**BÀI TẬP LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG CĐ2021**

**BÀI TẬP THỰC HÀNH SỐ 1**

1. **Thông tin chung:**

* Mã số bài tập : HW1-LTHDT
* Hình thức nộp bài : Nộp qua hệ thống EL môn học
* Thời hạn nộp bài : … / … / ……
* Nội dung : Chương 1: Giới thiệu lập trình hướng đối tượng

**Chuẩn đầu ra cần đạt:**

|  |  |
| --- | --- |
| L.O.1 | Phân tích, thiết kế chương trình theo hướng đối tượng. Cài đặt được sơ đồ lớp cho các chương trình vừa và nhỏ; |
| L.O.8 | Làm bài tập và nộp bài đúng quy định. |

**Sinh viên thực hành các yêu cầu sau:**

**BÀI TẬP CƠ BẢN**

1. **Sinh viên trả lời các câu hỏi sau**
   1. Giải thích sự khác nhau của phương pháp lập trình hướng đối tượng và phương pháp lập trình hướng thủ tục
   2. Đối tượng là gì? Lớp là gì? Phân biệt lớp và đối tượng
   3. Nêu các thành phần của một lớp và cách xác định chúng.
2. **Hãy xác định thuộc tính và hành vi (phương thức) của các lớp đối tượng**
3. Sinh viên (Student)
4. Lớp học (Class)
5. Cuốn sách (Book)
6. Nhân viên (Employee)
7. Tài khoản ngân hàng (Bank Account)
8. Điện thoại di động (Mobile)
9. Thời gian (Time)
10. Phân số (Fraction)
11. Xe hơi (Car)
12. Điểm trên trục tọa độ (Point)
13. **Hãy vẽ sơ đồ lớp của các lớp đối tượng trong Bài 2.**

**BÀI TẬP THỰC HÀNH SỐ 2**

1. **Thông tin chung:**

* Mã số bài tập : HW1-LTHĐT
* Hình thức nộp bài : Nộp qua hệ thống EL môn học
* Thời hạn nộp bài : … / … / ……
* Nội dung : Chương 2: Xây dựng lớp

**Chuẩn đầu ra cần đạt:**

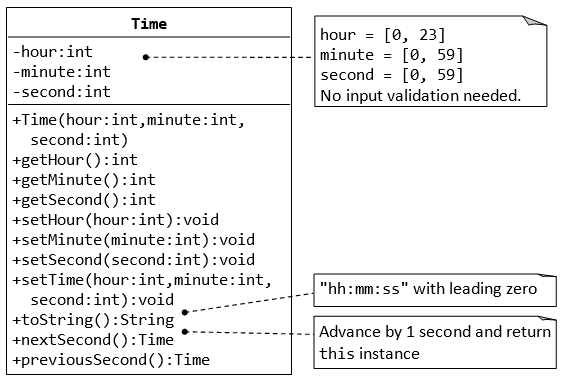
|  |  |
| --- | --- |
| L.O.1 | Phân tích, thiết kế chương trình theo hướng đối tượng. Cài đặt được sơ đồ lớp cho các chương trình vừa và nhỏ; |
| L.O.8 | Làm bài tập và nộp bài đúng quy định. |

**BÀI TẬP CƠ BẢN**

1. Sinh viên trả lời các câu hỏi sau
2. Giải thích sự khác nhau giữa bổ từ truy xuất (Access modifier)
3. Hãy phân biệt class và static class; class và partial class.
4. Nêu ý nghĩa thuộc tính read only, write only và read-write.
5. Có thể có bao nhiêu constructor và destructor trong một lớp? Giải thích sự khác nhau giữa chúng?
6. Nêu mục đích của các phương thức set, get.
7. Nêu tác dụng của từ khoá static trên các thành viên của lớp
8. Nêu các mối quan hệ không có tính kế thừa giữa hai class. Cho ví dụ minh hoạ.

Với mỗi bài toán sau hãy vẽ sơ đồ lớp sau đó cài đặt sơ đồ lớp đó.

Viết chương trình cho phép tạo ra các đối tượng Time (thời gian) được mô tả theo sơ đồ lớp sau:



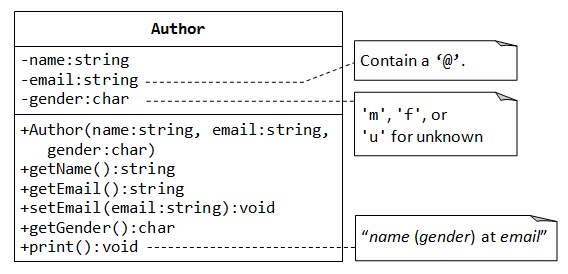
**Mô tả lớp Time:**

* Lớp Time bao gồm dữ liệu hour, minute, second với các giá trị hour hợp lệ trong khoảng từ 0 đến 23, minute và second trong khoảng từ 0 đến 59 và quyền truy cập và private.
* Lớp Time chỉ hỗ trợ một phương thức tạo có đầy đủ tham số và các hỗ trợ đầy đủ các thuộc tính get, set cho các trường.
* Phương thức toString() trả về chuỗi theo định dạng hh:mm:ss.
* Phương thức NextSecond(): trả về đối tượng Time là một giây sau.
* Phương thức PreviousSecond(): trả về đối tượng Time là một giây trước.

Viết đoạn chương trình theo yêu cầu sau:

* cho phép người dùng tạo đối tượng Time với các dữ liệu do người dùng nhập vào từ bàn phím. Nếu dữ liệu sai báo lỗi và không cho phép tạo đối tượng. Gọi phương thức toString() để in chuỗi theo định dạng ra màn hình.
* Thực hiện gán lại giá trị cho các trị cho tất cả các trường của đối tượng vừa tạo, trường hợp giá trị mới không hợp lệ thì thông báo lỗi. In giá trị mới của đối tượng ra màn hình.
* Gọi phương thức PreviousSecond() và in kết quả ra màn hình.
* Gọi phương thức NextSecond() và in kết quả ra màn hình.

Thực hiện khai báo class Author (tác giả) được mô tả theo sơ đồ lớp sau:



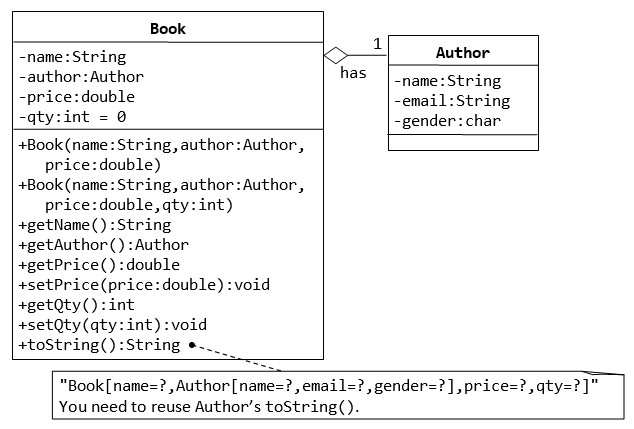
Mô tả lớp Author:

* Lớp Author gồm các trường name, email, gender. Trường email bắt buộc phải có ký tự '‘@'’ trường gender (phái) chỉ được phép chứa giá trị là một trong các ký tự m (male - nam), f (femail - nữ) hoặc u (unknown ).
* Phương thức tạo cho phép truyền đầy đủ tham số hoặc chỉ có thông tin về name, email.
* Các trường name, gender không hỗ trợ các hành động gán dữ liệu từ bên ngoài lớp. Người dùng có quyền xem và gán lại giá trị cho email.
* Phương thức Print() in chuỗi thông tin theo định dạng ra màn hình.

