

Họ và tên sinh viên:Mã sinh viên: Số máy:...

Chú ý:

1. Xóa toàn bộ ổ D. Đặt tên solution là tên_sinh_viên_3_số_cuối_mã_sinh_viên (ví dụ : **Anh003**), lưu solution vào ổ D
2. Chương trình không biên dịch được (có lỗi cú pháp) không được đánh giá (1 điểm)

Viết chương trình OOP quản lý nhân viên, thông tin cần quản lý của một nhân viên gồm: mã nhân viên, lương một giờ và số giờ làm việc. Chương trình có các yêu cầu sau:

1. Định nghĩa **class nhân viên**: (3 đ)
 - Khai báo các biến thành viên và các phương thức get|set cho các thuộc tính mã nhân viên, lương 1 giờ, số giờ làm việc
 - Tạo constructor không có tham số, nạp chồng constructor có các tham số là mã nhân viên, lương 1 giờ, số giờ làm việc
 - Viết phương thức để tính tiền lương của nhân viên (tiền lương = lương 1 giờ * số giờ làm việc).
 - Viết phương thức nạp chồng ToString để xuất thông tin nhân viên.
2. Sử dụng class đã định nghĩa để quản lý các nhân viên (7 đ)
 - 2.1. **Xây dựng menu** của chương trình gồm các chức năng: Nhập thông tin nhân viên; Hiển thị danh sách nhân viên; Xóa nhân viên; Tìm Nhân viên, Thoát. Chương trình chỉ kết thúc khi người dùng chọn thoát. Nếu người dùng nhập sai thì thông báo cho người dùng biết và cho phép chọn lại. (2 đ)
 - 2.2. Chức năng **Nhập thông tin nhân viên**: cho phép người sử dụng nhập vào thông tin của các nhân viên. (2 đ)
 - 2.3. Chức năng **Hiển thị danh sách nhân viên**: thông tin hiển thị gồm có mã nhân viên, lương 1 giờ, số giờ làm việc và tiền lương. (1 đ)
 - 2.4. Chức năng Tìm Nhân Viên với mã nhân viên được nhập vào (1đ)
 - 2.5. Chức năng **Xóa nhân viên**: cho phép xóa nhân viên theo mã được nhập vào (1 đ)

Họ và tên sinh viên:Mã sinh viên: Số máy:...

Chú ý:

1. Xóa toàn bộ ổ D. Đặt tên solution là tên_sinh_viên_3_số_cuối_mã_sinh_viên (ví dụ : **Anh003**), lưu solution vào ổ D
2. Chương trình không biên dịch được (có lỗi cú pháp) không được đánh giá (1 điểm)

Viết chương trình OOP quản lý sinh viên, thông tin cần quản lý của một sinh viên gồm: mã sinh viên, điểm toán, điểm văn, điểm anh. Chương trình có các yêu cầu sau:

1. Định nghĩa **class sinh viên**: (3 đ)
 - Khai báo các biến thành viên và **các phương thức get/set** cho các thuộc tính mã sinh viên, điểm toán, điểm văn, điểm anh
 - Tạo constructor không có tham số, nạp chồng constructor có các tham số là mã sinh viên, điểm toán, điểm văn, điểm anh
 - Viết phương thức để tính điểm trung bình của sinh viên (điểm trung bình = (điểm toán + điểm văn + điểm anh)/3).
 - Viết phương thức nạp chồng ToString để xuất thông tin sinh viên.
2. Sử dụng class đã định nghĩa để quản lý các sinh viên (7 đ)
 - 2.1. **Xây dựng menu** của chương trình gồm các chức năng: Nhập thông tin sinh viên; Hiển thị danh sách sinh viên; Xóa sinh viên; Tìm sinh viên; Thoát. Chương trình chỉ kết thúc khi người dùng chọn thoát. Nếu người dùng nhập sai thì thông báo cho người dùng biết và cho phép chọn lại. (2 đ)
 - 2.2. Chức năng **Nhập thông tin sinh viên**: cho phép người sử dụng nhập vào thông tin của các sinh viên. (2 đ)
 - 2.3. Chức năng **Hiển thị danh sách sinh viên**: thông tin hiển thị gồm có mã sinh viên, điểm toán, điểm văn, điểm anh, điểm trung bình. (1 đ)
 - 2.4. Chức năng Tìm Sinh Viên với mã sinh viên được nhập vào (1đ)
 - 2.5. Chức năng **Xóa sinh viên**: cho phép xóa sinh viên theo mã được nhập vào (1 đ)

Họ và tên sinh viên:Mã sinh viên: Số máy:...

