

BÀI TẬP

CHƯƠNG 15 - KẾ THỪA, ĐA HÌNH VÀ HÀM ẢO

Bài 1. Employee (bắt buộc)

- Xây dựng class **Employee** có các thuộc tính: tên, mã số, ngày (bắt đầu) thuê . Viết một hoặc nhiều hàm khởi tạo cho class này, viết hàm set và get cho các thuộc tính
- Xây dựng class **ProductionWorker** dẫn xuất từ lớp **Employee** có thêm các thuộc tính: ca làm việc (integer) , mức lương theo giờ. Ca làm việc bằng 1 nếu là ca ngày, bằng 2 nếu là ca đêm. Viết một hoặc nhiều hàm khởi tạo cho class này. Viết các hàm set/get cho các thuộc tính
- Sử dụng một đối tượng của class **ProductionWorker** trong một ví dụ minh họa, Ví dụ: tính lương của công nhân biết trong tháng người đó làm 26 ngày, mức lương theo giờ là 5\$ và ca đêm thì được tính thêm 50% so với mức trên.

Bài 2. ShiftSupervisor

- Xây dựng class **ShiftSupervisor** (có thể hiểu là đốc công) kế thừa lớp **Employee** ở bài 1, lớp này có thêm một biến cho biết lương cả năm của đốc công, một biến lưu tiền thưởng.
- Hãy viết một hoặc nhiều hàm khởi tạo, các hàm set/get cho class trên
- Viết một đoạn code sử dụng class trên. Ví dụ: tính thu nhập của đốc công trong cả năm biết tiền thưởng sẽ bằng 3 lần tiền lương nhân với số tháng làm việc trong năm và chia cho 12 (nếu làm đủ cả năm thì được tiền thưởng bằng 3 tháng lương)

Bài 3. TeamLeader (bắt buộc)

- Xây dựng class **TeamLeader** (trưởng nhóm) kế thừa class **ProductionWorker** ở bài 1. Lớp này cần có thêm thuộc tính “số giờ đào tạo tối thiểu hằng tháng”, thuộc tính “số giờ đào tạo mà anh ta tham gia”, thuộc tính “số tiền thưởng hàng tháng”.
- Hãy viết một hoặc nhiều hàm khởi tạo, các hàm set/get cho class trên.
- Viết một chương trình minh họa sử dụng class trên. Ví dụ: tính lương

Bài 4. Time (bắt buộc)

Cho class **Time** như sau:

```
class Time
{
    protected:
        int hour;
        int min;
        int sec;
    public:
        Time()
            { hour = 0; min = 0; sec = 0; }

        Time(int h, int m, int s)
            { hour = h; min = m; sec = s; }

        int getHour() const
            { return hour; }

        int getMin() const
            { return min; }

        int getSec() const
            { return sec; }
};
```

- Viết class **MilTime** kế thừa từ class **Time** ở trên, **MilTime** cần chuyển đổi giờ từ định dạng 24h của class **Time** thành định dạng 12h + sáng/chiều. Sau đó viết chương trình cho người dùng nhập vào các giá trị giờ phút giây sau đó hiển thị giờ ở định dạng giờ 12
- **Lưu ý:** cần phải kiểm tra các giá trị giờ, phút, giây có hợp lệ hay không

Bài 5. TimeClock

- Viết class **TimeClock** kế thừa class **MilTime** ở bài 4, class này phải cung cấp một hàm cho phép tính khoảng cách giữa 2 thời điểm. Ví dụ từ 9h00 đến 13h20 thì khoảng cách là 4h20

Bài 6. GradedActivity

- Cho class **GradedActivity** (chấm điểm) như sau:

```
class GradedActivity
{
    private:
        double score; // To hold the numeric score
    public:
        GradedActivity()
        { score = 0.0; }

        GradedActivity(double s)
        { score = s; }

        void setScore(double s)
        { score = s; }

        double getScore() const
```

```

    { return score; }

char getLetterGrade() const
{
    char letterGrade; // To hold the letter grade

    if (score > 89)
        letterGrade = 'A';
    else if (score > 79)
        letterGrade = 'B';
    else if (score > 69)
        letterGrade = 'C';
    else if (score > 59)
        letterGrade = 'D';
    else
        letterGrade = 'F';

    return letterGrade;
}
}

```

- Viết class **Essay** (tiểu luận) kế thừa class trên, class này cho biết điểm số mà sinh viên nhận được. Trong đó thang điểm là 0 đến 100, được chia thành 4 mục như sau:
 - + Ngữ pháp: tối đa 30 điểm
 - + Chính tả: tối đa 20 điểm
 - + Độ dài: tối đa 20 điểm
 - + Nội dung: tối đa 30 điểm

Bài 7. PersonData

- Viết class **PersonData** có các trường sau: họ, tên, địa chỉ, thành phố, tỉnh, mã bưu điện, số điện thoại và các hàm set/get thích hợp
- Viết class **CustomerData** kế thừa từ **PersonData** và có thêm 2 trường: ID, check. Trong đó check là trường bool sẽ có giá trị true nếu khách hàng đã có trong danh sách mail, bằng false nếu không có trong danh sách mail
- Viết một chương trình đơn giản sử dụng lớp **CustomerData**

