**BÀI THỰC TẬP 08 – LẬP TRÌNH XÂY DỰNG CHƯƠNG TRÌNH HƯỚNG ĐỐI TƯỢNG**

**A. MỤC TIÊU**

Cung cấp cho sinh viên kiến thức tổng quan về việc tổ chức, thiết kế, xây dựng một chương trình hướng đối tượng trên nền tảng C++

**B. NỘI DUNG**

* Tư duy về cách phân tích bài toán, tìm các lớp trong hệ thống.
* Thiết kế, xây dựng chương trình hoàn chỉnh trên C++.

**C. YÊU CẦU PHẦN CỨNG**

* Phần cứng: Dung lượng RAM từ 1GB
* Phần mềm: Hệ điều hành Windows, C-free 5.0

**D. KẾT QUẢ SAU KHI HOÀN THÀNH**

Hiểu và tự xây dựng được chương trình vận dụng kiến thức lập trình hướng đối tượng cài đặt lớp, kế thừa, đa hình, khuôn hình.

**E. TÓM TẮT LÝ THUYẾT**

1. **Các giai đoạn phát triển hệ thống**

Có năm giai đoạn để phát triển hệ thống phần mềm theo hướng đối tượng:

***Phân tích yêu cầu*** (Requirement analysis): Bằng việc tìm hiểu các trường hợp sử dụng (use case) để nắm bắt các yêu cầu của khách hàng, của vấn đề cần giải quyết. Qua trường hợp sử dụng này, các nhân tố bên ngoài có tham gia vào hệ thống cũng được mô hình hóa bằng các tác nhân.

***Phân tích*** (Analysis): Từ các đặc tả yêu cầu trên, hệ thống sẽ bước đầu được mô hình hóa bởi các khái niệm lớp, đối tượng và các cơ chế để diễn tả hoạt động của hệ thống. Trong giai đoạn phân tích chỉ mô tả các lớp trong lĩnh vực của vấn đề cần giải quyết chứ không đi sâu vào các chi tiết kỹ thuật.

***Thiết kế*** (Design: Trong giai đoạn thiết kế, các kết quả của quá trình phân tích được mở rộng thành một giải pháp kỹ thuật. Một số các lớp được thêm vào để cung cấp cơ sở hạ tầng kỹ thuật như lớp giao diện, lớp cơ sở dữ liệu, lớp chức năng, …

***Lập trình*** (Programming): Đây còn gọi là bước xây dựng, giai đoạn này sẽ đặc tả chi tiết kết quả của giai đoạn thiết kế.

***Kiểm tra*** (Testing): Trong giai đoạn kiểm tra, có bốn hình thức kiểm tra hệ thống: Kiểm tra từng đơn thể (unit testing) được dùng kiểm tra các lớp hoặc các nhóm đơn. Kiểm tra tính tích hợp (integration testing), được kết hợp với các thành phần và các lớp. Kiểm tra hệ thống (system testing) chỉ để kiểm tra xem hệ thống có đáp ứng được chức năng mà người dùng yêu cầu không. Kiểm tra tính chấp nhận được(acceptance testing) được thực hiện bởi khách hàng, việc kiểm tra cũng thực hiện giống như kiểm tra hệ thống.

**2. Cách tìm lớp**

Lớp nên được tìm từ phạm vi bài toán cần giải quyết, vì vậy tên của lớp cũng nên đặt tên các đối tượng thực mà cần biểu diễn. Để tìm ra lớp cho bài toán, cần trả lời các câu hỏi sau:

- Có thông tin nào cần lưu trữ hay phân tích không? Nếu có bất kỳ thông tin nào cần phải lưu trữ, biến đổi, phân tích hoặc xử lý thì đó chính là một lớp dự định cần xây dựng.

- Có hệ thống bên ngoài hay không? Hệ thống ngoài có thể được xem như các lớp mà hệ thống đang xây dụng chứa hoặc tương tác với nó.

- Có các mẫu thiết kế, thư viện lớp, thành phần, … hay không?

- Tác nhân đóng vai trò như thế nào trong hệ thống? Các tác nhân này nên được xem là lớp như người sử dụng, khách hàng, người điều khiển hệ thống,…

**3.** **Các bước cần thiết để thiết kế chương trình**

Để thiết kế một chương trình theo hướng đối tượng, cần trải qua bốn bước sau, từ đó xây dựng được một cây phả hệ mang tính kế thừa và các mối quan hệ giữa các đối tượng:

- Xác định các dạng đối tượng (lớp) của bài toán (định dang các đối tượng).

- Tìm kiếm các đặc tính chung (dữ liệu chung) trong các dạng đối tượng này, những gì chúng cùng nhau chia sẻ.

- Xác định được lớp cơ sở dựa trên cơ sở các đặc tính chung của các dạng đối tượng.