Chú ý:

1. Xóa toàn bộ ổ D. Đặt tên solution là tên_sinh_viên_3_số_cuối_mã_sinh_viên (ví dụ : **Anh003**), lưu solution vào ổ D
2. Chương trình không biên dịch được (có lỗi cú pháp) không được đánh giá (1 điểm)

Viết chương trình OOP quản lý giảng viên, thông tin cần quản lý của một giảng viên gồm: mã giảng viên, lương một tiết và số tiết giảng. Chương trình có các yêu cầu sau:

1. Định nghĩa **class giảng viên**: (3 đ)
 - Khai báo các biến thành viên và các phương thức get|set cho các thuộc tính mã giảng viên, lương một tiết và số tiết giảng
 - Tạo constructor không có tham số, nạp chồng constructor có các tham số là mã giảng viên, lương một tiết và số tiết giảng
 - Viết phương thức để tính tiền lương của giảng viên (tiền lương= lương một tiết* số tiết giảng).
 - Viết phương thức nạp chồng ToString để xuất thông tin giảng viên.
2. Sử dụng class đã định nghĩa để quản lý các giảng viên (7 đ)
 - 2.1. **Xây dựng menu** của chương trình gồm các chức năng: Nhập thông tin giảng viên; Hiển thị danh sách giảng viên; Xóa giảng viên; Tìm giảng viên; Thoát. Chương trình chỉ kết thúc khi người dùng chọn thoát. Nếu người dùng nhập sai thì thông báo cho người dùng biết và cho phép chọn lại. (2 đ)
 - 2.2. Chức năng **Nhập thông tin giảng viên**: cho phép người sử dụng nhập vào thông tin của các giảng viên. Số giảng viên được nhập thông tin không được xác định trước. (2 đ)
 - 2.3. Chức năng **Hiển thị danh sách giảng viên**: thông tin hiển thị gồm có mã giảng viên, lương một tiết, số tiết giảng, tiền lương. (2 đ)
 - 2.4. Chức năng **Tìm Giảng Viên** với mã giảng viên được nhập vào (1đ)
 - 2.5. Chức năng **Xóa giảng viên**: cho phép xóa giảng viên theo mã được nhập vào (1 đ)

Họ và tên sinh viên:Mã sinh viên: Số máy:...

Chú ý:

1. Xóa toàn bộ ổ D. Đặt tên solution là tên_sinh_vien_3_số_cuối_mã_sinh_vien (ví dụ : **Anh003**), lưu solution vào ổ D
2. Chương trình không biên dịch được (có lỗi cú pháp) không được đánh giá (1 điểm)

Viết chương trình OOP quản lý nhân viên, thông tin cần quản lý của một nhân viên gồm: tên nhân viên, lương một giờ và số giờ làm việc. Chương trình có các yêu cầu sau:

1. Định nghĩa **class nhân viên**: (3 đ)
 - Khai báo các biến thành viên và các phương thức get|set cho các thuộc tính tên nhân viên, lương 1 giờ, số giờ làm việc
 - Tạo constructor không có tham số, nạp chồng constructor có các tham số là tên nhân viên, lương 1 giờ, số giờ làm việc
 - Viết phương thức để tính tiền lương của nhân viên (tiền lương = lương 1 giờ * số giờ làm việc).
 - Viết phương thức nạp chồng ToString để xuất thông tin nhân viên.
2. Sử dụng class đã định nghĩa để quản lý các nhân viên (7 đ)
 - 2.1. **Xây dựng menu** của chương trình gồm các chức năng: Nhập thông tin nhân viên; Hiển thị danh sách nhân viên; Sửa thông tin nhân viên; Xóa thông tin nhân viên; Thoát. Chương trình chỉ kết thúc khi người dùng chọn thoát. Nếu người dùng nhập sai thì thông báo cho người dùng biết và cho phép chọn lại. (2 đ)
 - 2.2. Chức năng **Nhập thông tin nhân viên**: cho phép người sử dụng nhập vào thông tin của các nhân viên. (2 đ)
 - 2.3. Chức năng **Hiển thị danh sách nhân viên**: thông tin hiển thị gồm có tên nhân viên, lương 1 giờ, số giờ làm việc và tiền lương. (1 đ)
 - 2.4. Chức năng **Sửa thông tin nhân viên**: cho phép sửa thông tin nhân viên theo tên được nhập vào (1 đ)
 - 2.5. Chức năng Xóa Nhân Viên với mã nhân viên được nhập vào (1 đ)

Họ và tên sinh viên:Mã sinh viên: Số máy:...

