

BÀI THỰC HÀNH SỐ 2

I. Mục tiêu.

Hiểu và nắm vững các đặc điểm lập trình hướng đối tượng trong lập trình C#, và các kiểu dữ liệu mở rộng trong C#, các Collection.

II. Thời gian

Thực hành: 5 tiết

III. Hướng dẫn chung

Câu 1: Tạo lớp **Tài khoản** gồm các thuộc tính: Số tài khoản (kiểu **long**), Tên tài khoản (kiểu **string**), số tiền trong tài khoản (kiểu **double**); cùng với các phương thức:

- Constructor: có 2 constructor. (Mặc định và đầy đủ tham số với số tiền mặc định là 10 USD).
- Có thể sử dụng đặc tính (Property) để lưu trữ và truy xuất giá trị.
- Phương thức in ra toàn bộ thông tin tài khoản (ToString), yêu cầu định dạng tiền tệ cho số tiền của tài khoản theo USD.
- Phương thức nạp tiền vào tài khoản (Bảng tiền hiện tại cộng số tiền mới nạp).
- Phương thức rút tiền: Số tiền hiện có – (số tiền muốn rút + phí rút). Phí rút bằng 1% tiền rút.

Lưu ý: mỗi thao tác nạp tiền và rút tiền kiểm tra số tiền nhập vào có hợp lệ không (>0) hoặc rút tiền thì số tiền rút phải nhỏ hơn số tiền hiện có; nếu không thỏa phải nhập lại.

Hướng dẫn: Khai báo lớp **Tài khoản** theo mô hình sau:

TaiKhoan
+soTK: long +tenTK: string +soTien: double
+TaiKhoan() +TaiKhoan(long soTK, string tenTK, double soTien)

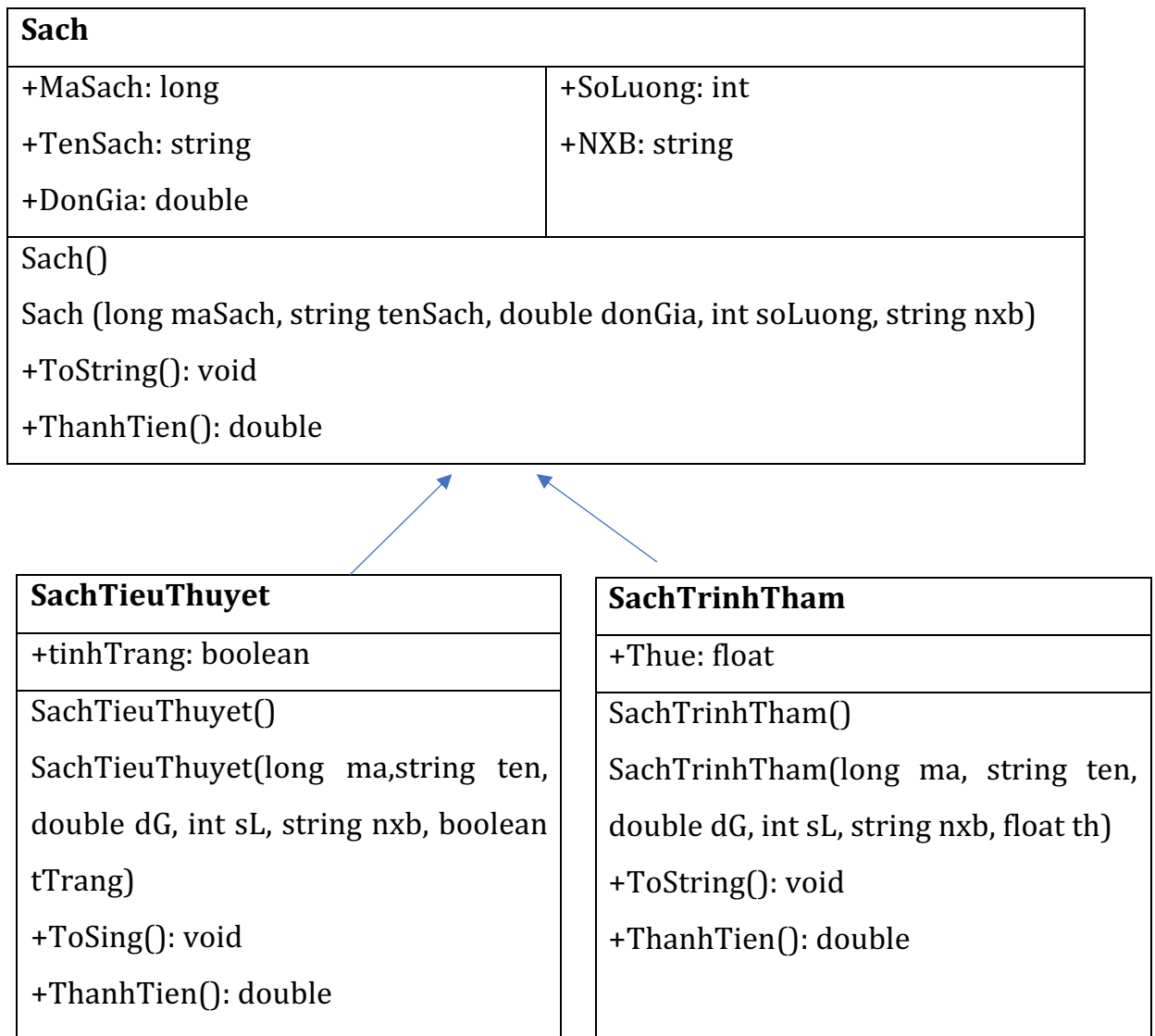
+InTaiKhoan(): void +NapTien(double st): boolean +RutTien: boolean
--

