

ĐỀ CƯƠNG THỰC HÀNH

Môn : Lập trình hướng đối tượng (CT176)

Số tiết: 30 tiết - Số buổi thực hành: 5 buổi, thi TH vào buổi 6

Áp dụng cho học kỳ 2, năm học 2024-2025

Chú ý: Sinh viên phải thực hiện tất cả các chủ điểm được quy định trong từng buổi thực hành, nhưng không yêu cầu phải thực hiện tất cả các bài trong mỗi buổi.

1. Buổi 1:

○ Mục đích:

Sinh viên tập làm quen với ngôn ngữ Java.

Làm quen với công cụ, môi trường lập trình Java: cài đặt, soạn thảo, biên dịch, thực thi, ...

Viết các chương trình Java đơn giản sử dụng cấu trúc điều khiển, các phép chuyển đổi kiểu, nhập xuất cơ bản, xử lý chuỗi, xử lý ngoại lệ, ... trong Java.

○ Yêu cầu:

Bài 1: Dùng Eclipse soạn thảo và thực thi chương trình HelloWorld dùng để hiển thị ra màn hình câu "Hello Java".

Bài 2: Đặt các biến môi trường để có thể biên dịch và thực thi chương trình Java bằng cơ chế dòng lệnh. Dùng 1 trình soạn thảo văn bản bất kỳ để biên soạn chương trình HelloWorld ở trên. Kiểm tra việc biên dịch và thực thi bằng cơ chế dòng lệnh.

Bài 3: Viết chương trình nhập vào một ký tự bất kỳ, in ký tự đó ra màn hình. Chương trình kết thúc khi nhập ký tự 'q'.

Bài 4: Viết chương trình nhập vào 2 số nguyên a và b. Tính tổng và hiển thị ra màn hình. Nếu số nhập vào không đúng định dạng thì hiển thị thông báo lỗi, yêu cầu nhập lại.

Bài 5: Viết chương trình giải phương trình bậc 2 trong đó có hàm **static** giải phương trình bậc 2 (3 tham số).

Bài 6: Viết chương trình nhập vào 1 số nguyên. Kiểm tra xem số đó có phải là số nguyên tố không? Hiển thị số nguyên đó ra màn hình dưới dạng số nhị phân.

Bài 7: Viết chương trình nhập vào 1 chuỗi họ tên. Hiển thị ra tên của họ tên đó. Nên thiết kế chương trình gồm 1 hàm dùng để tách tên và hàm main.

Bài 8: Viết chương trình thực hiện các công việc sau:

- Nhập vào 1 danh sách các số nguyên.
- Nhập vào 1 số nguyên x bất kỳ. Đếm xem trong danh sách có bao nhiêu số x.
- Sắp xếp danh sách theo thứ tự tăng dần.
- Hiển thị danh sách ra màn hình.

2. Buổi 2:

○ Mục đích:

Sinh viên tập làm quen với phong cách lập trình hướng đối tượng.

Cài đặt 1 số lớp đơn giản, viết các phương thức và hàm xây dựng của lớp.

Sử dụng các lớp vừa định nghĩa: tạo đối tượng và khai thác đối tượng.

○ Yêu cầu:

Bài 1: Cài đặt lớp **Diem** (Điểm trong không gian 2 chiều) gồm:

Thuộc tính: x, y là số nguyên.

Các phương thức bao gồm:

- + Hàm xây dựng mặc nhiên:
- + Hàm xây dựng có 2 tham số:
- + Nhập tọa độ cho điểm từ bàn phím:

Diem();
Diem(int, int);
void nhapDiem ();

- + Hiển thị ra màn hình tọa độ điểm theo dạng (x,y): `void hienThi();`
- + Dời điểm đi độ dời (dx, dy): `void doiDiem(int dx, int dy);`
- + Lấy ra giá trị hoành độ của điểm: `int giaTriX();`
- + Lấy ra giá trị tung độ của điểm: `int giaiTriY();`
- + Tính khoảng cách từ điểm đó đến gốc tọa độ: `float khoangCach();`
- + Tính khoảng cách từ điểm đó đến 1 điểm khác: `float khoangCach(Diem d);`

Viết lớp `SDDiem` có chứa hàm `main()` khai thác lớp vừa định nghĩa :

- + Tạo ra điểm A tọa độ (3,4). Hiển thị tọa độ điểm A ra màn hình.
- + Tạo ra điểm B với giá trị nhập từ bàn phím. Hiển thị tọa độ điểm B ra màn hình.
- + Tạo ra điểm C đối xứng với điểm B qua gốc tọa độ. Hiển thị tọa độ điểm C ra màn hình.
- + Hiển thị ra màn hình khoảng cách từ điểm B đến tâm O.
- + Hiển thị ra màn hình khoảng cách từ điểm A đến điểm B.