- Từ lớp cơ sở, sử dụng quan hệ tổng quát hóa để đặc tả trong việc đưa ra các lớp dẫn xuất chứa các thành phần, những đặc tính không chung còn lại của dạng đối tượng. Bên cạnh đó, còn đưa ra các lớp có quan hệ với các lớp cơ sở và lớp dẫn xuất; các quan hệ này có thể là quan hệ kết hợp, quan hệ tập hợp lại, quan hệ phụ thuộc. Từ các bước trên, thu được được cây phả hệ và quan hệ giữa các lớp

**F. HƯỚNG DẪN CHI TIẾT**

**Bài 1:** Công ty viễn thông ABC cung cấp dịch vụ điện thoại và internet hỗn hợp cho khách hàng theo hình thức trả sau. Để sử dụng dịch vụ, khách hàng cần ký hợp đồng với công ty. Trong hợp đồng, cần có các thông tin cá nhân của khách hàng (họ tên, chứng minh nhân dân, địa chỉ) và thông tin về cách tính cước mà khách hàng chọn. Cuối mỗi tháng, khách hàng sẽ được thông báo cước tùy theo lượng sử dụng của mình tương ứng với gói cước đã đăng ký ban đầu.

Hợp đồng với gói cước Basic có cách tính tiền như sau:

*Cước điện thoại* = *Thời gian gọi* (phút) \* *Đơn giá gọi* (1000 đồng/phút).

*Cước internet* = *Lưu lượng truy cập* (MB) \* *Đơn giá truy cập* (200 đồng/MB).

*Cước tổng* = *Cước điện thoại* + *Cước internet* + 10% VAT

Để thu hút thêm đối tượng khách hàng thường xuyên sử dụng internet, công ty mở rộng thêm hai loại hợp đồng mới có cách tính cước linh hoạt như sau:

|  |  |  |
| --- | --- | --- |
| **Gói cước** | **Cước điện thoại** | **Cước internet** |
| Data Free | Tương tự gói Basic | - Nếu *Lưu lượng truy cập <= Ngưỡng lưu lượng miễn phí* => Chỉ đóng cước thuê bao  - Nếu *Lưu lượng truy cập > Ngưỡng lưu lượng miễn phí* => Cước thuê bao + Cước lưu lượng vượt ngưỡng  Ghi chú:  - *Cước thuê bao và Ngưỡng lưu lượng miễn phí* được công ty xác định lúc lập hồ sơ hợp đồng đăng ký cho khách hàng và được ghi trong mỗi hợp đồng: Có thể khác nhau tùy vào lúc lập hợp đồng nhưng không đổi sau đó.  - *Cước lưu lượng vượt ngưỡng* tính theo công thức *Cước internet* của gói Basic. |
| Data Fix | Tương tự gói Basic + Giảm 10% giá cước | Mức cố định 1000.0000 đồng |

Hãy phân tích hệ thống, vẽ sơ đồ lớp thể hiện các lớp đối tượng, quan hệ giữa các lớp, các thuộc tính, các hàm trong mỗi lớp và viết chương trình cho phép công ty quản lý các hợp đồng trong một danh sách duy nhất với 2 chức năng: cho phép khách hàng đăng ký hợp đồng mới và thông báo tiền cước cho tất cả khách hàng vào cuối tháng.

**Hướng dẫn:**

**- Bước 1**: Từ yêu cầu đề bài cần xây dựng các lớp Hợp đồng *HopDong*, Lớp gói cước *Basic,* lớp gói cước *DataFree,*lớp gói cước *DataFix*và lớp danh sách hợp đồng*DanhSachHopDong.*Chi tiết hơn nữa về các thuộc tính và phương thức của các lớp được thể hiện qua các bảng dưới đây:

1. **Lớp HopDong**

*Bảng 8. 1 Thành phần lớp HopDong*

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Phạm vi** | **Giải thích** |
| hoTen | private | Họ tên |
| cmnd | private | Chứng minh nhân dân |
| diaChi | private | Địa chỉ |
| goiCuoi | private | Gói cước |
| **Phương thức** | **Phạm vi** | **Giải thích** |
| HopDong() | public | Hàm tạo đối tượng |
| nhap() | public | Hàm nhập |
| xuat() | public | Hàm xuất |
| tinhCuocCuoiThang() | public | Hàm tính cước cuối tháng |
| suDung() | public | Tính thời gian sử dụng |