Câu 2: Cửa hàng sách có bán 2 loại sách, thông tin về mỗi loại sách:

- Sách tiểu thuyết: Mã sách, tên sách, đơn giá, số lượng, nhà xuất bản, tình trạng (mới, cũ).
 - Nếu sách mới: thành tiền = số lượng * đơn giá.
 - Nếu sách cũ: thành tiền = số lượng * đơn giá * 20%
- Sách trinh thám: Mã sách, tên sách, đơn giá, số lượng, nhà xuất bản, thuế.
 - Thành tiền = số lượng * đơn giá + thuế

Xây dựng các lớp với chức năng thừa kế và các phương thức phù hợp cho từng lớp.

Hướng dẫn: Khai báo lớp với chức năng thừa kế theo mô hình sau:



Các thuộc tính có thể sử dụng đặc tính (property) để cài đặt.

Cài đặt các hàm khởi tạo sử dụng hàm khởi tạo ở lớp cha.

Ghi đè (override) phương thức ToString() và ThanhTien() trên 2 lớp SachTieuThuyet và SachTrinhTham theo yêu cầu của bài xuất đầy đủ thông tin của sách và để tính thành tiền của từng loại sách.

Câu 3: Sử dụng phương thức Generic cho phép tính trung bình các giá trị trong mảng.

```
public static double GetAverage<T>(List<T> numList)
{
    double avg = 0.0;
    foreach (var num in numList){
        avg += double.Parse(num.ToString());
    }
    return avg / numList.Count;
}

static void Main(string[] args)
{
    Console.OutputEncoding = Encoding.Unicode;

    List<int> listInt = new List<int> { 5, 8, 59, 9, 11, 24, 40 };
    List<float> listFloat = new List<float> { 7.9f, 13.12f, 50.56f, 109.5f, 80.4f };

    Console.WriteLine($"Trung bình của list int: {GetAverage(listInt)}");
    Console.WriteLine($"Trung bình của list float: {GetAverage(listFloat)}");
}
```

Câu 4: Dùng Generic để viết lớp PersonManager để quản lý các danh sách Student và Lecturer.

Hướng dẫn:

- Tạo class Student có các thuộc tính id, name, age (theo kiểu đặc tính) và hàm toString.
- Tạo class Lecturer có các thuộc tính id, name, salary (theo kiểu đặc tính) và hàm toString.
- Tạo class Generic PersonManager như sau:

```
class Student
{
    4 references
    public string ID { get; set; }
    4 references
    public string Name { get; set; }
    4 references
    public int Age { get; set; }

    1 reference
    public override string ToString()
    {
        return $"Mã: {ID}\t Tên: {Name}\t Tuổi: {Age}";
    }
}
```

```

class Lecturer {
    4 references
    public string ID { get; set; }
    4 references
    public string Name { get; set; }
    4 references
    public int Salary { get; set; }

    1 reference
    public override string ToString()
    {
        return $"Mã: {ID}\t Tên: {Name}\t Lương: {Salary}";
    }
}

class PersonManager<T>
{
    private List<T> _listPerson = new List<T>();
    6 references
    public void Add(T person)
    {
        _listPerson.Add(person);
    }
    2 references
    public void Display()
    {
        foreach (T obj in _listPerson)
        {
            Console.WriteLine($"{obj}");
        }
    }
}