Chú ý:

1. Xóa toàn bộ ổ D. Đặt tên solution là tên_sinh_viên_3_số_cuối_mã_sinh_viên (ví dụ : **Anh003**), lưu solution vào ổ D
2. Chương trình không biên dịch được (có lỗi cú pháp) không được đánh giá (1 điểm)

Viết chương trình OOP quản lý sinh viên, thông tin cần quản lý của một sinh viên gồm: tên sinh viên, điểm toán, điểm văn, điểm anh. Chương trình có các yêu cầu sau:

1. Định nghĩa **class sinh viên**: (3 đ)
 - Khai báo các biến thành viên và các phương thức get|set cho các thuộc tính tên sinh viên, điểm toán, điểm văn, điểm anh
 - Tạo constructor không có tham số, nạp chồng constructor có các tham số là tên sinh viên, điểm toán, điểm văn, điểm anh
 - Viết phương thức để tính điểm trung bình của sinh viên (điểm trung bình = (điểm toán + điểm văn + điểm anh)/3).
 - Viết phương thức nạp chồng ToString để xuất thông tin sinh viên.
2. Sử dụng class đã định nghĩa để quản lý các sinh viên (7 đ)
 - 2.1. **Xây dựng menu** của chương trình gồm các chức năng: Nhập thông tin sinh viên; Hiện thị danh sách sinh viên; Sửa thông tin sinh viên; Xóa thông tin sv, Thoát. Chương trình chỉ kết thúc khi người dùng chọn thoát. Nếu người dùng nhập sai thì thông báo cho người dùng biết và cho phép chọn lại. (2 đ)
 - 2.2. Chức năng **Nhập thông tin sinh viên**: cho phép người sử dụng nhập vào thông tin của các sinh viên. (2 đ)
 - 2.3. Chức năng **Hiện thị danh sách sinh viên**: thông tin hiện thị gồm có tên sinh viên, điểm toán, điểm văn, điểm anh, điểm trung bình. (1 đ)
 - 2.4. Chức năng **Sửa thông tin sinh viên**: cho phép sửa thông tin sinh viên theo tên được nhập vào (1 đ)
 - 2.5. Chức năng xóa Sinh Viên với mã sinh viên được nhập vào (1đ)

Họ và tên sinh viên:Mã sinh viên: Số máy:...

Chú ý:

1. Xóa toàn bộ ổ D. Đặt tên solution là tên_sinh_viên_3_số_cuối_mã_sinh_viên (ví dụ : **Anh003**), lưu solution vào ổ D
2. Chương trình không biên dịch được (có lỗi cú pháp) không được đánh giá (1 điểm)

Viết chương trình OOP quản lý giảng viên, thông tin cần quản lý của một giảng viên gồm: tên giảng viên, lương một tiết và số tiết giảng. Chương trình có các yêu cầu sau:

1. Định nghĩa **class giảng viên**: (3 đ)
 - Khai báo các biến thành viên và các phương thức get|set cho các thuộc tính tên giảng viên, lương một tiết và số tiết giảng
 - Tạo constructor không có tham số, nạp chồng constructor có các tham số là tên giảng viên, lương một tiết và số tiết giảng
 - Viết phương thức để tính tiền lương của giảng viên (tiền lương= lương một tiết* số tiết giảng).
 - Viết phương thức nạp chồng ToString để xuất thông tin giảng viên.
2. Sử dụng class đã định nghĩa để quản lý các giảng viên (7 đ)
 - 2.1. **Xây dựng menu** của chương trình gồm các chức năng: Nhập thông tin giảng viên; Hiển thị danh sách giảng viên; Sửa thông tin giảng viên; Xóa giảng viên; Thoát. Chương trình chỉ kết thúc khi người dùng chọn thoát. Nếu người dùng nhập sai thì thông báo cho người dùng biết và cho phép chọn lại. (2 đ)
 - 2.2. Chức năng **Nhập thông tin giảng viên**: cho phép người sử dụng nhập vào thông tin của các giảng viên. (2 đ)
 - 2.3. Chức năng **Hiển thị danh sách giảng viên**: thông tin hiển thị gồm có tên giảng viên, lương một tiết, số tiết giảng, tiền lương. (2 đ)
 - 2.4. Chức năng **Sửa thông tin giảng viên**: cho phép sửa thông tin giảng viên theo tên được nhập vào (1 đ)
 - 2.5. Chức năng xóa giảng Viên với mã giảng viên được nhập vào (1đ)

Họ và tên sinh viên:Mã sinh viên: Số máy:...

