lớp khi tồn tại quan hệ cụ thể hóa – trừu tượng hóa

➤ *Trường hợp 1: Giữa 2 lớp A và B có một số đặc điểm chung*



➤ Trường hợp 2: Lớp B chứa hoàn toàn lớp A



b. Lớp trừu tượng (abstract class), phương thức trừu tượng (abstract method)

Lớp trừu tượng, là một trong những khái niệm quan trọng trong lập trình hướng đối tượng, đơn giản được xem như là một lớp cha cho tất cả các lớp có *cùng bản chất* (kiểu, loại, nhiệm vụ của lớp).

Đặc điểm lớp trừu tượng:

- Được khai báo với từ khóa **abstract**.
- Có thể khai báo 0, 1 hoặc nhiều phương thức trừu tượng.
- Không thể khởi tạo đối tượng.

Phương thức trừu tượng là phương thức chỉ có phần khai báo, không có phần định nghĩa.

Lưu ý:

- Một lớp chứa ít nhất một phương thức trừu tượng thì bắt buộc phải khai báo lớp trừu tượng.
- Một lớp thừa kế từ lớp trừu tượng phải hiện thực tất cả các phương thức trừu tượng hoặc lớp đó cũng phải là lớp trừu tượng.

c. Cú pháp khai báo

Đối với lớp

```
[quyền truy cập] abstract class <Tên_lớp>
{
    ...
}
```

Đối với phương thức

- Phương thức trừu tượng trong lớp trừu tượng
[quyền truy cập] **abstract** <Kiểu trả về>

`<Tên_phương_thức>([ds_Thamsố]);`
- Lớp con kế thừa từ lớp trừu tượng thực thi phương thức trừu tượng của lớp cha
[quyền truy cập] **override** <Kiểu trả về>

`<Tên_phương_thức>([ds_Thamsố]){...}`

Lưu ý:

- Phương thức trừu tượng không được khai báo sử dụng từ khóa `virtual`. Vì bản thân từ khóa `abstract` đã bao hàm khái niệm `virtual`.
- Phương thức trừu tượng không thể là phương thức tĩnh (`static`).

2. Bài tập thực hành trên lớp

Bài tập có hướng dẫn

Bài 1. Trong công ty ABC nhân viên được chia làm 3 loại khác nhau trong quá trình vận hành sản xuất là nhân viên sản xuất, nhân viên kinh doanh và cán bộ điều hành. Mỗi nhân viên trong công ty đều có các thông tin chung như: mã nhân viên, tên nhân viên, năm sinh, giới tính, hệ số lương và năm vào làm. Để nâng cao tính tích cực cho nhân viên, công ty tổ chức đánh giá xếp loại thi đua cho nhân viên hàng tháng (loại A, B, C, D). Xếp loại này sẽ quyết định **thu nhập của mỗi nhân viên như sau:**

- Nếu xếp loại A: nhân viên được hưởng 100% Lương + phụ cấp thâm niên
- Nếu xếp loại B: nhân viên được hưởng 75% Lương + phụ cấp thâm niên
- Nếu xếp loại C: nhân viên được hưởng 50% Lương + phụ cấp thâm niên

- Nếu xếp loại D: nhân viên được hưởng 0% Lương + phụ cấp thâm niên.

Biết rằng phụ cấp thâm niên chỉ áp dụng cho các nhân viên có thời gian làm việc từ 5 năm trở lên và được tính như sau:

Phụ cấp thâm niên = số năm làm việc * lương cơ bản/100.

Do tính chất công việc nên mỗi nhân viên lại có những đặc điểm, cách xếp loại, cách tính lương và phụ cấp khác nhau như sau:

- **Nhân viên sản xuất:** do làm việc trong môi trường khắc nghiệt nên nhân viên sản xuất đều được hưởng một hệ số phụ cấp nặng nhọc. Hệ số này là chung cho tất cả các nhân viên sản xuất của công ty và đang được công ty qui định là 10%. Mỗi nhân viên sẽ có thêm thông tin về số ngày nghỉ trong một tháng và **xếp loại thi đua được tính như sau:**

- Nếu số ngày nghỉ ≤ 1 thì xếp loại A,
- $1 < \text{Số ngày nghỉ} \leq 3$ thì xếp loại B ,
- $4 < \text{Số ngày nghỉ} \leq 5$ thì xếp loại C
- Còn lại là xếp loại 'D'.

Lương của nhân viên sản xuất được tính theo công thức như sau:

Lương = hệ số lương * lương cơ bản * (1 + phụ cấp nặng nhọc)

- **Nhân viên kinh doanh:** Công ty không bắt buộc nhân viên kinh doanh phải làm việc theo ngày giống như nhân viên sản xuất mà chỉ quan tâm tới năng lực bán hàng của từng người. Mỗi nhân viên kinh doanh có một doanh số bán hàng tối thiểu khác nhau và một thông tin về doanh số bán hàng theo tháng.