1. **Lớp Basic**

*Bảng 8. 2 Thành phần lớp Basic*

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Phạm vi** | **Giải thích** |
| thoiGianGoi | protected | Thời gian gọi |
| donGiaGoi | protected | Đơn giá gọi |
| luuLuong | protected | Lưu lượng |
| donGiaInternet; | protected | Đơn giá Internet |
| **Phương thức** | **Phạm vi** | **Giải thích** |
| cuocDienThoai() | public | Tính cước điện thoại |
| cuocInternet() | public | Tính cước Internet |
| nhap() | public | Nhập thông tin |
| xuat() | public | Xuất thông tin |
| cuocTong() | public | Tính tổng cước |
| suDung() | public | Tính thời gian sử dụng |

1. **Lớp DataFree**

*Bảng 8. 3 Thành phần lớp DataFree*

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Phạm vi** | **Giải thích** |
| cuocThueBao  NguongMP | Private  private | Cước thuê bao  Ngưỡng Miễn phí |
| **Phương thức** | **Phạm vi** | **Giải thích** |
| DataFree() | public | Hàm tạo đối tượng DataFree |
| cuocInternet() | public | Tính cước Internet |
| nhap() | public | Nhập thông tin |
| xuat() | public | Xuất thông tin |
| tinhTienPhaiDong() | public | Tính tiền phải đóng |

1. **Lớp DataFix**

*Bảng 8. 4 Thành phần lớpDataFix*

|  |  |  |
| --- | --- | --- |
| **Phương thức** | **Phạm vi** | **Giải thích** |
| cuocDienThoai() | public | Tính cước điện thoại |
| cuocInternet() | public | Tính cước Internet |
| xuat() | public | Xuất thông tin |

1. **Lớp DanhSachHopDong**

*Bảng 8. 5 Thành phần lớp DanhSachHopDomg*

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Phạm vi** | **Giải thích** |
| dsHopDong | private | Danh sách hợp đồng |
| **Phương thức** | **Phạm vi** | **Giải thích** |
| DanhSachHopDong() | public | Hàm tạo danh sách hợp đồng |
| ~DanhSachHopDong() | public | Hàm hủy danh sách hợp đồng |
| dangKy() | public | Hàm đăng ký thông tin hợp đồng |
| thongBaoTienCuoc() | public | Thông báo tiền cước. |

**- Bước 2:** Cài đặt chương trình:

- Xây dựng lớp Basic mô tả gói cước Basic

|  |
| --- |
| **class Basic**  {  **protected:**  int thoiGianGoi;  int donGiaGoi;  int luuLuong;  int donGiaInternet;  **public:**  virtual int cuocDienThoai();  virtual int cuocInternet();  virtual void nhap();  virtual void xuat();  virtual int cuocTong();  void suDung(int, int);  Basic();  virtual ~Basic();  };  **Basic::Basic()**  {  thoiGianGoi = 0;  donGiaGoi = 1000;  luuLuong = 0;  donGiaInternet = 200;  }  **Basic::~Basic()**  {}  **int Basic::cuocDienThoai()**  {  return thoiGianGoi \* donGiaGoi;  }  **int Basic::cuocInternet()**  {  return luuLuong \* donGiaInternet;  }  **int Basic::cuocTong()**  {  return 1.1 \* (cuocInternet() + cuocDienThoai());  }  **void Basic::nhap()**  {  cout<<"- Nhap thoi gian goi: "; cin>>thoiGianGoi;  cout<<"- Nhap luu luong su dung: "; cin>>luuLuong;  }  **void Basic::xuat()**  {  cout << "\t - Ten goi cuoc: Basic";  }  **void Basic::suDung(int \_thoiGianGoi, int \_luuLuong)**  {  thoiGianGoi += \_thoiGianGoi;  luuLuong += \_luuLuong;  } |

- Xây dựng lớp DataFree kế thừa từ lớp Basic

|  |
| --- |
| **class DataFree: public Basic**  {  int cuocThueBao;  int NguongMp;  **public:**  DataFree();  int cuocInternet();  void nhap();  void xuat();  int cuocTong();  ~DataFree();  };  DataFree::DataFree() {}  DataFree::~DataFree() {}  **int DataFree::cuocInternet()**  {  if (luuLuong<= NguongMp)return cuocThueBao;  else return  cuocThueBao+(luuLuong-NguongMp)\* donGiaInternet;  }  **void DataFree::nhap()**  {  Basic::nhap();  cout << "- Nhap cuoc thue bao: ";  cin >> cuocThueBao;  cout << "- Nhap nguong luu luong mien phi: ";  cin >> NguongMp;  }  **void DataFree::xuat()**  {  cout << "\t -Ten goi cuoc: DataFree";  }  **int DataFree::cuocTong()**  {  return 1.1 \* (cuocInternet() + Basic::cuocDienThoai());  } |