Viết đoạn chương trình cho phép tạo ra các đối tượng Author với các đầy đủ tham số và đối tượng chỉ có hai tham số là name và gender. Thử nghiệm tất cả thuộc tính get, set các trường trên các đối tượng vừa tạo. Gọi phương thức Print() để in thông tin của các đối tượng ra màn hình.

Viết đoạn chương trình cho phép nhập một mảng các đối tượng kiểu Author. Thực hiện tìm kiếm một đối tượng bằng cách nhập vào name từ bàn phím, in ra thông tin của đối tượng tìm thấy.

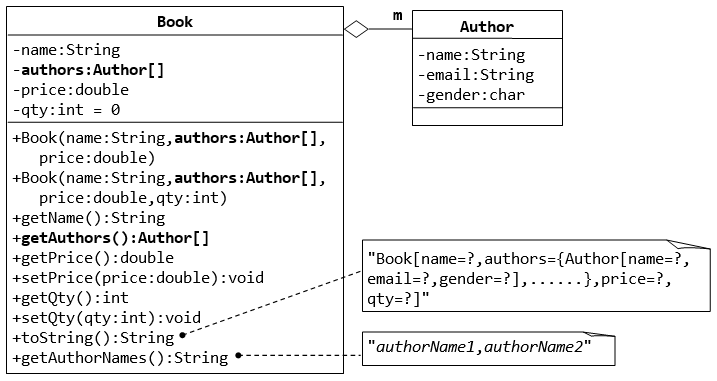
1. Hãy cài đặt viết chương trình sử dụng thử nghiệm lớp sau:



Viết đoạn chương trình cho phép tạo ra các đối tượng Book với các đầy đủ tham số. Thử nghiệm tất cả thuộc tính get, set các trường trên các đối tượng vừa tạo. Gọi phương thức toString() để in thông tin của các đối tượng ra màn hình.

Viết đoạn chương trình cho phép nhập một mảng các đối tượng kiểu Book. Thực hiện tìm kiếm một đối tượng bằng cách nhập vào name từ bàn phím, in ra thông tin của đối tượng tìm thấy.

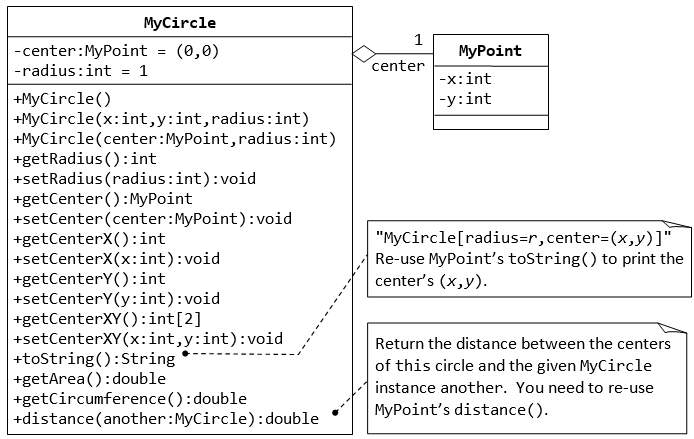
1. Hãy cài đặt viết chương trình sử dụng thử nghiệm lớp sau



Viết đoạn chương trình cho phép tạo ra các đối tượng Book với các đầy đủ tham số. Thử nghiệm tất cả thuộc tính get, set các trường trên các đối tượng vừa tạo. Gọi phương thức toString() để in thông tin của các đối tượng ra màn hình.

Viết đoạn chương trình cho phép nhập một mảng các đối tượng kiểu Book. Thực hiện tìm kiếm một đối tượng bằng cách nhập vào name từ bàn phím, in ra thông tin của đối tượng tìm thấy.

1. Hãy cài đặt viết chương trình sử dụng thử nghiệm lớp sau



Viết đoạn chương trình cho phép nhập một mảng các đối tượng kiểu MyCircle Thực hiện in ra thông tin của đối tượng thỏa điều kiện diện tích >=100.

**BÀI TẬP NÂNG CAO**

1. Viết chương trình quản lý sinh viên. Biết rằng mỗi sinh viên cần lưu trữ các thuộc tính sau : họ tên , lớp, điểm trung bình tích lũy.

Hãy thiết kế lớp SinhVien với những thuộc tính đã mô tả và các phương thức sau:

1. Khởi tạo và huỷ đối tượng
2. Xếp loại cho sinh viên biết :
   * Nếu đtb>8.5🡪Giỏi
   * Nếu đtb>=7🡪 Khá
   * Nếu đtb>=5🡪 Trung Bình
   * Còn lại Yếu
3. Viết chương trình chính cho phép tạo ra các đối tượng của lớp SinhVien và gọi các chức năng của đối tượng.
4. Viết chương trình để quản lý danh sách các máy tính của một cửa hàng. Biết rằng thông tin của một máy tính bao gồm: Loại máy**,** Nơi sản xuất **,** Thời gian bảo hành
5. Xây dựng lớp MayTinh với các thuộc tính đã cho và một số phương thức sau: Khởi tạo, hủy, xuất thông tin, phương thức đếm số lượng đối tượng máy tính có trong hệ thống
6. Viết chương trình chính cho phép tạo ra danh sách gồm n máy tính của cửa hàng. Sau đó thực hiện các chức năng sau:
7. Viết phương thức thống kê xem có bao nhiêu máy có thời gian bảo hành là 1 năm.
8. In ra danh sách các máy tính có xuất xứ từ Mỹ
9. In ra số lượng đố tượng máy tính có trong hệ thống

**BÀI TẬP THỰC HÀNH SỐ 3**

1. **Thông tin chung:**

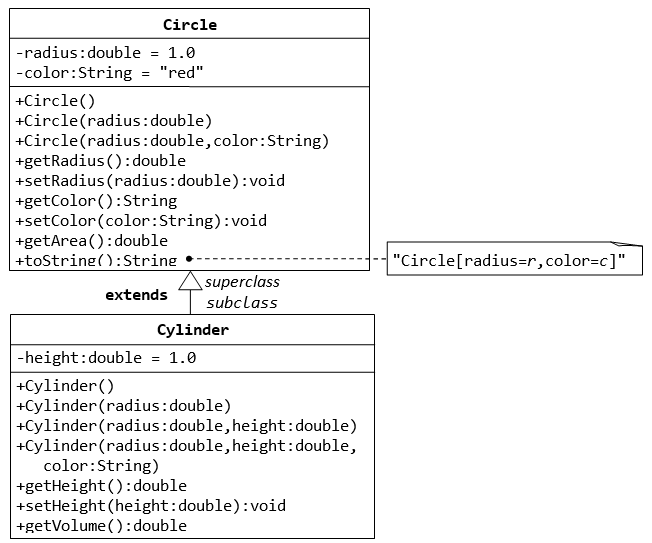
* Mã số bài tập : HW2-LTHĐT
* Hình thức nộp bài : Nộp qua hệ thống EL của môn học
* Thời hạn nộp bài : … / … / ……
* Nội dung : Chương 3. Tính Kế thừa