Nếu nhân viên hoàn thành doanh số tối thiểu thì xếp loại thi đua là B, nếu không hoàn thành thì xếp loại C. Nếu doanh số nhỏ hơn 50% doanh số tối thiểu thì xếp loại D. Ngược lại nếu nhân viên có doanh số cao hơn 2.0 lần doanh số tối thiểu thì xếp loại A.

Lương của nhân viên kinh doanh cũng được xác định như sau:

Lương = hệ số lương * lương cơ bản + hoa hồng.

Trong đó hoa hồng là 15% của doanh thu vượt mức doanh thu tối thiểu.

– **Cán bộ quản lý:** Cán bộ quản lý có trách nhiệm quản lý toàn bộ hoạt động sản xuất của công ty. Cán bộ quản lý có thêm thông tin về chức vụ và hệ số chức vụ. Xếp loại thi đua của cán bộ quản lý luôn luôn là A. Và lương của cán bộ quản lý được xác định như sau:

Lương = hệ số lương * lương cơ bản + phụ cấp chức vụ.

Trong đó phụ cấp chức vụ = hệ số phụ cấp chức vụ * 1100.

Giả sử bạn đang làm ở bộ phận tin học trong công ty ABC, hãy phân tích thiết kế lớp để xây dựng chương trình **tính thu nhập** cho tất cả các nhân viên trong công ty. Tính tổng được số tiền mà công ty phải trả lương cho nhân viên.

Lương cơ bản do nhà nước qui định và áp dụng cho tất cả các nhân viên. Hiện tại lương cơ bản có giá trị là 1150.

Hướng dẫn

Do trong công ty có nhiều nhân viên khác nhau nhưng có một số đặc điểm chung nên ta sẽ áp dụng TH1 để xây dựng lớp NhanVien chứa các đặc điểm chung như sau:

```
public class NhanVien
{
    string ma, ten;
    bool gioitinh;
    int nvl,snn;
    double hsl;
    protected static double LuongCoBan = 1150;
    public double PhuCapThamNien()
    {
        int tn = DateTime.Today.Year - nvl;
        if (tn >= 5)
            return tn * NhanVien.LuongCoBan / 100;
        return 0;
    }
    public double ThuNhap()
    {
        char xl = XepLoai();
        double hsThidua=0;
        if (xl == 'A') hsThidua=1.0;
        if (xl == 'B') hsThidua = 0.75;
        if (xl == 'C') hsThidua = 0.5;
        return hsThidua* TinhLuong()+PhuCapThamNien();
    }
}
```


Ngoài các đặc điểm chung đã được cài đặt trong lớp NhanVien như ở trên ta nhận thấy bất kỳ một nhân viên nào cũng có xếp loại thi đua và tính lương của mình. Nhưng do mỗi nhân viên có cách tính khác nhau do đó trong lớp nhân viên ta không thể tiến hành cài đặt cụ thể được. Vì vậy các phương thức như `TinhLuong()`, `XepLoai()` ta phải khai báo là phương thức trừu tượng như sau:

```
abstract public char XepLoai();  
abstract public double TinhLuong();
```

Như vậy trong lớp NhanVien hiện tại đã có 2 phương thức trừu tượng nên lớp NhanVien phải là lớp trừu tượng. Ta cần bổ sung từ khóa **abstract** vào khai báo của lớp nhân viên như sau:

```
abstract public class NhanVien
```

Như vậy ta đã xây dựng được xong thành phần chung nhất của các lớp trong chương trình. Ta tiến hành xây dựng 3 lớp con là NhanVienSX, NhanVienKD và CanBo như sau: (Các phương thức khởi tạo, nhập, xuất sinh viên tự cài đặt).

– Lớp NhanVienSX

```
class NhanVienSX:NhanVien  
{  
    int snn;  
    static double hsPhuCapNangNhoc = 0.1;  
    public override char XepLoai()  
    {  
        if (snn <= 1) return 'A';  
        if (snn <= 3) return 'B';  
        if (snn <= 5) return 'C';  
        return 'D';  
    }  
    public override double TinhLuong()  
    {  
        return hsl * NhanVien.LuongCoBan * (1+ NhanVienSX.hsPhuCapNangNhoc);  
    }  
}
```

– Lớp NhanVienKD

```
class NhanVienKD:NhanVien
{
    double doanhthuTT, doanhthu;
    public override char XepLoai()
    {
        if (doanhthu < 0.5 * doanhthuTT)
            return 'D';
        if (doanhthu < doanhthuTT)
            return 'C';
        if (doanhthuTT < 2 * doanhthuTT)
            return 'B';
        return 'A';
    }
    public override double TinhLuong()
    {
        return hsl * NhanVien.LuongCoBan + HoaHong();
    }
    public double HoaHong()
    {
        char xl = XepLoai();
        double vuot = doanhthu - doanhthuTT;
        if (vuot > 0) return 0.15 * vuot;
        return 0;
    }
}
```