Bài 2: Cài đặt lớp **Date** gồm :

Các thuộc tính: ngày, tháng, năm.

Các phương thức bao gồm:

- + Hàm xây dựng: mặc nhiên `Date()` và hàm có 3 tham số `Date(int, int, int)`.
- + Hàm hiển thị thông tin ngày ra màn hình.
- + Hàm kiểm tra xem ngày có hợp lệ hay không ? `boolean hopLe();`
 Chẳng hạn: Ngày 31/6/2000 hay 29/2/1999 là không hợp lệ.
- + Hàm nhập giá trị từ bàn phím, nếu không hợp lệ thì yêu cầu nhập lại.
- + Hàm tính ra ngày hôm sau là ngày nào: `Date ngayHomSau();`
 Ví dụ: Gọi hàm `ngayHomSau()` trên đối tượng 30/06/2007 thì kết quả là 01/07/2007
- + Hàm cộng 1 `Date` với số ngày `n` nào đó: `Date congNgay(int n);`
 Ví dụ: ngày 15/6/2000 cộng thêm 20 ngày là ngày 05/7/2000

Viết lớp `SDDate` có chứa hàm `main()` kiểm tra tính đúng đắn của lớp vừa định nghĩa.

Bài 3: Cài đặt lớp **PhanSo** (Phân số) gồm:

Các thuộc tính : tử số và mẫu số có kiểu số nguyên.

Các hàm xây dựng gồm :

- + Hàm xây dựng mặc nhiên : `PhanSo();`
- + Hàm xây dựng gồm nhiều đối số : `PhanSo(int tu , int mau);`

Các phương thức khác bao gồm :

- + Hàm nhập giá trị cho 1 phân số. Nếu phân số vừa nhập có mẫu số = 0 thì yêu cầu nhập lại.
 - + Hàm hiển thị phân số theo dạng `tu / mau` hoặc `-tu/mau`.
 Yêu cầu: nếu tử số =0 thì chỉ in ra số 0, nếu mẫu số =1 thì chỉ in ra tử số.
 - + Hàm nghịch đảo phân số (làm thay đổi giá trị phân số) `void nghichDao();`
 - + Hàm tính ra phân số nghịch đảo của 1 phân số (phân số sẽ giữ nguyên nhưng hàm trả ra giá trị là phân số nghịch đảo của nó). `PhanSo giaTriNghichDao();`
 - + Hàm tính ra giá trị thực của phân số. Chẳng hạn phân số 1/2 có giá trị là 0.5
 - + Hàm so sánh lớn hơn với phân số `a` `boolean lonHon(PhanSo a);`
 - + Hàm cộng, trừ, nhân, chia phân số với 1 phân số `a`. Kết quả của hàm là 1 phân số.
 Chẳng hạn: `PhanSo cong(PhanSo a); ...`
 - + Hàm cộng, trừ, nhân, chia phân số với 1 số nguyên. Kết quả của hàm là 1 phân số.
- Viết lớp `SDPhanSo` có chứa hàm `main()` sử dụng lớp `PhanSo` :
- + Tạo phân số `a = 3/7` , `b = 4/9` . Hiển thị giá trị của chúng ra màn hình.
 - + Tạo 2 phân số `x` và `y` . Nhập giá trị cho `x` và `y` từ bàn phím.
 - + Hiển thị giá trị nghịch đảo của phân số `x` ra màn hình (không làm thay đổi giá trị của `x`).
 - + Tính tổng của `x + y` và in kết quả ra màn hình.
 - + Nhập vào 1 danh sách gồm `n` phân số (`n` : nhập từ bàn phím).
 - + Tính tổng `n` phân số đó.
 - + Tìm phân số lớn nhất trong danh sách phân số trên.
 - + Sắp xếp danh sách phân số theo thứ tự tăng dần.

3. Buổi 3:

○ **Mục đích:**

Sinh viên tiếp tục thực hành cài đặt lớp phức tạp hơn.
Định nghĩa hàm xây dựng sao chép, sao chép sâu đối tượng, ...