**Chuẩn đầu ra cần đạt:**

|  |  |
| --- | --- |
| L.O.2 | Sử dụng cơ chế kế thừa và đa hình trong thiết kế hướng đối tượng |
| L.O.8 | Làm bài tập và nộp bài đúng quy định; |

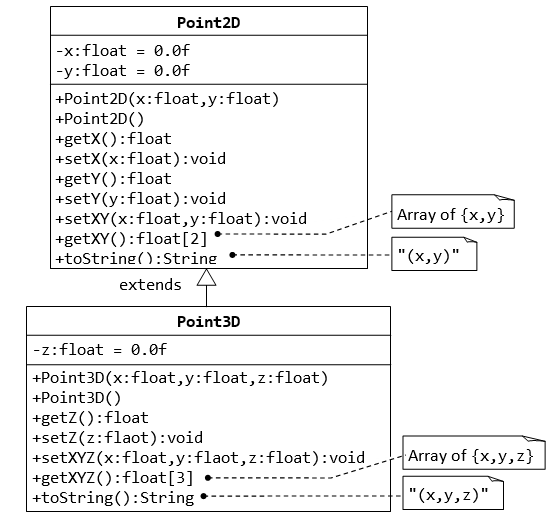
**BÀI TẬP CƠ BẢN**

1. **Hãy cài đặt các lớp sau**



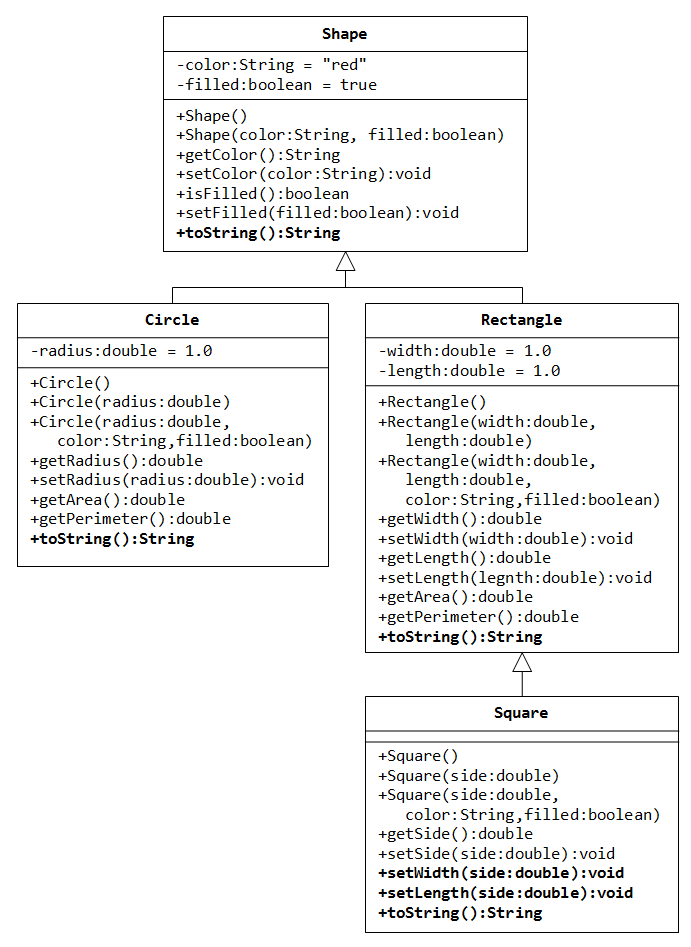
Thực hiện viết chương trình chạy thử nghiệm khởi tạo các đối tượng/ danh sách đối tượng và gọi sử dụng tất cả các phương thức của các lớp trên

1. **Hãy cài đặt các lớp sau :**



Thực hiện viết chương trình chạy thử nghiệm khởi tạo các đối tượng/ danh sách đối tượng và gọi sử dụng tất cả các phương thức của các lớp trên

1. **Hãy cài đặt các lớp sau:**



Thực hiện viết chương trình chạy thử nghiệm khởi tạo các đối tượng/ danh sách đối tượng và gọi sử dụng tất cả các phương thức của các lớp trên

1. (HAS-A, IS-A)

Giả sử công ty có 2 loại nhân viên: Nhân viên văn phòng và Nhân viên sản xuất. Viết chương trình quản lý và tính lương cho từng nhân viên của công ty. Mỗi nhân viên cần quản lý các thông tin sau: Họ tên, ngày sinh, địa chỉ. Trong đó **địa chỉ** là 1 đối tượng, cần lưu trữ thông tin cụ thể: *số nhà, tên đường, tên quận, thành phố.*

Công ty cần tính lương cho nhân viên như sau:

* + Đối với nhân viên sản xuất: Lương = lương căn bản + số sản phẩm \* 5000

Đối với nhân viên văn phòng: Lương = số ngày làm việc \* 100000

|  |
| --- |
| **DiaChi** |
| -soNha:string  -tenDuong: string  -tenQuan: string  -thanhPho:string |
|  |

Tại chương trình chính hãy tạo ra 1 danh sách gồm các Nhân viên sản xuất và 1 danh sách gồm các nhân viên văn phòng. Gọi các chức năng in thông tin nhân viên, xem lương của từng nhân viên.

Gợi ý sơ đồ lớp:

|  |
| --- |
| **NhanVien** |
| -hoTen: string  -ngaySinh:DateTime  -diaChi: DiaChi |
| +NhanVien(hoTen:string, ngaySinh:DateTime, diaChi:DiaChi)  +InThongTin():void |

Has-a

|  |
| --- |
| **NhanVienVanPhong** |
| -soNgayLamViec:int |
| + NhanVienVanPhong( hoTen:string, ngaySinh:DateTime, diaChi:DiaChi, soNgayLamViec:int)  +InThongTin():void  +TinhLuong():double |

|  |
| --- |
| **NhanVienSanXuat** |
| -luongCB:double  -soSP: int |
| +NhanVienSanXuat( hoTen:string, ngaySinh:DateTime, diaChi:DiaChi, luongCB:double, soSP:int)  +InThongTin():void  +TinhLuong():double |

Is-a

Is-a

**BÀI TẬP NÂNG CAO**

**Bài 5**. (HAS-N) Phòng đào tạo của Trường đại học ABC cần xây dựng ứng dụng quản lý sinh viên để dễ dàng theo dõi tình hình học tập và rèn luyện của sinh viên trong trường. Biết rằng mỗi sinh viên được mô tả bởi m*ã sinh viên, họ tên, năm sinh và danh sách các môn học mà sinh viên đã tích lũy đến thời điểm hiện tại*. Biết rằng với mỗi môn học cần được lưu trữ thông tin: *mã môn học, tên môn học, số tín chỉ và điểm môn học.*

Hãy xây dựng các lớp cần thiết cho ứng dụng. Với lớp sinh viên cần xây dựng các chức năng sau:

1. Nhập dữ liệu cho sinh viên
2. Xem thông tin sinh viên (ngoài những thông tin được mô tả, cần xuất thêm thông tin về điểm trung bình tích lũy, xếp loại học tập)

Mô phỏng mô hình khi xuất dữ liệu:

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**MaSV: sv01**

**Ho ten: Nguyen Van An**

**Nam sinh: 1996**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Danh sach gom 2 mon hoc:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| * **STT** | * **Ma Mon** | * **Ten Mon** | * **So tin chi** | * **Diem** |
| * **1** | * **Csc1060001** | * **Nhap mon** | * **3** | * **5** |
| * **2** | * **Csc1070001** | * **KTLT** | * **3** | * **5** |

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Diem trung binh tich luy: 5.0**

**Xep loai: Trung binh**

1. Tính điểm trung bình tích lũy (TBTL) theo công thức:

TBTL = (điểm môn 1. \* tín chỉ môn 1 + điểm môn 2 \* tín chỉ môn 2 + …)/(tín chỉ 1 + tín chỉ 2 +…)

1. Xếp loại học tập sinh viên. Biết rằng
   * + - 10 > TBTL >= 9 : Xuất sắc
       - 9> TBTL >= 8: Giỏi
       - 8 > TBTL >= 7: Khá
       - 7 > TBTL >=5: Trung bình
       - TBTL < 5: Yếu
2. Xây dựng thêm lớp DanhSachSinhVien. Biết rằng mỗi danh sách cần lưu trữ mảng sinh viên và số lượng sinh viên thực tế. Các phương thức cần xây dựng cho lớp này bao gồm: Nhập danh sách, xuất danh sách, tìm sinh viên có điểm trung bình cao nhất. Sau đó tại chương trình chính cho phép tạo ra 1 danh sách sinh viên (L) của trường (đối tượng của lớp DanhSachSinhVien) và gọi các chức năng như sau:

L.NhapDanhSach();

L.XuatDanhSach();

L.TinhDiemTBCaoNhat();

**BÀI TẬP THỰC HÀNH SỐ 4**

1. **Thông tin chung:**

* Mã số bài tập : HW2-LTHĐT
* Hình thức nộp bài : Nộp qua hệ thống EL của môn học
* Thời hạn nộp bài : … / … / ……
* Nội dung : Chương 4. Kế thừa – Đa hình

**Chuẩn đầu ra cần đạt:**

|  |  |
| --- | --- |
| L.O.2 | Sử dụng cơ chế kế thừa và đa hình trong thiết kế hướng đối tượng |
| L.O.8 | Làm bài tập và nộp bài đúng quy định; |

**BÀI TẬP CƠ BẢN:**

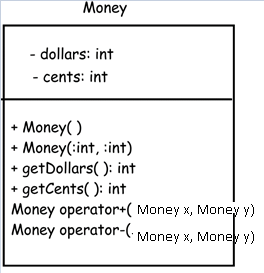
**Hãy cài đặt các lớp sau:**

1. Trả lời các câu hỏi sau:
2. Điểm khác nhau giữa Composition và lớp kế thừa Inheritance là gì?
3. Giải thích sự khác nhau giữa các thành phần protected và thành phần private?
4. Các Constructor và Destructor hoạt động như thế nào trong Derived class?
5. So sánh Overloading methods và Overriding methods.
6. Nêu các nguyên tắc khi sử dụng virtual methods.
7. So sánh static binding và dynamic binding. Cho vi dụ minh hoạ.

**Bài 2.** Tạo một lớp Phân số gồm có đặc tính tử số và mẫu số và các phương thức tiện ích như khởi tạo, in thông tin, rút gọn phân số và các phương thức nạp chồng toán tử sau :

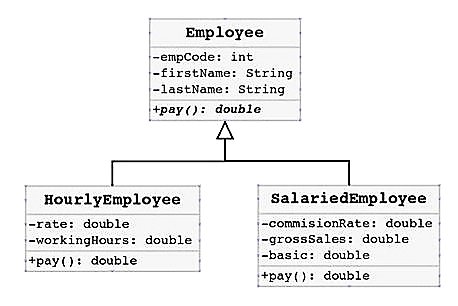
* 1. Các toán tử hai ngôi : **+ , - , \*, /**
  2. Các toán tử so sánh == , > , >=, <, <=, !=
  3. Toán tử một ngôi: -

**Bài 3.** **Cài đặt sơ đồ lớp sau**



Thực hiện viết chương trình chạy thử nghiệm một vài đối tượng Money và gọi sử dụng tất cả các phương thức của lớp

**Bài 4.** **Cài đặt sơ đồ lớp sau :**



Thiết kế 3 lớp trong sơ đồ trên. Mỗi lớp xây dựng phương thức khởi tạo, toString().

Lương của nhân viên được tính như sau:

* + HourlyEmployee::pay(): *rate \* workingHours*
  + SalerieEmployee::pay() : *basic +comissionRate \* grossSales*

Thực hiện viết chương trình cho phép:

- Tạo ra một vài đối tượng chạy thử nghiệm tính đa hình động.

- Tạo ra ***một*** danh sách chứa tất cả các loại nhân viên (HourlyEmployee và SalariedEmployee). Sau đó viết và thực thi các chức năng sau:

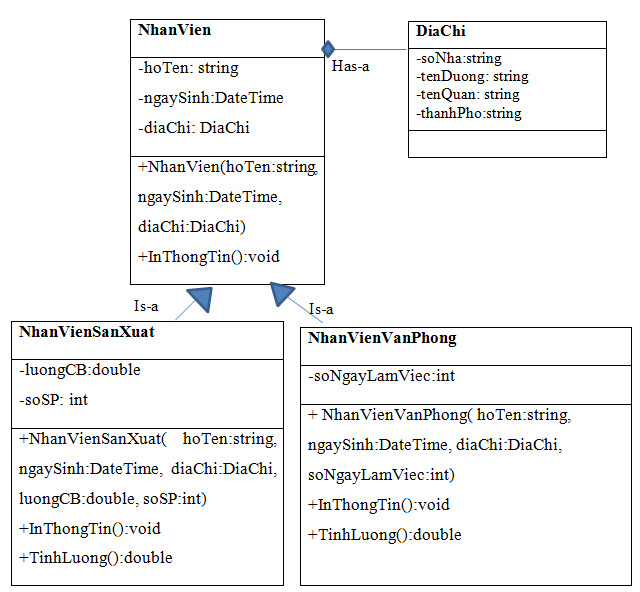
a. Nhập danh sach n nhân viên

b. Xuất danh sách

c. Tính tổng lương của tất cả các nhân viên trong công ty

* 1. Tính tổng số từng loại
  2. Liệt kê toàn bộ thông tin nhân viên HourEmployee

***Bài 5.*** Cài đặt sơ đồ lớp sau



Trong đó lương nhân viên được tính như sau:

* + Đối với nhân viên sản xuất: Lương = lương căn bản + số sản phẩm \* 5000
  + Đối với nhân viên văn phòng: Lương = số ngày làm việc \* 100000

Viết chương trình chính cho phép tạo ra **1 danh sách** chứa tất cả nhân viên trong công ty (bao gồm Nhân viên sản xuất và Nhân viên văn phòng).