– Lớp CanBo

```
public class CanBo:NhanVien
{
    string chucvu;
    double hspcCV;
    public override char XepLoai()
    {
        return 'A';
    }
    public double PhuCapCV()
    {
        return 1100 * hspcCV;
    }
    public override double TinhLuong()
    {
        return hsl * NhanVien.LuongCoBan + PhuCapCV();
    }
}
```

Bài tập sinh viên tự làm trên lớp:

Bài 2. Mỗi hộ gia đình trong chung cư XYZ đều có các thông tin sau: Tên chủ hộ, địa chỉ, chỉ số nước cũ, chỉ số nước mới. Hãy thiết kế lớp cho chương trình tính tiền nước hàng tháng cho tất cả các hộ gia đình trong chung cư XYZ. Biết rằng tiền nước của mỗi hộ gia đình được tính bằng công thức:

$$\text{Tiền nước} = \text{Số nước phải trả tiền} * 8000$$

Các hộ gia đình trong chung cư XYZ được chia làm 3 loại với các tính số nước phải trả tiền cũng khác nhau như sau:

- **Hộ gia đình loại A:** mỗi hộ thêm thông tin về số nhân khẩu
Số nước phải trả tiền = Khối nước sử dụng – 5*số nhân khẩu
Khối nước sử dụng = chỉ số mới – chỉ số cũ.
- **Hộ gia đình loại B:** mỗi hộ gia đình sẽ được miễn giảm 20 khối nước sử dụng.

- **Hộ gia đình loại C:** mỗi hộ gia đình có thêm thông tin về số nhân khẩu phục vụ công tác của chung cư. Cứ mỗi nhân khẩu phục vụ công tác của chung cư được giảm 10 khối nước sử dụng.

3. Bài tập về nhà


Bài 3. Mỗi môn học đều có các thông tin là mã môn, tên môn, số tín chỉ và khoa phụ trách, điểm. Hãy xây dựng chương trình tính học phí, tính điểm tích lũy cho của một sinh viên trong một học kỳ biết rằng mỗi sinh viên trong một học kỳ có thể học rất nhiều môn khác nhau. Thông tin của các môn học này được lưu trong file xml có cấu trúc như sau:

```
<Monhocs>
  <MH>
    <Loai>...<Loai>
    <Ma>...</Ma>
    <Ten>...</Ten>
    <SoTc>...</SoTc>
  </MH>
  ...
</Monhocs>
```

Các môn học trong trường được chia làm 3 loại như sau:

- **Môn lý thuyết:** Môn lý thuyết là môn có 3 cột điểm, điểm tiểu luận, điểm giữa kỳ và điểm cuối kỳ với hệ số là 0.2, 0.3 và 0.5. Học phí của môn lý thuyết sẽ được tính với đơn giá là 250000/ 1 tín chỉ.
- **Môn thực hành:** có 4 cột điểm kiểm tra và điểm trung bình sẽ bằng trung bình cộng của bốn bài kiểm tra. Môn thực hành cũng được tính với đơn giá 350000/ 1 tín chỉ cộng với 100000 tiền cơ sở vật chất.
- **Môn đồ án:** sẽ có điểm của GVHD và GVPB. Điểm của môn học là trung bình cộng của điểm GVHD và GVPB với điểm của GVHD tính hệ số 2. Mỗi môn đồ án đều có học phí là 2000000.

--HẾT--

Trường: ĐH CNTP TP.HCM Khoa: Công nghệ Thông tin Bộ môn: Công nghệ phần mềm MH: TH Lập trình hướng đối tượng	BÀI 8. INTERFACE	
---	-------------------------	--

A. MỤC TIÊU:

- Phân biệt giữa interface và lớp trừu tượng
- Xây dựng được interface cho bài toán cụ thể
- Xây dựng lớp kế thừa từ một hoặc nhiều interface

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

1. Tóm tắt lý thuyết

a. Cú pháp xây dựng interface trong C#

```
public interface <Ten_interface>
{
    //noi dung interface
}
```

- Interface **không được** khai báo thành phần dữ liệu.
- Tất cả các phương thức của Interface là trừu tượng và public (nhưng **không được** khai báo).

b. Cú pháp xây dựng lớp kế thừa từ interface

```
public class <TenLopCon>:<LopCha>, interfce1, interface2,..
{
    //cai dat lop
}
```

- Một lớp chỉ được kế thừa từ một lớp cha.
- Một lớp có thể kế thừa từ nhiều interface.