- Xây dựng lớp DataFix kế thừa từ lớp Basic

|  |
| --- |
| **class DataFix: public Basic**  {  **public:**  int cuocDienThoai();  int cuocInternet();  void xuat();  int cuocTong();  ~DataFix();  };  **DataFix::~DataFix()** {}  **int DataFix::cuocDienThoai()**  {  return Basic::cuocDienThoai() \* 0.9;  }  **int DataFix::cuocInternet()**  {  return 1000000;  }  **int DataFix ::cuocTong()**  {  return 1.1 \* (cuocInternet() + cuocDienThoai());  }  **void DataFix::xuat()**  {  cout << "\t - Ten goi cuoc: DataFix";  } |

- Xây dựng lớp HopDong

|  |
| --- |
| **class HopDong**  {  char hoTen[30];  char cmnd[20];  char diaChi[40];  Basic \*goiCuoc;  **public:**  HopDong();  ~HopDong();  void nhap();  void xuat();  int tinhCuocCuoithang();  void suDung(int, int);  }; |

- Xây dựng lớp DanhSachHopDong

|  |
| --- |
| **class DanhSachHopDong**  {  **private:**  HopDong\* dsHopDong;  int sohopdong;  **public:**  void dangKy();  void thongBaoTienCuoc();  DanhSachHopDong();  ~DanhSachHopDong();  }; |

**Bài 2:** Siêu thị VinMart là một khu kinh doanh sầm uất của thành phố. Mặt bằng siêu thị được chia thành các quầy có diện tích khác nhau cho tiểu thương thuê. Có ba loại quầy tương ứng với ba loại mặt hàng được phép kinh doanh tại chợ là: thực phẩm, quần áo, và trang sức. Đơn giá thuê ở mỗi loại quầy là như nhau: 50.000.000 đồng/m2/năm. Thông tin chung để quản lý mỗi quầy bao gồm:

- Mã quầy hàng.

- Số thứ tự quầy.

- Diện tích quầy (m2).

- Doanh thu

Vào cuối năm, số tiền mỗi tiểu thương thuê quầy phải đóng = tiền thuê quầy + thuế doanh thu. Tiền thuê quầy = đơn giá thuê \* diện tích quầy. Thuế doanh thu tùy thuộc vào từng loại quầy và cho bởi bảng sau:

|  |  |
| --- | --- |
| **Loại quầy** | **Thuế doanh thu** |
| Quầy thực phẩm | 5% |
| Quầy quần áo | 10% |
| Quầy trang sức | - Phần doanh thu < 100.000.000 đồng: 20%  - Phần doanh thu >= 100.000.000 đồng: 30% |

Ngoài ra, những tiểu thương thuê quầy thực phẩm phải đóng thêm tiền sử dụng dịch vụ đông lạnh để bảo quản thực phẩm mà quầy của mình đã sử dụng trong năm (số tiền khác nhau ở từng quầy). Hãy vẽ sơ đồ lớp và cài đặt chương trình cho phép quản lý thực hiện các chức năng sau:

* Nhập vào danh sách thông tin các quầy được thuê.
* Đếm số lượng từng loại quầy hàng trong siêu thị.
* Hiển thị thông tin từng quầy hàng.
* Tính tổng số tiền các quầy phải đóng hàng năm.

**Hướng dẫn:**

**- Bước 1:**

Từ yêu cầu đề bài cần xây dựng các lớp gồm: Lớp quầy *Quay*, lớp quầy quần áo *QuayQuanAo*, lớp quầy thực phẩm *QuayThucPham*, lớp quầy trang sức *QuayTrangSuc* và lớp siêu thị *SieuThi.* Chi tiết hơn nữa về các thuộc tính và phương thức của các lớp được thể hiện qua các bảng dưới đây:

1. **Lớp Quay**

*Bảng 8. 6 Thành phần lớp Quay*

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Phạm vi** | **Giải thích** |
| **maQuay** | **private** | **Mã quầy** |
| soThuTu | private | Số thứ tự quầy hàng |
| dienTich | private | Diện tích quầy |
| doanhThu | protected | Doanh thu hàng tháng. |
| **Phương thức** | **Phạm vi** | **Giải thích** |
| tinhTinhThueDoanhThu() | protected | Tính thuế doanh thu mặt hàng |
| tinhTienDichVu() | protected | Tính tiền dịch vụ, |
| tinhTienThueQuay() | protected | Tính tiền thuê quầy |
| tinhTienPhaiDong() | protected | Tính tiền phải đóng |
| check() | protected | Đánh dấu loại quầy hàng. |
| nhap() | protected | Nhập thông tin quầy hàng |
| xuat() | protected | Xuất thông tin quầy hàng |

1. **Lớp QuayQuanAo**

*Bảng 8. 7 Thành phần lớp QuayQuanAo*

|  |  |  |
| --- | --- | --- |
| **Phương thức** | **Phạm vi** | **Giải thích** |
| tinhTinhThueDoanhThu() | protected | Tính thuế doanh thu mặt hàng |
| tinhTienPhaiDong() | protected | Tính tiền phải đóng |
| nhap() | protected | Nhập thông tin quầy hàng quần áo |
| xuat() | protected | Xuất thông tin quầy hàng quần áo |
| check() | protected | Đánh dấu loại quầy hàng. |