Các chức năng chính của chương trình:

* 1. Nhập danh sách n nhân viên công ty
  2. In toàn bộ thông tin của từng nhân viên trong công ty (in cả thông tin lương)
  3. Tính tổng lương của toàn bộ nhân viên.
  4. Liệt kê nhân viên đến tuổi về hưu, biết rằng nhân viên có tuổi là 60 sẽ về hưu.
  5. Hàm thống kê tổng số lượng nhân viên sản xuất và lọc ra chi tiết thông tin những nhân viên là nhân viên sản xuất

**BÀI TẬP NÂNG CAO:**

**Bài 6**. **Cài đặt sơ đồ lớp sau :**



Thực hiện viết chương trình cho phép tạo ra 1 danh sách chứa các loại hình (bao gồm Circle, Rectangle, Square).

Các chức năng của chương trình chính gồm:

1. Nhập danh sách
2. Xuất thông tin các hình ra màn hình (gọi toString() để in thông tin, in thêm chu vi và diện tích của từng hình)
3. Thống kê tổng số lượng từng loại hình.

**BÀI THỰC HÀNH SỐ 5**

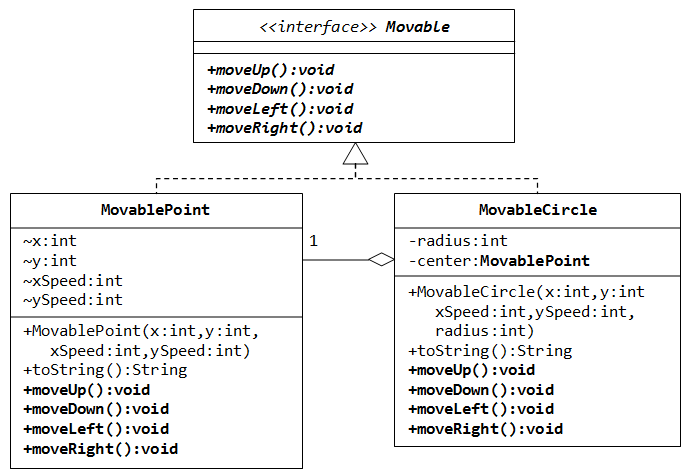
1. **Thông tin chung:**

* Mã số bài tập : HW3 - LTHĐT
* Hình thức nộp bài : Nộp qua Moodle môn học
* Thời hạn nộp bài : … / … / ……
* Nội dung : Chương 5. Lớp giao diện
* **Chuẩn đầu ra cần đạt:**

|  |  |
| --- | --- |
| L.O.3 | Sử dụng giao diện trong thiết kế hướng đối tượng |
| L.O.8 | Làm bài tập và nộp bài đúng quy định; |

**BÀI TẬP CƠ BẢN:**

1. Hãy cài đặt sơ đồ lớp sau:



Trong đó:

xSpeed và ySpeed là một khoảng toạ độ và mỗi lần dịch chuyển tương ứng có thể đi được.

MoveablePoint: x, y, xSpeed, ySpeed;

moveUp: y = y + ySpeed

moveDown: y = y – ySpeed

moveLeft: x = x – xSpeed

moveRigth: x = x + xSpeed

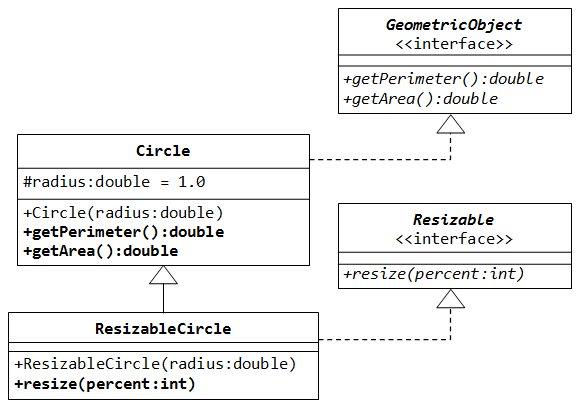
Mỗi lần MoveCircle thực hiện moveUp, hay moveRight,… tất chính là move center.

**Yêu cầu**: Cài đặt sơ đồ lớp trên, sau đó viết chương trình chạy thử nghiệm:

-Tạo danh sách các đối tượng MovableCircle

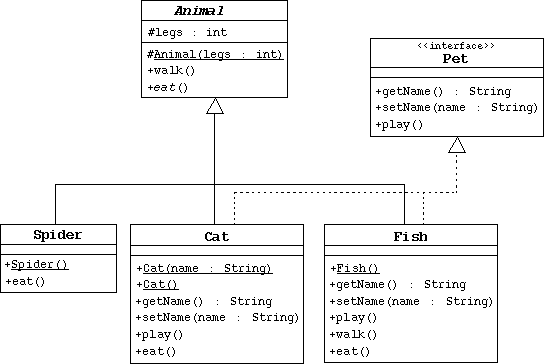
-Gọi sử dụng tất cả các phương thức của MovableCircle

1. Cài đặt sơ đồ sau:



Thực hiện viết chương trình chạy thử nghiệm, tạo đối tượng 2 lớp Circle và ResizeableCircle

1. Cài đặt sơ đồ sau:



Thay thế bằng property

* Thiết kế sơ đồ lớp với các gơi ý sau:
  + Tạo lớp Animal là một Abstract class chứa abstract method eat(). Phương thức walk() in ra trạng thái về di chuyển có bao gồm số chân (legs). Ví dụ : “Đi bằng 4 chân”.
  + Tạo lớp Spider class kế thừa từ Animal. Định nghĩa phương thức tạo với tất cả Spider đi bằng 8 chân.
  + Tạo Pet interface theo sơ đồ.
  + Tạo lớp Cat kết thừa từ Animal và triển khai Pet.
* Tại chương trình chính hãy tạo ra một danh sách chứa các động vật (Spider, Cat và Fish). Sau đó viết và thực thi các chức năng sau:

Nhập danh sách

Hàm yêu cầu hành động eat() cho tất cả các động vật có trong danh sách, riêng với đối tượng thuộc lớp Cat và Fish thì yêu cầu thêm hoạt động play().

**BÀI TẬP NÂNG CAO**

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

* Đất trồng: giá trị = diện tích \* 30,000,000;
* Nhà ở có thêm thông tin về số lầu: giá trị = diện tích \* 60,000,000 + số lầu \* 100,000,000.
* Khách sạn có thêm thông tin về số sao: giá trị = diện tích \* 70,000,000 + số sao \* 50,000,000
* Biệt thự: giá trị = diện tích \* 100,000,000

Trong 4 loại bất động sản trên thì có 2 loại là biệt thự và khách sạn phải đóng phí kinh doanh. Phí kinh doanh được tính như sau:

* Biệt thự: chiều rộng \* 5000
* Khách sạn: chiều rộng \* 10000

**Yêu cầu:**

1. Thiết kế các lớp BatDongSan (lớp cha) và các lớp con: DatTrong, NhaO, KhachSan, BietThu với các thuộc tính và chức năng như mô tả. Thiết kế giao diện PhiKinhDoanh chứa một chức năng tính phí kinh doanh. Sau đó 2 lớp BietThu và KhachSan sẽ triển khai giao diện PhiKinhDoanh.

2.Tại chương trình chính xây dựng các chức năng sau:

1. Nhập 1 danh sách bất động sản
2. Cho biết số lượng bất động sản theo từng loại
3. In ra danh sách các bất động sản có diện tích trên 1000
4. Tính tổng phí dinh doanh thu được

**Bài 5.** Hàng hóa quản lý trong kho của một siêu thị gồm có hàng thực phẩm, hàng sành sứ và hàng điện máy. Mỗi loại hàng đều có mã hàng (không được sửa, không được để trống), tên hàng (không được rỗng), số lượng tồn (>=0), đơn giá (>0).