- Khi lớp thực thi Interface bắt buộc phải override tất cả các phương thức khai báo trong Interface (nhưng **không ghi** từ khóa override trong khai báo prototype của phương thức).
- Interface có thể kế thừa từ Interface khác (đây gọi là mở rộng Interface hay kết hợp Interface).

```
public interface <Ten_interface>:<interface1>,<interface2>,...
{
    //khai bao interface
}
```

2. Bài tập thực hành trên lớp

Bài tập có hướng dẫn

Bài 1. Trong một ứng dụng có quản lý bất động sản của công ty ABC có 4 đối tượng là: đất trống, nhà ở, biệt thự và khách sạn. Biết rằng tất cả các đối tượng này đều có các thông tin sau: mã số, chiều dài, chiều rộng và phương thức tính giá trị. Biết rằng giá trị được tính như sau:

- Đất trống: giá bán = diện tích * 10000
- Nhà ở có thêm thông tin về số lầu: giá bán = diện tích * 10000 + số lầu * 100000
- Khách sạn có thêm thông tin về số sao: giá bán = 100000 + số sao * 50000
- Biệt thự: giá bán = diện tích * 400000

Tuy nhiên trong 4 loại bất động sản trên thì có 2 loại sản phẩm khi mua khách hàng phải đóng thêm phí kinh doanh là biệt thự và khách sạn. Biết rằng phí kinh doanh sẽ được tính như sau:

- Biệt thự: Diện tích * 1000
- Khách sạn: Chiều rộng * 5000

Xây dựng ứng dụng quản lý bất động sản. Biết rằng danh sách bất động sản được lưu trữ trong file XML có cấu trúc như sau:

<BDSs>

<BDS >

<Loai>...</Loai>

```

        <Ma>...</Ma>
        <Dai>...</Dai>
        <Rong>...</Rong>
        ..
    </BDS>
    ...
</BDSs>

```

Hãy tính tổng giá trị của tất cả các bất động sản đang có và phí kinh doanh phải đóng của các bất động sản trên.

Hướng dẫn

Như trong đề bài đã yêu cầu, chúng ta cần xây dựng 4 lớp thể hiện cho các loại bất động sản khác nhau là đất trống, biệt thự, nhà ở và khách sạn. Cả 4 lớp này đều có một số các đặc điểm chung nên ta sẽ xây dựng một lớp cha cho cả 4 lớp này (giả sử gọi là **lớp BatDongSan**)

```

abstract public class BatDongSan
{
    protected double chieurong;
    protected double chieudai;
    string maso;
    abstract public double GiaBan();
    public double dientich()
    {
        return chieudai * chieurong;
    }
}

```

Cài đặt lớp đất trống

```

public class DatTrong : BatDongSan
{
    public override double GiaBan()
    {
        return dientich() * 10000;
    }
}

```

Cài đặt lớp **Nhà ở**

```
public class NhaO : BatDongSan
{
    int solau;
    public override double GiaBan()
    {
        return dientich() * 10000 + solau * 100000;
    }
}
```

Còn 2 lớp là Biệt thự và khách sạn cũng được xây dựng bằng cách kế thừa từ lớp BatDongSan nhưng vì 2 lớp này phải tính được phí kinh doanh nên ta cần phải xây dựng một interface. Trong trường hợp này ta không thể xây dựng thêm một lớp mới vì nếu xây dựng thêm một lớp mới thì dẫn tới lớp khách sạn và biệt thự sẽ kế thừa từ 2 lớp. Điều này không cho phép trong C#, vì C# chỉ cho phép đơn kế thừa.

```
public interface IPhi
{
    double PhiKinhDoanh();
}
```

Cài đặt lớp **khách sạn**

```
public class KhachSan : BatDongSan, IPhi
{
    int sosao;
    public override double GiaBan()
    {
        return 100000 + sosao * 50000;
    }

    public double PhiKinhDoanh()
    {
        return chieurong * 5000;
    }
}
```


Cài đặt lớp biệt thự

```
public class BietThu:BatDongSan,IPhi
{
    public override double GiaBan()
    {
        return dientich() * 400000;
    }
    public double PhiKinhDoanh()
    {
        return dientich() * 1000;
    }
}
```

Để có thể tính được tổng tiền của các bất động sản mà công ty đang có ta sẽ xây dựng lớp danh sách bất động sản để quản lý toàn bộ bất động sản mà công ty ABC đang có.

```
public class DSBatDongSan
{
    List<BatDongSan> ds = new List<BatDongSan>();
    public void nhap(string file)
    {
        XmlDocument read = new XmlDocument();
        read.Load(file);
        XmlNodeList nodelist = read.SelectNodes("/BDSs/BDS");
        foreach (XmlNode node in nodelist)
        {
            BatDongSan bds;
            int loai=Convert.ToInt32(node["Loai"].ToString());
            string ma = node["Ma"].InnerText;
            double dai = Convert.ToDouble(node["Dai"].InnerText);
            double rong = Convert.ToDouble(node["Rong"].InnerText);
            if (loai == 1){
                int so = Convert.ToInt32(node["So"].InnerText);
                bds = new NhaO(ma, dai, rong,so);
            }
            else if (loai == 2)
                bds = new DatTrong(ma, dai, rong);
            else if (loai == 3)
                bds = new BietThu(ma, dai,rong);
            else {
                int so = Convert.ToInt32(node["So"].InnerText);
                bds = new KhachSan(ma, dai, rong, so);
            }
            ds.Add(bds);
        }
    }
}
```