```

○ Ở main, tạo đối tượng thuộc lớp **PersonManager<Lecturer>** và **PersonManager<Student>** để thêm lecturer và student; sau đó xuất thông tin các lecturer và student ra.

```

static void Main(string[] args)
{
    Console.Title = "BTTH2";
    Console.OutputEncoding = Encoding.Unicode;
    //Input();
    PersonManager<Lecturer> lecturer = new PersonManager<Lecturer>();
    lecturer.Add(new Lecturer { ID = "1001", Name = "Ted", Salary = 200 });
    lecturer.Add(new Lecturer { ID = "1002", Name = "Robin", Salary = 500 });
    lecturer.Add(new Lecturer { ID = "1003", Name = "Batman", Salary = 1000 } );

    PersonManager<Student> student = new PersonManager<Student>();
    student.Add(new Student { ID = "2001", Name = "King Kong", Age = 10 });
    student.Add(new Student { ID = "2002", Name = "T-Rex", Age = 15 });
    student.Add(new Student { ID = "2003", Name = "Shark", Age = 16 });

    Console.WriteLine("Thông tin Lecture:");
    lecturer.Display();
    Console.WriteLine("Thông tin Student:");
    student.Display();
}

```

IV. Thực hành

Phần 1: Nhóm bài tập về xác định các lớp, thuộc tính và phương thức của từng lớp.

Mục tiêu: Các bài tập trong phần này giúp sinh viên có khả năng: thành thạo trong việc xác định và cài đặt lớp, thuộc tính và phương thức của lớp.

Câu 1: Viết chương trình tính diện tích, chu vi hình chữ nhật.

Hướng dẫn: cài đặt lớp **HìnhChữNhật** gồm có:

- Thuộc tính : chiều dài, chiều rộng.
- Dùng đặc tính (property) để cài đặt chiều dài, chiều rộng.
- Phương thức tính diện tích, chu vi.
- Phương thức toString gồm các thông tin dài, rộng, diện tích, chu vi.
- Xây dựng lớp chứa hàm main cho phần kiểm nghiệm. Dài rộng có thể nhập từ bàn phím.

Câu 2: Viết chương trình quản lý sinh viên đơn giản: Nhập, xuất thông tin sinh viên, tính điểm TB.

Hướng dẫn:

○ Cài đặt lớp **Sinh viên** như sau:

- Thuộc tính (private):
 - Mã sinh viên là số nguyên.
 - Họ tên: chuỗi ký tự.
 - Điểm LT, điểm TH : float
- Constructor:
 - Constructor mặc định (để khởi tạo đối tượng với các thông tin kiểu số là 0, kiểu chuỗi là chuỗi rỗng).
 - Constructor thứ hai nhận đầy đủ thông tin để khởi tạo giá trị cho tất cả các biến của lớp.
- Phương thức:
 - Dùng đặc tính (property) để cài đặt các thuộc tính private.
 - Tính điểm trung bình.
 - toString để diễn tả đối tượng ở dạng chuỗi.

○ Xây dựng class chứa hàm main: tạo các đối tượng sinh viên và đưa vào list (tùy số lượng), trong đó:

- Nhập thông tin cho đối tượng sinh viên trên (tùy ý).
- In danh sách sinh viên gồm 5 cột là Mã sinh viên, họ tên, điểm LT , điểm TH, điểm TB.

Câu 3: Thông tin của xe bao gồm: tên chủ xe, loại xe, trị giá xe, dung tích xylanh của xe. Dựa vào trị giá xe và dung tích xylanh của xe để tính mức thuế phải đóng sau khi mua xe như sau:

- Dung tích dưới 100cc, 1% trị giá xe.
- Dung tích từ 100 đến 175cc, 3% trị giá xe.
- Dung tích trên 175cc, 5% trị giá xe.

Hãy cài đặt lớp Xe với các thuộc tính và phương thức phù hợp yêu cầu.