* Hàng thực phẩm thì cần quan tâm đến thông tin ngày sản xuất, ngày hết hạn (ngày hết hạn phải sau hoặc là ngày sản xuất và ngày là đối tượng lớp MyDateTime) và nhà cung cấp.

**Lưu ý**: Lớp MyDateTime sẽ có 3 field: day, month, year. Lớp này sẽ implement giao diện Icomparable của thư viện, cung cấp chức năng

int CompareTo (object obj);

* + - <0: đối tượng hiện hành < đối tượng truyền vào
    - > đối tượng hiện hành > đối tượng truyền vào
    - = đối tượng hiện hành = đối tượng truyền vào
* Hàng điện máy cần biết thời gian bảo hành bao nhiêu tháng (>=0), công suất bao nhiêu KW (>0).
* Hàng sành sứ thì cần biết thông tin về nhà sản xuất và ngày nhập kho.

Ngoài ra, người quản lý cần quan tâm đến số lượng tồn kho và các yếu tố khác của từng loại hàng hóa để đánh giá mức độ bán buôn, tiền VAT từng loại hàng hóa. Biết rằng VAT của hàng điện máy và sành sứ là 10%, VAT của hàng thực phẩm là 5%

Để đánh giá mức độ bán buôn thì:

+ Hàng điện máy, nếu số lượng tồn kho < 3 thì đánh giá là bán được

+ Hàng thực phẩm, nếu vẫn còn tồn kho và bị hết hạn thì đánh giá là khó bán

+ Hàng sành sứ, nếu số lượng tồn kho >50 và thời gian lưu kho >10 ngày thì đánh giá là bán chậm

+ Các trường hợp còn lại xem như không đánh giá

**Yêu cầu:** Hãy thiết kế các lớp theo mô tả trên. Sau đó tại chương trình tính hãy viết ứng dụng quản lý các hàng hóa trong kho. Các chức năng cần có.

Nhập danh sách hàng hóa

Xuất danh sách hàng hóa

Liệt kê các hàng điện máy được đánh giá là bán được

Liệt kê các hàng thực phẩm được đánh giá là khó bán

Liệt kê các hàng sứ được đánh giá là bán chậm.

**BÀI THỰC HÀNH SỐ 6**

1. **Thông tin chung:**

* Mã số bài tập : HW3 - LTHĐT
* Hình thức nộp bài : Nộp qua Moodle môn học
* Thời hạn nộp bài : … / … / ……
* Nội dung : Chương 6. Lập trình khái quát

**Chuẩn đầu ra cần đạt:**

|  |  |
| --- | --- |
| L.O.3 | Lập trình khái quát bằng phương pháp dùng template |
| L.O.8 | Làm bài tập và nộp bài đúng quy định. |

**BÀI TẬP CƠ BẢN**

1. Xây dựng generic method chứa các xử lý trên mảng một chiều sau:
   1. Tìm kiếm tuần tự các phần tử mảng
   2. In các phần tử mảng ra màn hình
   3. Thêm một phần tử vào cuối mảng
   4. Xóa một phần tử tại vị trí index
   5. Sắp xếp mảng theo thứ tự tăng dần hoặc giảm dần

Viết ứng sử dụng generic class vừa tạo ở trên với các kiểu số thực, nguyên, chuỗi, object,….

1. Xây dựng generic class (WORD) mô tả từ trong cuốn từ điển. Biết mỗi từ chứa dữ liệu gồm hai trường key (chứa từ) và value (là nghĩa của từ) , các methods sau:
   1. Khởi tạo, set, get.
   2. toán tử = = thực hiện so sánh hai đối tượng biết chúng bằng nhau khi hai trường value bằng nhau
   3. Phương thức Equals() trả về true khi hai đối tượng giống cả key và valuep

**BÀI TẬP NÂNG CAO**

1. Xây dựng khuôn hình lớp mô tả lớp danh sách các phần tử (List). Danh sách này dùng để tạo ra các đối tượng như: Danh sách số nguyên, số thực, chuỗi,...

Thuộc tính:

* Mảng phần tử (arr)
* Số phần tử tối đa (MAXSIZE)
* size

Phương thức:

* Add(): Thêm một phần tử vào cuối danh sách.
* Display(): Xuất danh sách
* Property Count: trả về kích thước của danh sách
* Remove(): Xóa một phần tử có giá trị value trong danh sách
* RemoveAt(): Xóa một phần tử tại vị trí cho trước

1. Xây dựng khuôn hình lớp Stack

Thuộc tính:

* Mảng phần tử (arr)
* Số phần tử tối đa (MAXSIZE)
* Int top;

Phương thức:

* Push(): Thêm một phần tử vào danh sách.
* Display(): Xuất danh sách
* Property Count: trả về kích thước của danh sách
* Pop(): Xóa một phần tử trong danh sách
* IsEmpty(); // kiểm tra stack rỗng hay không
* IsFull(); // kiểm tra stack đầy hay không
* Peek(); // xem phần tử trên đỉnh stack

-----Hết-----

PHỤ LỤC

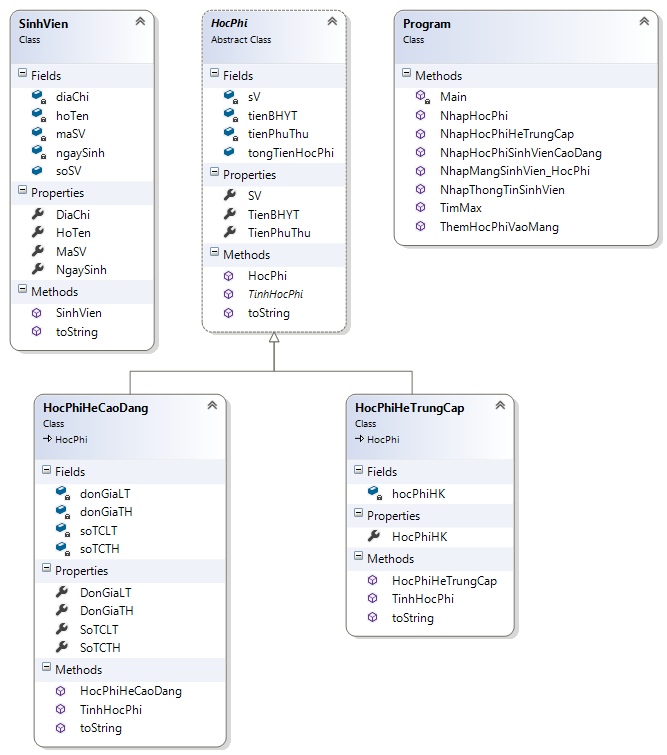
* XâydựngỨng dụng Quản lý học phí của sinh viên hệ niên chế và hệ tín chỉ trong một trường Cao Đẳng.