Phương thức xuất và tính tổng giá trị, tổng số phí kinh doanh:

```
public void xuat()
{
    foreach (BatDongSan bds in ds)
        bds.xuat();
}

public double TongGiaTri()
{
    double tong = 0;
    foreach (BatDongSan bds in ds)
        tong = tong + bds.GiaBan();
    return tong;
}

public double TongThue()
{
    double tong = 0;
    foreach (BatDongSan bds in ds)
        if (bds is IPhi)
        {
            IPhi t = (IPhi)bds;
            tong = tong + t.PhiKinhDoanh();
        }
    return tong;
}
```

Bài tập sinh viên tự làm

Bài 2. Trong công tác quản lý của tổng công ty T có 2 công ty con là công ty ABC và công ty BCD.

Trong công ty ABC nhân viên được chia làm 3 loại khác nhau trong quá trình vận hành sản xuất là nhân viên sản xuất, nhân viên kinh doanh và cán bộ điều hành. Mỗi nhân viên trong công ty đều có các thông tin chung như: mã nhân viên, tên nhân viên, năm sinh, giới tính, hệ số lương và năm vào làm. Để nâng cao tính tích cực cho nhân viên, công ty tổ chức đánh giá xếp loại thi đua cho nhân viên hàng tháng (loại A, B, C, D). Xếp loại này sẽ quyết định thu nhập của mỗi nhân viên như sau:

- Nếu xếp loại A: nhân viên được hưởng 100% Lương + phụ cấp thâm niên
- Nếu xếp loại B: nhân viên được hưởng 75% Lương + phụ cấp thâm niên
- Nếu xếp loại C: nhân viên được hưởng 50% Lương + phụ cấp thâm niên
- Nếu xếp loại D: nhân viên được hưởng 0% mức lương + phụ cấp thâm niên

Biết rằng phụ cấp thâm niên chỉ áp dụng cho các nhân viên có thời gian làm việc từ 5 năm trở lên và được tính như sau: phụ cấp thâm niên = số năm làm việc * lương cơ bản/100. Lương cơ bản dùng chung cho tất cả nhân viên là 1210.

Do tính chất công việc nên mỗi nhân viên lại có những đặc điểm, cách xếp loại, cách tính lương và phụ cấp khác nhau như sau:

- Nhân viên sản xuất: do làm việc trong môi trường khắc nghiệt nên nhân viên sản xuất đều được hưởng một hệ số phụ cấp nặng nhọc. Hệ số này là chung cho tất cả các nhân viên sản xuất của công ty và đang được công ty quy định là 10%. Mỗi nhân viên sẽ có thông tin về số ngày nghỉ trong một tháng và xếp loại thi đua được tính như sau:

- Nếu số ngày nghỉ ≤ 1 thì xếp loại A, số ngày nghỉ ≤ 3 thì xếp loại B, số ngày nghỉ ≤ 5 thì xếp loại C còn lại là xếp loại D.
- Lương của nhân viên sản xuất được tính theo công thức như sau:

Lương = hệ số lương * lương cơ bản * (1 + phụ cấp nặng nhọc)

- Nhân viên kinh doanh: công ty không bắt buộc nhân viên kinh doanh phải làm việc theo ngày giống như nhân viên sản xuất mà chỉ quan tâm tới năng lực bán hàng của từng người. Mỗi nhân viên kinh doanh có một doanh số bán hàng tối thiểu khác nhau và một thông tin về doanh số bán hàng theo tháng.

- Nếu nhân viên hoàn thành doanh số tối thiểu thì xếp loại thi đua là B, nếu không hoàn thành thì xếp loại C. Nếu doanh số nhỏ hơn 50% doanh số tối thiểu thì xếp loại D. Ngược lại nếu có danh số cao hơn 2.0 lần doanh số tối thiểu thì xếp loại A.
- Lương của nhân viên kinh doanh cũng được xác định như sau:

$$\text{Lương} = \text{hệ số lương} * \text{lương cơ bản} + \text{hoa hồng.}$$

Trong đó hoa hồng là 15% của doanh thu (doanh thu = doanh số vượt mức * 500).

– Cán bộ quản lý: có trách nhiệm quản lý toàn bộ hoạt động sản xuất của công ty. Cán bộ quản lý có thêm thông tin về chức vụ và hệ số chức vụ.