○ **Yêu cầu:**

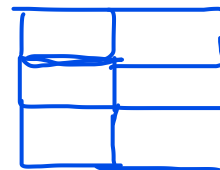
Bài 1: Cài đặt lớp **Gach** (gạch lót nền) như sau:

Thuộc tính:

- | | |
|---|-------------|
| - mã số: | kiểu String |
| - màu: | kiểu String |
| - số lượng viên trong 1 hộp: | kiểu int |
| - chiều dài viên gạch (tính theo cm): | kiểu int |
| - chiều ngang viên gạch (tính theo cm): | kiểu int |
| - giá bán 1 hộp: | kiểu long |

Phương thức:

- + Các hàm xây dựng (mặc nhiên, nhiều đối số, sao chép).
- + Hàm nhập thông tin cho 1 hộp gạch.
- + Hàm hiển thị thông tin của 1 hộp gạch.
- + Hàm tính ra giá bán lẻ 1 viên gạch: float giaBanLe();
Biết rằng: giá bán lẻ sẽ cao hơn bán nguyên hộp là 20%.
- + Hàm tính ra diện tích nền tối đa có thể lót được của hộp gạch.
- + Hàm tính ra số lượng hộp gạch ít nhất cần có khi lót 1 nền có diện tích là D*N
int soLuongHop(int D, int N)



Quy định là lót đúng chiều gạch, không cho xoay viên gạch.

Viết lớp SDGach có chứa hàm main() thực hiện các công việc sau:

- + Nhập 1 danh sách gồm n loại gạch lót nền (với n được nhập từ bàn phím).
- + Hiện ra màn hình thông tin các loại gạch vừa nhập.
- + Hiện thị ra màn hình loại gạch có chi phí lót thấp nhất (giá tiền / đơn vị diện tích).
- + Tính ra chi phí mua gạch khi ta lót 1 diện tích có chiều ngang là 500cm và chiều dài là 1500cm khi ta dùng từng loại gạch trong danh sách trên.

Bài 2: Cài đặt lớp **DoanThang** (đoạn thẳng) gồm :

- + Thuộc tính:
d1, d2 là 2 điểm đầu mút của đoạn thẳng; là đối tượng thuộc lớp Diem (đã có sẵn).
- + Các hàm xây dựng:
DoanThang();
DoanThang(Diem, Diem);
DoanThang(int ax, int ay, int bx, int by);
- + Các phương thức:
 - Nhập tọa độ của đoạn thẳng.
 - Hiện thị giá trị 2 đầu mút của đoạn thẳng.
 - Tịnh tiến đoạn thẳng đi độ dời (dx, dy) nào đó.
 - Tính độ dài của đoạn thẳng.

Viết lớp SDDoanThang có chứa hàm main thực hiện các công việc sau :

- + Tạo 2 điểm là A(2, 5), B(20, 35). Tạo đoạn thẳng AB. Tịnh tiến AB đi đoạn (5,3) .
- + Tạo một đoạn thẳng bất kỳ CD . Nhập giá trị cho đoạn thẳng CD đó .
- + Hiện thị ra màn hình độ dài CD.

Bài 3: Cài đặt lớp **SinhVien** (sinh viên) gồm :

Thuộc tính:

- | | |
|--------------------------------|------------------|
| - mã số sinh viên: | kiểu String |
| - họ tên: | kiểu String |
| - ngày sinh: | kiểu Date |
| - số lượng học phần đăng ký: | kiểu số nguyên |
| - tên các học phần đã đăng ký: | mảng kiểu String |

- điểm của các học phần: mảng kiểu String (điểm tính theo A, B+, B, ...)

Phương thức

- + Các hàm xây dựng
- + Hàm nhập thông tin cơ bản của sinh viên
- + Hàm nhập điểm cho các học phần của sinh viên
- + Hàm toString(): xuất ra chuỗi là thông tin của sinh viên
- + Hàm tính ra điểm trung bình của sinh viên theo thang điểm 4.
- + Hàm đăng ký thêm 1 học phần cho sinh viên
- + Hàm xóa 1 học phần của sinh viên

Viết lớp SDSinhVien có chứa hàm main thực hiện các công việc sau :

- + Tạo sinh viên a. Nhập thông tin cho sinh viên a. Đăng ký thêm cho sinh viên a 1 học phần là “LTHDT”. Hiển thị thông tin của a.
- + Tạo 1 danh sách sinh viên. Nhập thông tin cho danh sách sinh viên trên.
- + Tìm các sinh viên bị cảnh báo học vụ.
- + Tìm sinh viên có điểm trung bình cao nhất lớp.
- + Hiển thị danh sách sinh viên trên theo thứ tự Alphabet của Tên.

4. Buổi 4:

o Mục đích:

Sinh viên thực hành cách cài đặt thừa kế trong Java.

Thử nghiệm cách nạp đè phương thức (method overriding), liên kết động và tính đa hình.

o Yêu cầu:

Bài 1: Cài đặt lớp **DiemMau** (Điểm có màu) thừa kế từ lớp Diem (định nghĩa trong buổi 1) bổ sung thêm

Thuộc tính: màu (kiểu String)

Các phương thức:

- + Hàm xây dựng: DiemMau(int x, int y, String mau)
- + Hàm gán giá trị màu cho điểm: void GanMau(String mau)
- + Hàm nhập, hàm hiển thị thông tin
- + ...