***Gợi ý****: Hệ thống cho phép ghi lại các thông tin cá nhân của sinh viên. Sinh viên hệ trung cấp học theo hệ niên chế và Sinh viên Cao đẳng học theo hệ tính chỉ.Học phí từng học kỳ được tính như sau:*

*Hoc phí theo hệ niên chế = học phí học kỳ + bảo hiểm y tế + phụ thu*

*Hoc phí theo hệ tính chỉ = số tín chỉ lý thuyết \* đơn giá môn lý thuyết + số tính chỉ thực hành \* đơn giá môn thực hành + bảo hiểm y tế + phụ thu*

**Mô hình lớp tham khảo:**

****

**Code cài đặt tham khảo:**

* **Cài đặt lớp SinhVien:**

|  |
| --- |
| **File: SinhVien.cs** |
| **/\* File: SinhVien.cs**  **\*/**  **using System;**  **namespace QLSinhVien2**  **{**  **public class SinhVien**  **{**  **//fields**  **private string maSV;**  **private string hoTen;**  **private string diaChi;**  **private DateTime ngaySinh;**  **public static int soSV = 0;**  **//Properites**  **public string MaSV**  **{**  **get**  **{**  **return maSV;**  **}**  **private set**  **{**  **maSV = value;**  **}**  **}**  **public string HoTen**  **{**  **get**  **{**  **return hoTen;**  **}**  **private set**  **{**  **hoTen = value;**  **}**  **}**  **public string DiaChi**  **{**  **get**  **{**  **return diaChi;**  **}**  **set**  **{**  **diaChi = value;**  **}**  **}**  **public DateTime NgaySinh**  **{**  **get**  **{**  **return ngaySinh;**  **}**  **set**  **{**  **ngaySinh = value;**  **}**  **}**  **//constructors**  **public SinhVien(string maSV, string hoTen, string diaChi, DateTime ngaySinh)**  **{**  **this.MaSV = maSV;**  **this.HoTen = hoTen;**  **this.DiaChi = diaChi;**  **this.NgaySinh = ngaySinh;**  **soSV++;**  **}**  **//method**  **public string toString()**  **{**  **string s = $"{maSV} - {hoTen} - {ngaySinh}";**  **return s;**  **}**  **}**  **}** |

* **Cài đặt lớp HocPhi:**

|  |
| --- |
| **File: HocPhi.cs** |
| **/\* File: HocPhi.cs**  **\*/**  **using System;**  **namespace QLSinhVien2**  **{**  **public abstract class HocPhi**  **{**  **//Fields**  **private SinhVien sV;**  **private long tienBHYT; //Tiền thu bảo hiểm y tế**  **private long tienPhuThu; // Tiền phụ thu**  **//Properties**  **public SinhVien SV**  **{**  **get**  **{**  **return sV;**  **}**  **set**  **{**  **sV = value;**  **}**  **}**  **public long TienBHYT**  **{**  **get**  **{**  **return tienBHYT;**  **}**  **set**  **{**  **tienBHYT = value;**  **}**  **}**  **public long TienPhuThu**  **{**  **get**  **{**  **return tienPhuThu;**  **}**  **set**  **{**  **tienPhuThu = value;**  **}**  **}**  **//Construcotrs**  **public HocPhi(SinhVien sV, long tienBHYT, long tienPhuThu)**  **{**  **this.sV = sV;**  **this.tienBHYT = tienBHYT;**  **this.tienPhuThu = tienPhuThu;**  **}**  **//Methods**  **abstract public long TinhHocPhi();**  **virtual public string toString()**  **{**  **string s = $"{sV.toString()} - {tienBHYT} - {tienPhuThu}";**  **return s;**  **}**  **}**  **}** |

* **Cài đặt các lớp kế thừa:**

|  |
| --- |
| **File:HocPhiHeCaoDang.cs** |
| **/\* File: HocPhiHeCaoDang.cs**  **\* Date:**  **\* Name**  **\*/**  **using System;**  **namespace QLSinhVien2**  **{**  **public class HocPhiHeCaoDang: HocPhi**  **{**  **//fields**  **int soTCLT; //số tín chỉ lý thuyết**  **int soTCTH; //Số tín chỉ thực hành**  **long donGiaLT;//Đơn giá môn lý thuyết**  **long donGiaTH; //Đơn giá môn thực h**  **public HocPhiHeCaoDang(SinhVien sV, long tienBHYT, long tienPhuThu, int soTCLT, int soTCTH, long donGiaLT, long donGiaTH) : base(sV, tienBHYT, tienPhuThu)**  **{**  **this.soTCLT = soTCLT;**  **this.soTCTH = soTCTH;**  **this.donGiaTH = donGiaTH;**  **this.donGiaLT = donGiaLT;**  **}**  **public int SoTCLT**  **{**  **get**  **{**  **return soTCLT;**  **}**  **set**  **{**  **soTCLT = value;**  **}**  **}**  **public int SoTCTH**  **{**  **get**  **{**  **return soTCTH;**  **}**  **set**  **{**  **soTCTH = value;**  **}**  **}**  **public long DonGiaLT**  **{**  **get**  **{**  **return donGiaLT;**  **}**  **set**  **{**  **donGiaLT = value;**  **}**  **}**  **public long DonGiaTH**  **{**  **get**  **{**  **return donGiaTH;**  **}**  **set**  **{**  **donGiaTH = value;**  **}**  **}**  **public override long TinhHocPhi()**  **{**  **long hp = 0;**  **hp = soTCLT \* DonGiaLT + SoTCTH \* donGiaTH + TienBHYT + TienPhuThu;**  **return hp;**  **}**  **public override string toString()**  **{**  **string s = $"{SV.toString()}\t {TinhHocPhi()}";**  **return s;**  **}**  **}**  **}** |

|  |
| --- |
| **File: HocPhiHeTrungCap.cs** |
| /\* File: HocPhiHeTrungCap.cs  \* Date:  \* Name:  \*/  using System;  namespace QLSinhVien2  {  public class HocPhiHeTrungCap : HocPhi  {  //Fileds  long hocPhiHK;  //Properties  public long HocPhiHK  {  get  {  return hocPhiHK;  }  set  {  hocPhiHK = value;  }  }  //Constructors  public HocPhiHeTrungCap(SinhVien sv, long tienBHYT, long tienPhuThu, long hocPhiHK): base(sv, tienBHYT, tienBHYT)  {  this.hocPhiHK = hocPhiHK;  }  //Methods  public override long TinhHocPhi()  {  long hp = hocPhiHK + TienBHYT + TienPhuThu;  return hp;  }  public override string toString()  {  string s = $"{SV.toString()} - {TinhHocPhi()}";  return s;  }  }  } |