1. **Lớp QuayThucPham**

*Bảng 8. 8 Thành phần lớp QuayThucPham*

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Phạm vi** | **Giải thích** |
| tienDichVu | private | Tiền dịch vụ |
| **Phương thức** | **Phạm vi** | **Giải thích** |
| tinhTinhThueDoanhThu() | protected | Tính thuế doanh thu mặt hàng |
| tinhTienDichVu() | protected | Tính tiền dịch vụ |
| tinhTienPhaiDong() | protected | Tính tiền phải đóng |
| nhap() | protected | Nhập thông tin quầy hàng quần áo |
| xuat() | protected | Xuất thông tin quầy hàng quần áo |
| check() | protected | Đánh dấu loại quầy hàng. |

1. **Lớp QuayTrangSuc.**

*Bảng 8. 9 Thành phần lớp QuayTrangSuc*

|  |  |  |
| --- | --- | --- |
| **Phương thức** | **Phạm vi** | **Giải thích** |
| tinhTinhThueDoanhThu() | protected | Tính thuế doanh thu mặt hàng |
| tinhTienPhaiDong() | protected | Tính tiền phải đóng |
| nhap() | protected | Nhập thông tin quầy hàng quần áo |
| xuat() | protected | Xuất thông tin quầy hàng quần áo |
| check() | protected | Đánh dấu loại quầy hàng. |

1. **Lớp SieuThi**

*Bảng 8. 10 Thành phần lớp SieuThi*

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Phạm vi** | **Giải thích** |
| dsQuay | private | Danh sách quầy hàng |
| **Phương thức** | **Phạm vi** | **Giải thích** |
| Nhap() | public | Nhập thông tin danh sách các quầy hàng |
| Xuat() | public | Xuất thông tin danh sách các quầy hàng |
| tinhTongsoTien() | public | Tính tổng số tiền phải đóng của danh sách quầy hàng |
| ThongkeSL() | public | Thống kê số lượng từng quầy trong danh sách. |

**- Bước 2:** Cài đặt chương trình:

Chương trình sử dụng tính đa hình trong quá trình cài đặt các lớp.

- Xây dựng lớp Quay quản lý các đối tượng quầy hàng.

|  |
| --- |
| **class Quay**  {  **private:**  int soThuTu;  int dienTich;  **protected:**  double doanhThu;  **virtual double tinhThueDoanhThu()=0;**  **virtual double tinhTienDichVu()**  {  return 0;  }  **double tinhTienThueQuay()**  {  return 400000000 \* dienTich;  }  **public:**  **virtual double tinhTienPhaiDong()**  {  return tinhTienThueQuay() + tinhTienDichVu() + tinhThueDoanhThu();  }  **virtual void nhap(istream & is)**  {  cout << "Nhap so thu tu: ";  is >> soThuTu;  cout << "Nhap dien tich: ";  is >> dienTich;  cout << "Nhap doanh thu: ";  is >> doanhThu;  }  **virtual void xuat()**  {  cout << "\n So thu tu: ";  cout << soThuTu;  cout << "\nDien tich: ";  cout<< dienTich;  cout <<"\nDoanh thu: ";  cout<< doanhThu;  }  **virtual int check() = 0;**  **int Getdt()**  {  return dienTich;  }  }; |

- Xây dựng lớp QuayThucPham kế thừa từ lớp Quay.

|  |
| --- |
| **class QuayThucPham : public Quay**  {  **private:**  double tienDichVu;  **protected:**  **double tinhThueDoanhThu()**  {  return doanhThu \* 5;  }  **double tinhTienDichVu()**  {  return tienDichVu;  }  **virtual double tinhTienPhaiDong()**  {  return tinhTienThueQuay() + tinhTienDichVu() + tinhThueDoanhThu();  }  **virtual void nhap(istream & is)**  {  Quay::nhap(is);  cout << " Nhap tien dich vu: ";  is>>tienDichVu;  }  **virtual void xuat()**  {  Quay::xuat();  cout << "\nTien dich vu: ";  cout << tienDichVu;  cout << "\nThue doanh thu: ";  cout << QuayThucPham::tinhThueDoanhThu();  cout<<"\nTien phai dong: "<<QuayThucPham::tinhTienPhaiDong();  }  **virtual int check()**{return 1; }  }; |

- Tương tự, sinh viên cài đặt lớp QuayQuanAo, QuayTrangSuc kế thừa từ lớp Quay. Cài đặt lớp SieuThi quản lý thông tin chung các quầy hàng. Sau đây là một số hàm được cài đăt trong lớp