- Xếp loại thi đua của cán bộ quản lý luôn luôn là A.
- Lương của cán bộ quản lý được xác định như sau:

$$\text{Lương} = \text{hệ số lương} * \text{lương cơ bản} + \text{phụ cấp chức vụ.}$$

Trong đó phụ cấp chức vụ = hệ số phụ cấp chức vụ * 1100.

Để có thể tham gia đánh giá của tổng công ty, công ty ABC coi các nhân viên có xếp loại thi đua loại A thì danh hiệu thi đua là chiến sĩ thi đua, loại B là lao động tiên tiến.

Đối với công ty BCD, nhân viên chỉ cần các thông tin về mã nhân viên, tên nhân viên. Công ty BCD không có xếp loại thi đua hàng tháng và lương chỉ tính bằng 15% của tổng số tiền mà nhân viên đó tạo ra trong một tháng mà thôi. Công ty BCD sẽ xem các nhân viên có lương trên 10000 là lao động tiên tiến và trên 20000 là chiến sĩ thi đua.

Tổng công ty T có tổ chức đánh giá các nhân viên trong các công ty thành viên. Muốn một nhân viên được đánh giá là “Có năng lực” hay “Năng lực tốt” thì tổng công ty chỉ quan tâm tới là mỗi nhân viên ở trong các công ty con có phải là chiến sĩ thi đua (nghĩa là có ‘năng lực tốt’) hay lao động tiên tiến (nghĩa là “Có năng lực”) hay không mà thôi, không quan tâm tới công ty con đánh giá vấn đề đó dựa vào tiêu chí nào.

Giả sử bạn đang làm trong tổng công ty. Hãy tạo danh sách chứa nhân viên của 2 công ty và thực hiện các yêu cầu sau:

- Đọc dữ liệu từ file XML (file này SV tự thiết kế cấu trúc chứa dữ liệu cho phù hợp các truy vấn của bài).
- Đếm số nhân viên của mỗi công ty con.
- Cho biết tổng công ty T có bao nhiêu nhân viên có “Năng lực tốt”.
- Xuất thông tin các nhân viên chưa được xét thi đua ở công ty BCD.
- Xuất thông tin các nhân viên lao động tiên tiến của công ty ABC.

3. Bài tập về nhà

Bài 3. Xây dựng các lớp phân số, số phức (theo yêu cầu các bài tập tuần 3). Mỗi lớp đều có các hàm cộng, trừ, nhân, chia và so sánh giữa 2 đối tượng số.


Tạo 2 danh sách chứa mỗi loại số trên. Biết rằng mỗi danh sách đều có các thao tác: sắp xếp tăng/giảm, tìm max, min và đếm tổng số phần tử.

Bài 4. Tạo chương trình quản lý các quân cờ trong bàn cờ. Biết rằng quân cờ nào cũng có các thao tác chung: cách đi, cách thắng; cùng các thông số: tên, hình dáng, vị trí (bàn cờ là ma trận 8x8, vị trí quân cờ tại $(i, j) \rightarrow$ dòng i , cột j với $0 \leq i, j \leq 7$).

Viết hàm lưu thông tin của quân mã và xe trong bàn cờ vua. Biết rằng:

- Quân xe đi thẳng, còn quân mã đi xéo 2 ô cờ.
- Quân xe thắng các con cờ nằm cùng dòng (hoặc cột); Quân mã thắng các quân cờ cách mã 2 cột, 1 dòng hoặc 2 dòng 1 cột.

--HẾT--

Trường: ĐH CNTP TP.HCM Khoa: Công nghệ Thông tin Bộ môn: Công nghệ phần mềm MH: TH Lập trình hướng đối tượng	BÀI 9. BÀI TẬP TỔNG HỢP	
---	--------------------------------	--

A. MỤC TIÊU:

- Phân tích được mối quan hệ giữa các lớp
- Xác định được các thành phần chung để xây dựng lớp cha, lớp trừu tượng và interface.
- Xác định được các phương thức trừu tượng.

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

Bài tập trên lớp: Quản lý dịch vụ taxi của Uber

Công ty Uber Taxi kinh doanh dịch vụ taxi dưới 3 hình thức xe gắn máy, ô tô và ô tô tải (taxi tải dùng để vận chuyển hàng). Công ty quản lý các tài xế và cuộc gọi của khách thông qua tài khoản đăng ký trên ứng dụng online của di động. Mỗi tài khoản đăng ký trên hệ thống đều có các thông tin: username, password, số điện thoại, trạng thái (giả sử có 2 giá trị: online, offline). Đồng thời mỗi tài khoản có các xử lý cơ bản: đăng nhập/ đăng xuất (thay đổi trạng thái), lấy địa chỉ của tọa độ trên Google Map (giả sử hàm trả về giá trị số nguyên ngẫu nhiên < 1000).