Xây dựng các hàm để thực hiện các yêu cầu sau:

- Nhập danh sách xe vào một mảng, việc nhập kết thúc khi ta nhập tên chủ xe bằng rỗng.
- Xuất bảng kê khai tiền thuế phải đóng của các xe trong danh sách. Thông tin xuất bao gồm: tên chủ xe, loại xe, trị giá xe, dung tích xylanh của xe, thuế phải đóng.
- Nhập tên chủ xe và loại xe vào, xuất hiện thuế phải đóng của chủ xe.

Phần 2: Nhóm bài tập về kế thừa, lớp trừu tượng (Abstract Class) và giao diện (Interface Class) và sử dụng Collections.

Mục tiêu: Các bài tập trong phần này giúp sinh viên có khả năng: xác định được mối liên hệ của nhiều lớp qua các đặc tính của hướng đối tượng như thừa kế hoặc đa hình

Câu 1: Cài đặt các lớp trong chương trình quản lý nhân viên của một công ty. Các lớp lớp được mô tả như sau:

- Lớp **Person** là một lớp trừu tượng, có hai thuộc tính là tên (**name**) và tuổi (**age**). Để đóng gói dữ liệu các thuộc tính này có dạng **private**. Lớp có một phương thức trừu tượng là **show()**.
- Lớp **Employee** là lớp kế thừa từ lớp **Person**, có thêm thuộc tính là lương (**salary**), thuộc tính này có dạng **private**. Lớp này cài đặt lại phương thức **show()**. Hơn nữa, lớp **Employee** còn có thêm hai phương thức **addSalary()** và **addSalary(float)** để tính tăng lương cho nhân viên: một phương thức tăng lương theo tỉ lệ mặc định là **10% (không cần tham số)**, và một phương thức tăng theo giá trị cụ thể đưa vào (cần tham số).

Câu 2: Công ty vận tải A có các chuyến xe nội thành và ngoại thành. Thông tin của 2 loại chuyến xe:

- Chuyến nội thành: Mã số chuyến, họ tên tài xế, số xe, khối lượng hàng hóa, quãng đường đi, doanh thu.
- Chuyến ngoại thành: Mã số chuyến, họ tên tài xế, số xe, khối lượng hàng hóa, nơi đến, số ngày vận chuyển, doanh thu.

Thực hiện các yêu cầu sau:

- Xác định các lớp với chức năng thừa kế.
- Cài đặt các lớp đã thiết kế.
- Viết hàm nhập xuất danh sách các chuyến xe không quá 20 chuyến cho cả chuyến nội thành và ngoại thành.
- Tính tổng doanh thu của chuyến nội thành và ngoại thành.
- In ra thông tin chuyến xe có doanh thu cao nhất của từng loại chuyến.

Hướng dẫn: dùng phương thức `GetType` để xác định kiểu lớp của đối tượng và `typeof` để xác định lớp. VD: `if (obj.GetType() == typeof(Example Class))`

Câu 3: Dùng Generic để viết lớp với 2 phương thức tính max và min của một mảng.

Câu 4: Viết chương trình tra cứu danh bạ điện thoại lưu trữ các thông tin sau: Số điện thoại, địa chỉ đăng ký. Sử dụng cấu trúc Collection cho phù hợp để lưu trữ thông tin danh bạ và thực hiện các công việc sau:

- Tra cứu thông tin theo số điện thoại.
- Tra cứu theo số điện thoại theo địa chỉ đăng ký. Lưu ý: một địa chỉ đăng ký có thể có nhiều số điện thoại.

V. Bài tập thêm

Câu 1: Công ty điện lực ABC có 3 đối tượng khách hàng sử dụng điện bao gồm sinh hoạt, kinh doanh, sản xuất.

- Sinh hoạt gồm: Mã khách hàng, Tên khách hàng, ngày hóa đơn, số lượng điện, đơn giá, định mức, thành tiền:
 - Nếu số lượng điện \leq định mức thì thành tiền = số lượng điện * đơn giá.
 - Ngược lại: thành tiền = số lượng điện * đơn giá * định mức + số lượng điện vượt định mức * đơn giá * 2.

- Kinh doanh gồm: Mã khách hàng, Tên khách hàng, ngày hóa đơn, số lượng điện, đơn giá, thành tiền:
 - Thành tiền = số lượng điện * đơn giá.
 - Khi số lượng điện vượt 400 thì thành tiền = số lượng điện * đơn giá * 5%.
- Loại Sản xuất: : Mã khách hàng, Tên khách hàng, ngày hóa đơn, số lượng điện, đơn giá, loại điện (2 pha hoặc 3 pha), thành tiền:
 - Đối với loại điện 2 pha: thành tiền = số lượng điện * đơn giá và giảm 2% tiền khi số lượng điện vượt 200.
 - Đối với loại điện 3 pha: thành tiền = số lượng điện * đơn giá và giảm 3% tiền khi số lượng điện vượt 150.