**Bài 3: Xây dựng chương trình quản lý thư viên**

Chương trình quản lý thông tin liên quan đến việc tổ chức các hoạt động hằng ngày của một thư viện như: mượn và trả sách, tìm kiếm thông tin về sách. Chương trình cũng cho phép lưu trữ các thông tin về sách, bạn đọc. Các yêu cầu về chương trình như sau:

Khi làm việc với thư viện một bạn đọc mới phải thực hiện đăng ký thành viên. Sau khi đăng ký bạn đọc mới sẽ có thẻ đăng ký và có thể chính thức đọc sách và được mượn sách. Thủ tục đăng ký chỉ đơn giản là bạn đọc cần điền các thông tin cá nhân của mình vào phiếu đăng ký. Để đơn giản hóa bài toán, ta chỉ cần 2 thông tin quan trọng nhất là số hiệu bạn đọc và tên bạn đọc. Để mượn sách, bạn đọc cần phải điền vào một phiếu mượn, trong đó nội dung sẽ bao gồm thông tin về bạn đọc và các cuốn sách mà bạn đọc muốn mượn. Chỉ xét trường hợp phiếu mượn đơn giản bao gồm một đầu sách nào đó. Khi bạn đọc mượn một cuốn sách nào đó thì số lượng các đầu sách của cuốn sách này sẽ được cập nhật lại (giảm đi một cuốn). Thư viện quản lý sách sách theo cách, mỗi cuốn sách có thể có nhiều bản (đầu sách) khác nhau phục vụ cho nhiều bạn đọc cùng một lúc. Mỗi cuốn sách có các thông tin chính là mã sách và tên sách. Mỗi đầu sách cũng có một số hiệu riêng phân biệt với các đầu sách khác.Thư viện thường xuyên thực hiện bổ sung thêm sách và số các đầu sách của từng loại sách. Khi có bạn đọc trả sách, thủ thư sẽ nhận sách trả từ bạn đọc dựa trên phiếu mượn của bạn đọc đó. Nếu trả sách thành công thì số lượng đầu sách sẽ được cập nhật (tăng lên một cuốn). Ngoài những chức năng trên thủ thư có thể thống kê danh sách tất cả các bạn đọc đã tham gia vào thư viện, liệt kê danh sách các cuốn sách có trong thư viện cùng với số đầu sách của cuốn sách đó.[1]

**Hướng dẫn:**

**- Bước 1:** Từ yêu cầu đề bài cần xây dựng các lớp gồm: Lớp bạn đọc *BanDoc*, lớp sách *Sach*, lớp đầu sách *DauSach*, lớp phiếu mượn *PhieuMuon*. Chi tiết hơn nữa về các thuộc tính và phương thức của các lớp được thể hiện qua các bảng dưới đây:

1. **Lớp BanDoc:**

*Bảng 8. 11 Thành phần lớp BanDoc*

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Phạm vi** | **Giải thích** |
| ma | private | Mã số bạn đọc |
| ten | private | Tên bạn đọc |
| sachMuon | private | Danh sách các phiếu mượn- tối đa năm phiếu |
| cacBandoc | stalic | Danh sách các bạn đọc |
| tep | stalic | Tên tệp lưu trữ |
| **Phương thức** | **Phạm vi** | **Giải thích** |
| ghiMuon() | public | Ghi thông tin mượn trả sách |
| muon(maSach) | public | Mượn một cuốn sách |
| tra(dauSach) | public | Trả một cuốn sách |
| dangKy() | public | Đăng ký một bạn đọc mới |
| timBD(maBD) | public | Tìm kiếm một bạn đọc |
| xoaDL() | stalic | Xóa dữ liệu khỏi bộ nhớ |
| docTep() | stalic | Đọc dữ liệu từ tệp |
| ghiTep() | stalic | Ghi dữ liệu ra tệp |
| inDS() | stalic | In danh sách các bạn đọc |

1. **Lớp Sach:**

*Bảng 8. 12 Thành phần lớp Sach*

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Phạm vi** | **Giải thích** |
| ma | private | Mã số sách |
| ten | private | Tên sách |
| cacDauSach | private | Danh sách các đầu sách |
| khoSach | stalic | Danh sách các sách trong thư viện |
| tep | stalic | Tệp lưu trữ |
| **Phương thức** | **Phạm vi** | **Giải thích** |
| timChuaMuon | public | Tìm một đầu sách chưa mượn |
| timSach(maSach) | stalic | Tìm một cuốn sách |
| timDauSach(maSach,id) | stalic | Tìm một đầu sách |
| nhapSach() | public | Nhập một cuốn sách mới vào thư viện |
| xoaDL() | stalic | Xóa dữ liệu khỏi bộ nhớ |
| docTep() | stalic | Đọc dữ liệu từ tệp |
| ghiTep() | stalic | Ghi dữ liệu ra tệp |
| inDS | stalic | In danh sách các cuốn sách |