Tài khoản được chia làm 2 loại: tài khoản tài xế và tài khoản của khách hàng.

- Tài khoản của khách: có thêm thông tin loại khách (giá trị VIP hay thường), cùng phương thức xử lý Gọi xe: chuyển trạng thái online, gửi thông tin cần loại xe nào và địa chỉ đón khách.
- Tài khoản dành cho tài xế cần có thêm các thông tin: mã số, tên tài xế, hình thức (gắn máy, ô tô hay ô tô tải). Mỗi tài khoản tài xế đều có các thao tác:

- + Phát tín hiệu nhận cuộc gọi của khách: xuất ra vị trí của khách gọi xe.
- + Tính tiền chuyển đi: mỗi loại hình xe có cách tính khác nhau.
- + Trả phí lại cho công ty: mỗi loại hình xe có cách tính khác nhau.

Mỗi chuyến xe của khách chứa các thông tin: tài xế, khách hàng, địa chỉ đón, địa chỉ đến, khoảng cách (tính theo km), thời gian bắt đầu đi (giờ đón), ngày đi và hàm tính thành tiền cho chuyến xe. Hàm tính tiền cho mỗi chuyến như sau:

- + Đối với xe gắn máy: 2km đầu tiên giá 8000, mỗi km tiếp theo giá 5000. Tuy nhiên nếu khách gọi đi trong khoảng thời gian từ 22h00 → 5h00 thì phụ thu thêm 3000 cho mỗi km đường đi.
- + Đối với ô tô: xe có thêm thông tin số chỗ ngồi (có 2 giá trị 4 chỗ và 7 chỗ).
 - Nếu xe 4 chỗ: 2km đầu giá 15000, 5km tiếp theo giá 12000 và các km còn lại giá 8000 và mỗi km đường đi phụ thu thêm 500.
 - Nếu xe 7 chỗ: 2km đầu giá 17000, 5km tiếp theo giá 15000 và các km còn lại giá 10000 và mỗi km đường đi phụ thu thêm 500.
- + Đối với ô tô tải: xe có thêm thông tin trọng tải.
 - Nếu trọng tải ≤ 3 tấn: 5km đầu giá 60000, 5km tiếp theo giá 50000, các km còn lại giá 30000 và mỗi km đường đi phụ thu 5000.
 - Ngược lại: 5km đầu giá 70000, 5km tiếp theo giá 60000, các km còn lại giá 40000 và mỗi km đường đi phụ thu 5000.

Lưu ý nếu khách hàng loại VIP sẽ giảm 10% tổng thành tiền nếu tiền cước > 200000.

Sau mỗi chuyến đi, tài xế cần trả phí lại cho công ty theo qui định: Xe máy trả 5%, ô tô trả 10% và ô tô tải trả 15% tiền chuyến xe. Tuy nhiên, công ty ưu tiên tính thưởng cho loại hình taxi ô tô và ô tô tải như sau:

- + Ô tô: nếu doanh thu trong 1 ngày > 2000000 thì tài xế được thưởng 200000.
- + Ô tô tải: nếu doanh thu trong 1 ngày > 3000000 thì tài xế được thưởng 300000.

Công ty lưu thông tin các tài khoản khách hàng, tài xế, các loại xe và các chuyến xe theo danh sách từng loại. Hãy phân tích và thiết kế lớp cho chương trình để thực hiện các yêu cầu:

1. Tạo cấu trúc lớp cho chương trình quản lý xe.
2. Tạo dữ liệu XML tương ứng cho từng danh sách và đọc vào chương trình.
3. Xuất thông tin tất cả các danh sách: khách hàng, tài xế, chuyến xe.
4. Nếu khách hàng nào đi từ chuyến thứ 10 trở đi sẽ được chuyển thành khách VIP.

Bài tập về nhà: Phát triển tiếp bài tập trên lớp:

1. Đếm số xe mỗi loại.
2. Đếm số tài khoản khách hiện có trong danh sách.
3. Xuất thông tin các tài xế lái xe gắn máy.
4. Tính tổng doanh thu từng loại xe trong một ngày (tùy chọn).
5. Xuất thông tin chuyến xe có tiền cược nhiều nhất đối với xe máy.
6. Xuất thông tin chuyến xe có tiền cược nhiều nhất đối với xe ô tô.
7. Xuất thông tin chuyến xe có tiền cược nhiều nhất đối với xe ô tô tải.
8. Sắp xếp các chuyến xe tăng dần theo tiền cược và mã tài xế.
9. Xuất thông tin khách hàng VIP.
10. Tìm khách hàng có số chuyến đi nhiều nhất.
11. Tìm tài xế chạy cho doanh thu cao nhất.
12. Xuất thông tin các tài xế ô tô được thưởng.
13. Xuất thông tin các tài xế ô tô tải được thưởng.