Bài 8. PreferredCustomer

- Viết class **PreferredCustomer** kế thừa từ lớp **CustomerData** ở trên. Lớp này có thêm 2 trường: điểm tích lũy và chiết khấu. Biết quy tắc chiết khấu như sau:
 - + nếu tích lũy được 500 \$ mua hàng thì các lần mua sau được chiết khấu 5%
 - + nếu tích lũy được 1000 \$ mua hàng thì các lần mua sau được chiết khấu 6%
 - + nếu tích lũy được 1500 \$ mua hàng thì các lần mua sau được chiết khấu 7%
 - + nếu tích lũy được 2000 \$ mua hàng thì các lần mua sau được chiết khấu 10%
- Viết một chương trình minh họa sử dụng class **PreferredCustomer**, viết thêm các hàm nếu cần

Bài 9. FilterFile

- Bộ lọc tệp là một class thực hiện đọc một file đầu vào, chuyển đổi nó theo một cách nào đó, sau đó ghi kết quả ra một file output. Viết 1 class **FilterFile** trừu tượng có hàm thuần ảo tên là **transform**, sau đó tạo 1 class kế thừa **FilterFile** cài đặt hàm **transform** có tác dụng mã hóa, một class khác tương tự nhưng thực hiện chuyển tất cả thành chữ hoa, một class khác thì chỉ copy nội dung file
- Các class phải có hàm : **void doFilter(ifstream &in, ofstream &out)** dùng để chuyển đổi file, và hàm: **char transform(char ch)** dùng để chuyển đổi

Bài 10. FilterFile++

- Viết 1 class kế thừa class **FilterFile** ở bài 9 , chức năng của class này là chèn thêm 1 dòng trống giữa mỗi 2 dòng trong file đầu vào

Bài 11. None

- Thiếu dữ kiện nên bỏ qua

Bài 12. Ship (bắt buộc)

- Viết class **Ship** (tàu thủy) có:
 - + Tên con tàu
 - + Năm xuất xưởng
 - + Một hàm khởi tạo và các hàm set/get
 - + Một hàm ảo tên là print để in ra tên con tàu
- Viết class **CruiseShip** (tàu du lịch) kế thừa từ class Ship có:
 - + Số hành khách tối đa
 - + Một hàm khởi tạo và các hàm set/get
 - + Hàm print ghi đè hàm print của lớp cha in ra tên con tàu và số hành khách
- Viết class **CargoShip** (tàu hàng) kế thừa class Ship có:
 - + Số tấn hàng chở được
 - + Một hàm khởi tạo và các hàm set/get
 - + Hàm print ghi đè hàm print của lớp cha in ra tên con tàu và số hàng chở được
- Sử dụng 3 class trên trong một chương trình minh họa đơn giản

Bài 13. BasicShape (bắt buộc)

- Viết một class thuần ảo tên là **BasicShape** (hình học) có :
 - + 1 thuộc tính private là **area** chứa diện tích của hình.
 - + hàm getArea trả lại giá trị của area
 - + hàm calcArea là hàm thuần ảo chịu trách nhiệm tính diện tích, sẽ được cài đặt ở các lớp con.
- Viết class **Circle** (hình tròn) kế thừa từ **BasicShape** có:
 - + centerX: tọa độ tâm hình tròn trên trục X
 - + centerY: tọa độ tâm hình tròn trên trục Y
 - + radius: bán kính hình tròn
 - + hàm khởi tạo 3 tham số ứng với 3 thuộc tính trên
 - + hàm get của 3 thuộc tính
 - + cài đặt hàm calcArea thực hiện tính diện tích và trả lại giá trị

- Viết class **Rectangle** (hình chữ nhật) kế thừa từ **BasicShape** có:
 - + thuộc tính dài và rộng
 - + hàm khởi tạo 2 tham số ứng với 2 thuộc tính dài và rộng
 - + các hàm get
 - + cài đặt hàm calcArea thực hiện tính diện tích và trả lại giá trị
- Viết một chương trình minh họa với 3 class trên

Bài 14: Bài tập nhóm

- Bài này được khuyến khích thực hiện theo nhóm để sinh viên làm quen với việc làm dự án theo team. Các yêu cầu cần được phân tích , thảo luận bởi cả nhóm để phân tích thành các class, thuộc tính và hàm của từng class cần phải nhận input gì, trả lại output là gì. Sau đó các sinh viên phân công mỗi người làm một số class để có thể ghép lại thành chương trình hoàn chỉnh.
- Thiết kế một lớp tài khoản ngân hàng chung chứa các thông tin như sau:
 - + số dư tài khoản
 - + số tiền gửi vào trong tháng
 - + số lần rút tiền trong tháng
 - + lãi suất hàng năm
 - + phí dịch vụ hàng tháng
- Thiết kế lớp tài khoản tiết kiệm kế thừa từ lớp trên có thêm trường status mô tả trạng thái tài khoản là active hay inactive.
 - + nếu số dư dưới 25\$ thì tài khoản trở thành trạng thái inactive và không rút thêm được tiền. tài khoản chỉ chuyển sang active khi có số dư lớn hơn hoặc bằng 25\$
- Bổ sung một số thuộc tính và phương thức cho 2 class trên nếu cần, sau đó tạo một chương trình minh họa