* **Chương trình thử nghiệm:**

|  |
| --- |
| **File: Program.cs** |
| /\* File: Program.cs: Chương trình quản lý Sinh viên - Học phí  \* Date:  \* Name:  \*/  using System;  namespace QLSinhVien2  {  class Program  {  static void Main(string[] args)  {  //static fields  SinhVien.soSV = 0;  //Tạo mảng học phí có thể chưa cả hai loại Cao Đẳng và Trung cấp  HocPhi[] arrHp = new HocPhi[0];  NhapMangSinhVien\_HocPhi(ref arrHp);  //In mảng học phí  foreach (var item in arrHp)  {  Console.WriteLine(item.toString());  }  //Thống kê  Console.Write("Hoc phi lon nhat la:");  Console.WriteLine(TimMax(arrHp));  //sử dụng biến static  Console.WriteLine("So SV da tao: " + SinhVien.soSV);    }  //Nhập Mảng SV và Học phí  public static void NhapMangSinhVien\_HocPhi(ref HocPhi[] arrHp)  {  int soSV = 0;  Console.WriteLine("Nhap thong tin Sinh Vien va hoc phi: ");  Console.Write("Nhap so luong sinh vien: ");  int.TryParse(Console.ReadLine(), out soSV);  for (int i = 0; i < soSV; i++)  {  //\*\*\*Nhập thông tin Sinh viên  SinhVien sv = NhapThongTinSinhVien();  char chon = '\0';  do  {  Console.Write("Chon 1-SV Cao Dang ; 2 - SV Trung Cap. Ban chon: ");  char.TryParse(Console.ReadLine(), out chon);  } while (chon != '1' && chon != '2');    HocPhi hp;  switch(chon)  {  case '1':  //Nhập thông tin học phí cho sv1  hp = NhapHocPhiSinhVienCaoDang(sv);  //Thêm vào mảng học phí  ThemHocPhiVaoMang(ref arrHp, hp);  break;  case '2':  //Nhập thông tin học phí cho sv1  hp = NhapHocPhiHeTrungCap(sv);  //Thêm vào mảng học phí  ThemHocPhiVaoMang(ref arrHp, hp);  break;  }  }  }  //Tim học phí lớn nhất trong mang HocPhiHeCaoDang  public static long TimMax(HocPhi[] arrHp)  {  long max = arrHp[0].TinhHocPhi();  foreach (var item in arrHp)  {  long t = item.TinhHocPhi();  if (max < t)  {  max = t;  }  }  return max;  }  //Thêm 1 phần tử vào mảng học phí  public static HocPhi[] ThemHocPhiVaoMang(ref HocPhi[] arrHp, HocPhi hp )  {  //tăng kích thước mảng  Array.Resize(ref arrHp, arrHp.Length + 1);  //gán vào cuối mảng  arrHp[arrHp.Length -1] = hp;  return arrHp;  }  //Nhập hoc phí hệ trung cấp  public static HocPhiHeTrungCap NhapHocPhiHeTrungCap(SinhVien sv)  {    long hocPhiHK = 0;  long tienBHYT = 0;  long tienPhuThu = 0;  Console.WriteLine();  Console.WriteLine("\*\*\*\*Nhap hoc phi Trung Cap\*\*\*\*\*");  NhapHocPhi(out tienBHYT, out tienPhuThu);  Console.Write("Nhap hoc phi hoc ky: ");  long.TryParse(Console.ReadLine(), out hocPhiHK);  HocPhiHeTrungCap obj = new HocPhiHeTrungCap(sv, tienBHYT, tienPhuThu, hocPhiHK);  return obj;  }  //Nhập học phí hệ cao đẳng  public static HocPhiHeCaoDang NhapHocPhiSinhVienCaoDang(SinhVien sv)  {  int soTCLT = 0;  int soTCTH = 0;  long donGiaLT = 0;  long donGiaTH = 0;  long tienBHYT = 0;  long tienPhuThu = 0;  Console.WriteLine();  Console.WriteLine("\*\*\*\* Nhap hoc phi Cao Dang \*\*\*\*\*");  NhapHocPhi(out tienBHYT, out tienPhuThu);  Console.Write("Nhap so tin chi lý thuyet: ");  int.TryParse(Console.ReadLine(), out soTCLT);  Console.Write("Nhap so tin chi thuc hanh : ");  int.TryParse(Console.ReadLine(), out soTCTH);  Console.Write("Nhap don gia lý thuyet: ");  long.TryParse(Console.ReadLine(), out donGiaLT);  Console.Write("Nhap don gia thuc hanh: ");  long.TryParse(Console.ReadLine(), out donGiaTH);  HocPhiHeCaoDang obj = new HocPhiHeCaoDang(sv, tienBHYT, tienPhuThu, soTCLT, soTCTH, donGiaLT, donGiaTH);  return obj;  }  //Nhap hoc phi  public static void NhapHocPhi(out long tienBHYT, out long tienPhuThu)  {  Console.WriteLine();  Console.Write("Nhap tien BHYT:");  long.TryParse(Console.ReadLine(), out tienBHYT);  Console.Write("Nhap tien phu thu:");  long.TryParse(Console.ReadLine(), out tienPhuThu);  }  //Nhập thông tin Sinh Viên  public static SinhVien NhapThongTinSinhVien()  {  string maSV = "";  string hoTen = "";  string diaChi = "";  DateTime ngaySinh;  Console.WriteLine("\*\*\*\*Nhap Thong tin Sinh vien\*\*\*\*\*");  Console.Write("Nhap ma SV: ");  maSV = Console.ReadLine();  Console.Write("Nhap ho ten SV: ");  hoTen = Console.ReadLine();  Console.Write("Nhap dia chi SV: ");  diaChi = Console.ReadLine();  Console.Write("Nhap ngay sinh SV: ");  DateTime.TryParse(Console.ReadLine(), out ngaySinh);  SinhVien sv = new SinhVien(maSV, hoTen, diaChi, ngaySinh);  return sv;  }  }  } |

**Kết quả:**

|  |
| --- |
| **Nhap thong tin Sinh Vien va hoc phi:**  **Nhap so luong sinh vien: 2**  **\*\*\*\*Nhap Thong tin Sinh vien\*\*\*\*\***  **Nhap ma SV: SV001**  **Nhap ho ten SV: Tran Nham**  **Nhap dia chi SV: Tien Giang**  **Nhap ngay sinh SV: 1/1/2003**  **Chon 1-SV Cao Dang ; 2 - SV Trung Cap. Ban chon: 1**  **\*\*\*\* Nhap hoc phi Cao Dang \*\*\*\*\***  **Nhap tien BHYT:560000**  **Nhap tien phu thu:50000**  **Nhap so tin chi ly thuyet: 10**  **Nhap so tin chi thuc hanh : 15**  **Nhap don gia ly thuyet: 30000**  **Nhap don gia thuc hanh: 35000**  **\*\*\*\*Nhap Thong tin Sinh vien\*\*\*\*\***  **Nhap ma SV: SV002**  **Nhap ho ten SV: Xuan Tai**  **Nhap dia chi SV: Tp.HCM**  **Nhap ngay sinh SV: 25/12/2000**  **Chon 1-SV Cao Dang ; 2 - SV Trung Cap. Ban chon: 2**  **\*\*\*\*Nhap hoc phi Trung Cap\*\*\*\*\***  **Nhap tien BHYT:580000**  **Nhap tien phu thu:20000**  **Nhap hoc phi hoc ky: 2500000**  **SV001 - Tran Nham - 01/01/2003 12:00:00 AM 1435000**  **SV002 - Xuan Tai - 25/12/2000 12:00:00 AM - 3660000**  **Hoc phi lon nhat la:3660000**  **So SV da tao: 2**  **Press any key to continue . . .** |

**TÀI LIỆU THAM KHẢO**

[1] Keogh and Mario Giannini, OOP Demystified: A Self-Teaching Guide, McGraw-Hill/Osborne, 2004

[2] Richard L.Halterman , Fundamentals of C++ program, 2017

[3] Trần Đan Thư, Lập trình hướng đối tượng, Nhà xuất bản khoa học và kỹ thuật, 2010, 370.