1. **Lớp DauSach**

*Bảng 8. 13 Thành phần lớp DauSach*

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Phạm vi** | **Giải thích** |
| sach | private | Thông tin cuốn sách |
| id | private | Mã số đầu sách |
| muon | private | Phiếu mượn sách (NULL nếu không có bạn độc nào mượn đầu sách này) |
| **Phương thức** | **Phạm vi** | **Giải thích** |
| chuaMuon | public | Đặt trạng thái của đầu sách về thành chưa mượn |
| choMuon(banDoc) | public | Cho banDoc mượn một đầu sách |
| tra() | public | Trả một cuốn sách |

1. **Lớp PhieuMuon:**

*Bảng 8. 14 Thành phần lớp PhieuMuon*

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Phạm vi** | **Giải thích** |
| dauSach | private | Thông tin đầu sách được mượn |
| banDoc | private | Thông tin bạn đọc mượn sách |
| Tep | stalic | Tệp lưu trữ |
| **Phương thức** | **Phạm vi** | **Giải thích** |
| docTep() | stalic | Đọc dữ liệu từ tệp |
| ghiTep() | stalic | Ghi dữ liệu ra tệp |

**Bước 2**: Hướng dẫn cài đặt chương trình:

- Lớp bạn đọc BanDoc:

|  |
| --- |
| **class BanDoc**  {  **friend class PhieuMuon;**  char \*ma;  char \*ten;  int soMuon;  //moi nguoi chi muon toi da 5 quyen  PhieuMuon \*sachMuon[5];  //du lieu luu tru cac ban doc  **static int soBanDoc;**  //co toi da 100 ban doc dang ky  **static BanDoc \*cacBanDoc[100];**  //ten tep luu tru thong tin ban doc  **static const char \*tep;**  //ghi thong tin mot ban doc ra tep  **void ghiMuon(ofstream &os);**  //mot ban doc thuc hien muon mot dau sach, ham nay chi dung de cai dat  **void muon(DauSach\* Dausach);**  **public:**  **BanDoc(char \*m, char \*t)**  {  ma= strdup(m);  ten= strdup(t);  soMuon = 0;  }  **~BanDoc()**  {  delete ma;  delete ten;  }  //phuong thuc phuc vu viec muon tra sach cua ban doc  **PhieuMuon\* muon(char\* maSach);**  **void tra(DauSach\* dauSach);**  //cac ham dung de quan ly cac ban doc  **static void dangki();**  **static BanDoc\* timBD(char \*ma);**  //Xoa du lieu ban doc khoi bo nho  **static void xoaDL();**  //doc ghi thong tin ban doc tren tep  **static void ghiTep();**  **static void docTep();**  //in danh sach cac ban doc  **static void inDS();**  }; |

- Lớp sách Sach:

|  |
| --- |
| **class Sach**  {  **friend class DauSach;**  //thong tin ve sach  char \*ma;  char \*ten;  //so dau sach n  int soDauSach;  //mang du lieu dong luu ve cac dau sach  DauSach \*\*cacDauSach;  //du lieu ve cac quyen sach trong thu vien toi da 500 quyen  static int soSach;  static Sach \*khoSach[500];  //ten tep du lieu luu tru sach  static const char \*tep;  **public:**  Sach(char \*ma\_, char \*ten\_,int n);  ~Sach();  //tim mot dau sach chua muon cho quyen sach nay  //tra ve null ci nghia la da cho muon het  DauSach\* timChuaMuon();  //cac phuong thuc quan ly kho sach  static Sach\* timSach(char \*maSach);  static DauSach\* timDauSach(char \*maSach, int id);  static void nhapSach();  //Xoa du lieu ve sach khoi bo nho  static void xoaDL();  //doc ghi du lieu ve sach voi tep  static void docTep();  static void ghiTep();  //in danh sach cac quyen sach  static void inDS();  }; |

- Lớp đầu sách DauSach

|  |
| --- |
| **class DauSach**  {  friend class PhieuMuon;  friend class BanDoc;  friend class Sach;  Sach\* sach;  int id;  PhieuMuon \*muon;  **void ghiSach(ofstream& os)**  {  os << sach -> ma << endl;  os << id << endl;  }  **public:**  DauSach(Sach\* s, int id\_):  sach(s), id(id\_), muon(0){}  ~DauSach()  {  if (muon) delete muon;  }  int chuaMuon() {return muon==0; }  //thuc hien cho mot ban doc muon  PhieuMuon\* choMuon(BanDoc\* bd)  {  muon=new PhieuMuon(this,bd);  return muon;  }  //khi mot ban doc tra sach se goi den ham nay  void tra()  {  delete muon;  muon=0;  }  }; |