Viết lớp SDDiemMau có hàm main() thực hiện các công việc sau:

- + Tạo 1 điểm màu A có tọa độ là (5, 10) và màu là trắng. Hiển thị thông tin ra màn hình.
- + Tạo 1 điểm màu tổng quát B. Nhập giá trị từ bàn phím cho điểm B. Dời điểm B đi 1 độ dời (10,8). Hiển thị tọa độ điểm B ra màn hình. Gán màu mới cho điểm B là màu “Vàng”.

Bài 2: Cài đặt lớp **SinhVienCNTT** (Sinh viên công nghệ thông tin) thừa kế từ lớp SinhVien (đã định nghĩa trong buổi trước) như sau:

Thuộc tính:

- taikhoan: kiểu String (tài khoản sử dụng trên hệ thống ELCIT)
- matkhau: kiểu String (mật khẩu ELCIT)
- email: kiểu String (email của sinh viên)

Phương thức:

- + Các hàm xây dựng
- + Nhập thông tin
- + Nạp đè phương thức toString() để xuất ra thông tin của 1 sinh viên CNTT
- + Đổi mật khẩu: void doiMatKhau(String newpass);
- + Lấy ra địa chỉ email của sinh viên: String getEmail();

Viết lớp SDSVCNTT có hàm main() thực hiện các công việc sau:

- + Tạo 1 danh sách gồm n sinh viên CNTT. Nhập thông tin cho danh sách đó.
- + Nhập vào 1 địa chỉ email. Tìm tài khoản ELCIT của sinh viên có địa chỉ email trên. Hiển thị kết quả học tập của sinh viên đó.

Bài 3: Một nông trại có nuôi một số các con vật như sau: bò, heo, dê.

Tất cả các con vật trên đều có những thông tin chung như : giống, màu lông, cân nặng, ... nhưng tiếng kêu của các con vật là khác nhau.

Thiết kế sơ đồ thừa kế gồm các lớp:

Lớp ConVat gồm các thông tin chung của các con vật nêu trên và phương thức Kêu().

Các lớp Bo, Heo, De thừa kế từ lớp ConVat và nạp đè phương thức Kêu().

Viết lớp chứa hàm main() thực hiện các công việc sau:

- + Tạo ra n con vật bất kỳ gồm cả 3 loài vật trên(có thể cho ngẫu nhiên).
- + Nhập thông tin cho các con vật.
- + Cho n con vật đó kêu.

5. Buổi 5:

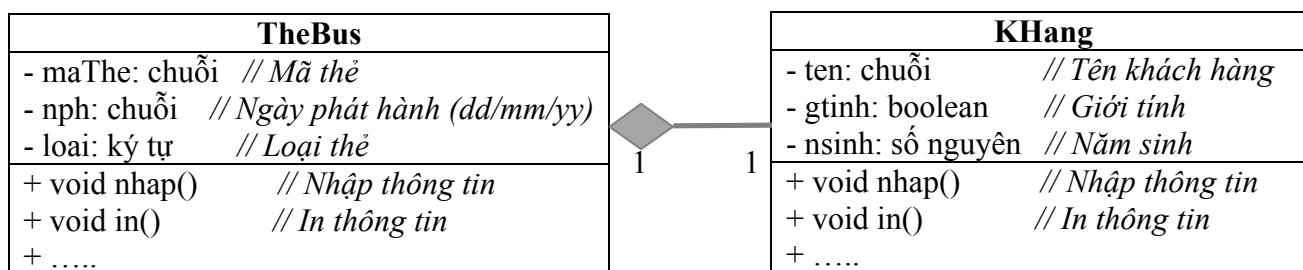
○ Mục đích:

Sinh viên hiểu cách thức mô hình hoá một bài toán

Sinh viên tiến hành lập trình hướng đối tượng với Java cho bài toán đó

○ Yêu cầu:

Sinh viên khai báo các lớp và định nghĩa các phương thức phù hợp cho bài toán quản lý thẻ xe trong dịch vụ giao thông công cộng. Một thẻ bus thông minh thuộc về một khách hàng sử dụng khi di chuyển với hệ thống xe buýt. Loại thẻ 'H' có giá trị trong 1 giờ kể từ lần xác thực đầu tiên, 'D' có giá trị trong 24 giờ, 'M' có giá trị trong 1 tháng, 'Y' có giá trị trong 1 năm.



Sinh viên viết hàm main để thực thi chương trình quản lý.

+ Nhập và hiển thị một danh sách ds1 có chứa m đối tượng thẻ buýt, với m được nhập từ bàn phím.

+ Hãy liệt kê tất cả thẻ xe buýt loại có giá trị sử dụng 1 năm.

+ Hãy liệt kê tất cả khách hàng có thẻ xe được phát hành trong ngày 20/05/22.