– HẾT –

Trường: ĐH CNTP TP.HCM Khoa: Công nghệ Thông tin Bộ môn: Công nghệ phần mềm MH: TH Lập trình hướng đối tượng	BÀI 10. ÔN TẬP	
---	-----------------------	--

A. MỤC TIÊU:

- Phân tích được mối quan hệ giữa các lớp.
- Xác định được các thành phần chung để xây dựng lớp cha, lớp abstract và interface.
- Xác định được các phương thức trừu tượng.

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

Một đại lý mua bán xăng dầu quản lý hàng và hóa đơn bán hàng như sau:

Mỗi mặt hàng xăng dầu lưu thông tin: mã hàng, tên hàng, đơn giá (giá/1 lít).

Hóa đơn bán hàng cho khách có 3 loại: Hóa đơn khách VIP, hóa đơn khách thân thiết và hóa đơn dành cho khách vắng lại. Hóa đơn nào cũng chứa thông tin: mã số, họ tên khách (nếu là khách vắng lại thì họ tên không cần ghi), ngày lập, mặt hàng, số lượng và hàm tính thành tiền. Biết rằng *tổng thành tiền = thành tiền - tiền khuyến mãi*, với *thành tiền = số lượng * đơn giá* và cách tính tiền khuyến mãi dành cho mỗi loại khách khác nhau:

- **Hóa đơn khách VIP:**
 - Nếu số lượng > 50 lít thì giảm 5% giá bán cho mỗi lít.
 - Ngược lại nếu số lượng ≤ 50 nhưng thành tiền ≥ 600.000 thì giảm 4% trên thành tiền

- Các trường hợp còn lại, ngoại trừ số lượng < 10 lít, thì giảm 1% trên thành tiền hóa đơn.
- **Hóa đơn khách thân thiết:**
 - Nếu số lượng > 60 lít thì giảm 4% giá bán cho mỗi lít.
 - Ngược lại nếu số lượng ≤ 50 nhưng thành tiền ≥ 800.000 thì giảm 3% trên thành tiền
 - Các trường hợp còn lại thì không khuyến mãi.
- **Hóa đơn khách vắng lại:**
 - Nếu thành tiền > 1.000.000 VNĐ, thì giảm 3% tổng thành tiền hóa đơn, ngược lại tiền khuyến mãi không có.
 - Ngoài ra trong hóa đơn dành cho khách VIP và thân thiết còn có thêm các hàm xử lý riêng sau: thanh toán tạm ứng và tính lãi suất trả chậm.
- **Hóa đơn khách VIP:**
 - Thanh toán tạm ứng: 40% của tổng thành tiền.
 - Lãi xuất trả chậm: 2% * (tổng thành tiền – thanh toán tạm ứng).
- **Hóa đơn khách thân thiết:**
 - Thanh toán tạm ứng: 60% của tổng thành tiền.
 - Lãi xuất trả chậm: 3% * (tổng thành tiền – thanh toán tạm ứng).

Hướng dẫn cấu trúc file Hoadon.xml:

```
<DSHD>
  <HoaDon>
    <Loai>1</Loai> //Hóa đơn khách VIP
    <MS>...</MS>
    <Khach>...</Khach>
    <NgàyLap>23/5/2016</NgàyLap>
    <Hang>
```

```

        <MH>A92</MH>
        <TenHang>Xang 92</TenHang>
        <Gia>15500</Gia>
    </Hang>
    <Soluong>20</Soluong>
</HoaDon>
<HoaDon>
    <Loai>2</Loai> //Hóa đơn khách thường
    <MS>...</MS>
    <Khach>...</Khach>
    <NgayLap>3/5/2016</NgayLap>
    <Hang>
        <MH>A95</MH>
        <TenHang>Xang 95</TenHang>
        <Gia>16500</Gia>
    </Hang>
    <Soluong>50</Soluong>
</HoaDon>
...
</DSHD>

```

Yêu cầu:

1. Lập danh sách hóa đơn lưu thông tin tất cả các hóa đơn của cửa hàng.
2. Tạo danh sách lưu thông tin tất cả các hóa đơn của cửa hàng. Đọc dữ liệu từ file xml và xuất ra tất cả các thông tin trong danh sách.
3. Xuất danh sách các hóa đơn của khách vắng lại.
4. Tính tổng thành tiền các hóa đơn trong danh sách.
5. Xuất hóa đơn có tổng thành tiền cao nhất.
6. Đếm số hóa đơn của khách VIP và khách thân thiết.
7. Tính tổng thanh toán tạm ứng của khách Nguyễn Long.

8. Sắp xếp hóa đơn tăng dần theo tổng thành tiền, nếu thành tiền bằng thì sắp giảm theo mã số hóa đơn.

--HẾT--