**-** Lớp phiếu mượn PhieuMuon

|  |
| --- |
| **class PhieuMuon**  {  friend class BanDoc;  DauSach\* dauSach;  BanDoc\* banDoc;  //ten tep du lieu phieu muon  static const char\* tep;  **public:**  PhieuMuon(DauSach\* ds, BanDoc\* bd):  dauSach(ds), banDoc(bd){}  //doc ghi thong tin ve cac phieu muon  static void docTep();  static void ghiTep();  }; |

**Bài 4**: Xây dựng chương trình quản lý danh sách các giao dịch. Hệ thống quản lý 2 loại giao dịch

- Giao dịch vàng: Mã giao dịch, ngày giao dịch (ngày, tháng, năm), đơn giá, số lượng, loại vàng. Thành tiền được tính như sau: thành tiền = số lượng \* đơn giá.

Giao dịch tiền tệ: Mã giao dịch, ngày giao dịch (ngày, tháng, năm), Đơn giá, số lượng, tỉ giá, loại tiền tệ có 3 loại: tiền Việt Nam, tiền USD, tiền Euro. Thành tiền được tính như sau:

- Nếu là tiền USD hoặc Euro thì: thành tiền = số lượng \* đơn giá\* tỉ giá

- Nếu là tiền VN thì: thành tiền = số lượng \* đơn giá

Tạo menu chương trình thực hiện các chức năng sau:

* Nhập xuất danh sách các loại giao dịch.
* Tính tổng số lượng cho từng loại.
* Tính trung bình thành tiền của giao dịch tiền tệ.
* Xuất ra các giao dịch có đơn giá > 1 tỷ.
* Đếm những giao dich vàng ngày 20/3/2015
* Sắp xếp danh sách theo chiều giảm của ngày giao dịch.
* Thêm giao dịch vào cuối danh sách
* Xóa giao dịch vàng ngày 20/05/2010

**Bài 5:** Xây dựng chương trình quản lý danh sách hoá đơn tiền điện của khách hàng. Thông tin bao gồm các loại khách hàng :

* *Khách hàng Việt Nam:* mã khách hàng, họ tên, ngày ra hoá đơn (ngày, tháng, năm), đối tượng khách hàng (sinh hoạt, kinh doanh, sản xuất): số lượng (số KW tiêu thụ), đơn giá, định mức. Thành tiền được tính như sau:

- Nếu số lượng <= định mức thì: thành tiền = số lượng \* đơn giá.

- Ngược lại thì: thành tiền = số lượng \* đơn giá \* định mức + số lượng KW vượt định mức \* Đơn giá \* 2.5.

* *Khách hàng nước ngoài:* mã khách hàng, họ tên, ngày ra hoá đơn (ngày, tháng, năm), quốc tịch, số lượng, đơn giá. Thành tiền được tính = số lượng \* đơn giá.

Tạo menu chương trình thực hiện các chức năng sau:

- Xây dựng các lớp với chức năng thừa kế.

- Nhập xuất danh sách các hóa đơn khách hàng.

- Tính tổng số lượng cho từng loại khách hàng.

- Tính trung bình thành tiền của khách hàng người nước ngoài.

- Xuất ra các hoá đơn trong tháng 09 năm 2013 (cùa cả 2 loại khách hàng).

**Bài 6:** Giả sử cần xây dựng chương trình quản lý dùng cho một học viện nghiên cứu giảng dạy và ứng dụng. Đối tượng quản lý bao gồm các sinh viên đang theo học, các nhân viên đang làm việc tại học viện, các khách hàng đến giao dịch mua bán sản phẩm ứng dụng. Dựa vào một số đặt tính của từng đối tượng, người quản lý cần đưa ra cách thức đánh giá khác nhau. Vậy hãy xây dựng các lớp sau:

- Lớp Person: bao gồm các thuộc tính họ tên, địa chỉ.

- Các lớp Student, Employee, Customer (mô tả dưới đây) thừa kế lớp Person.

* Lớp Student: bao gồm các thuộc tính điểm môn học 1, điểm môn học 2, và các phương thức: tính điểm TB, đánh giá.
* Lớp Employee: bao gồm thuộc tính heSoLương, và các phương thức: tính lương, đánh giá.
* Lớp Customer: bao gồm thuộc tính tên công ty, trị giá hóa đơn, đánh giá

- Lớp có 1 biến danh sách để lưu các sinh viên, nhân viên, khách hàng, biến lưu tổng số người có trong danh sách, phương thức thêm một người vào danh sách (thông số Person), xóa 1 người khỏi danh sách (nhận thông số là họ tên của người cần xóa), sắp xếp danh sách theo thứ tự họ tên, phương thức xuất danh sách.