Hãy thực hiện các yêu cầu sau:

- Xác định các lớp với chức năng thừa kế và cài đặt phương thức phù hợp với chức năng của từng lớp:
 - Constructor không tham số và đầy đủ tham số.
 - Phương thức thành tiền của từng loại đối tượng sử dụng điện
 - Phương thức in ra toàn bộ thông tin của các loại khách hàng.
- Viết các hàm sau:
 - Nhập khách hàng sử dụng điện với loại điện sử dụng vào một mảng, việc nhập kết thúc khi ta nhập Mã khách hàng bằng rỗng; mã khách hàng và tên khách hàng giống nhau thì phải cùng loại điện sử dụng, nhưng khác nhau ngày hóa đơn.
 - Xuất hóa đơn tiền điện của các khách hàng với tháng và năm hóa đơn nhập từ bàn phím. Thông tin gồm Mã khách hàng, tên khách hàng, số lượng điện và thành tiền.

Câu 2: Hàng hóa trong siêu thị ABC gồm có hàng điện máy, hàng thực phẩm và hàng gia dụng. Mỗi loại hàng gồm có mã hàng, tên hàng, số lượng tồn, đơn giá.

- Hàng điện máy thì có thêm các thông tin thương hiệu, loại máy (VD: laptop, điện thoại, máy lạnh...), thời gian bảo hành (tính theo tháng).
- Hàng thực phẩm thì cần có thông tin ngày sản xuất, ngày hết hạn (ngày hết hạn phải sau hoặc bằng ngày sản xuất), nhà cung cấp.
- Hàng gia dụng thì cần biết thông tin nhà sản xuất, ngày nhập vào siêu thị, loại (VD: ly, chén, nồi...)

- Ngoài ra, cần quan tâm đến số lượng tồn kho để nhập hàng về:
 - Hàng điện máy, nếu số lượng tồn < 3 thì sẽ nhập hàng.
 - Hàng thực phẩm, nếu khi bị hết hạn thì sản phẩm sẽ cập nhật số lượng hàng mới và ngày hết hạn mới.
 - Hàng gia dụng, nếu số lượng tồn < 5 thì sẽ nhập hàng.

Hãy thực hiện các yêu cầu sau:

- Xác định các lớp với chức năng thừa kế và cài đặt phương thức phù hợp với chức năng của từng lớp:
 - Constructor không tham số và đầy đủ tham số.
 - Phương thức in ra toàn bộ thông tin của các loại hàng.
 - Phương thức thành tiền của từng loại hàng: số lượng * đơn giá và thuế VAT. Biết rằng VAT của hàng điện máy và hàng gia dụng là 10%, VAT của hàng thực phẩm là 5%. Mỗi lần tính tiền thì sẽ cập nhật lại số lượng tồn của hàng hóa.
- Viết các hàm sau:
 - Nhập hàng hóa vào một mảng, việc nhập kết thúc khi ta nhập Mã hàng bằng rỗng và thêm thành công hàng hóa vào mảng khi mã hàng hóa không trùng. (Có thể sử dụng SortedList).
 - Xuất danh sách hàng hóa có số lượng tồn nhỏ hơn theo quy định để nhập hàng. Thông tin gồm Mã hàng, tên hàng, số lượng tồn (với hàng thực phẩm thì hiển thị ngày hết hạn)

Câu 3: Tiếp theo phần câu 2 thêm các chức năng sau:

- Tạo ra danh sách mua hàng của một khách hàng (dựa trên danh sách hàng hóa đã làm ở trên), nhập mã hàng hóa, và số lượng từng hàng mà khách mua. Việc nhập mua hàng kết thúc khi nhập một chuỗi khác với mã hàng đã có.
- In ra màn hình hóa đơn mua hàng của khách hàng với các thông tin: Mã hàng, tên hàng, số lượng, đơn giá, thành tiền từng hàng và tổng thành tiền (với tổng tiền của hóa đơn hơn 1 triệu thì giảm 10% cho đơn hàng, lớn hơn 2 triệu thì giảm 15%).