Chú ý:

1. Xóa toàn bộ ổ D. Đặt tên solution là tên_sinh_vien_3_số_cuối_mã_sinh_vien (ví dụ : **Anh003**), lưu solution vào ổ D
2. Chương trình không biên dịch được (có lỗi cú pháp) không được đánh giá (1 điểm)

Viết chương trình OOP quản lý nhân viên, thông tin cần quản lý của một nhân viên gồm: mã nhân viên, lương một giờ và số giờ làm việc. Chương trình có các yêu cầu sau:

1. Định nghĩa **class nhân viên**: (3 đ)
 - Khai báo các biến thành viên và các phương thức get|set cho các thuộc tính mã nhân viên, tên nhân viên, lương 1 giờ, số giờ làm việc
 - Tạo constructor không có tham số, nạp chồng constructor có các tham số là mã nhân viên, tên nhân viên, lương 1 giờ, số giờ làm việc
 - Viết phương thức để tính tiền lương của nhân viên (tiền lương = lương 1 giờ * số giờ làm việc).
 - Viết phương thức nạp chồng ToString để xuất thông tin nhân viên.
2. Sử dụng class đã định nghĩa để quản lý các nhân viên (7 đ)
 - 2.1. **Xây dựng menu** của chương trình gồm các chức năng: Nhập thông tin nhân viên; Hiển thị danh sách nhân viên; Tìm nhân viên; Xóa nhân viên; Thoát. Chương trình chỉ kết thúc khi người dùng chọn thoát. Nếu người dùng nhập sai thì thông báo cho người dùng biết và cho phép chọn lại. (2 đ)
 - 2.2. Chức năng **Nhập thông tin nhân viên**: cho phép người sử dụng nhập vào thông tin của các nhân viên. (2 đ)
 - 2.3. Chức năng **Hiển thị danh sách nhân viên**: thông tin hiển thị gồm có mã nhân viên, tiền lương. (1 đ)
 - 2.4. Chức năng **Tìm nhân viên**: cho tìm nhân viên theo mã được nhập vào và hiển thị thông tin của nhân viên đó gồm Tên, số giờ làm việc, lương 1 giờ, tiền lương(1 đ)
 - 2.5. Chức năng Xóa Nhân Viên với mã nhân viên được nhập vào (1đ)

Họ và tên sinh viên:Mã sinh viên: Số máy:...

Chú ý:

1. Xóa toàn bộ ổ D. Đặt tên solution là tên_sinh_viên_3_số_cuối_mã_sinh_viên (ví dụ : **Anh003**), lưu solution vào ổ D
2. Chương trình không biên dịch được (có lỗi cú pháp) không được đánh giá (1 điểm)

Viết chương trình OOP quản lý sinh viên, thông tin cần quản lý của một sinh viên gồm: mã sinh viên, tên sinh viên, điểm toán, điểm văn, điểm anh. Chương trình có các yêu cầu sau:

1. Định nghĩa **class sinh viên**: (3 đ)
 - Khai báo các biến thành viên và các phương thức get|set cho các thuộc tính mã sinh viên, tên sinh viên, điểm toán, điểm văn, điểm anh
 - Tạo constructor không có tham số, nạp chồng constructor có các tham số là mã sinh viên, tên sinh viên, điểm toán, điểm văn, điểm anh
 - Viết phương thức để tính điểm trung bình của sinh viên (điểm trung bình = (điểm toán + điểm văn + điểm anh) / 3).
 - Viết phương thức nạp chồng ToString để xuất thông tin sinh viên.
2. Sử dụng class đã định nghĩa để quản lý các sinh viên (7 đ)
 - 2.1. **Xây dựng menu** của chương trình gồm các chức năng: Nhập thông tin sinh viên; Hiển thị danh sách sinh viên; Tìm sinh viên; Xóa sinh viên; Thoát. Chương trình chỉ kết thúc khi người dùng chọn thoát. (2 đ)
 - 2.2. Chức năng **Nhập thông tin sinh viên**: cho phép người sử dụng nhập vào thông tin của các sinh viên. (2 đ)
 - 2.3. Chức năng **Hiển thị danh sách sinh viên**: thông tin hiển thị gồm có mã sinh viên, điểm trung bình. (1 đ)
 - 2.4. Chức năng **Tìm sinh viên**: cho phép tìm sinh viên theo mã được nhập vào, hiển thị thông tin cho sinh viên tìm được gồm tên sinh viên, điểm toán, điểm văn, điểm anh, điểm trung bình (1 đ)
 - 2.5. Chức năng xóa sinh Viên với mã nhân viên được nhập vào (1đ)

Họ và tên sinh viên:Mã sinh viên: Số máy:...

Chú ý:

1. Xóa toàn bộ ổ D. Đặt tên solution là tên_sinh_viên_3_số_cuối_mã_sinh_viên (ví dụ : **Anh003**), lưu solution vào ổ D
2. Chương trình không biên dịch được (có lỗi cú pháp) không được đánh giá (1 điểm)

Viết chương trình OOP quản lý giảng viên, thông tin cần quản lý của một giảng viên gồm: mã giảng viên, tên giảng viên, lương một tiết và số tiết giảng. Chương trình có các yêu cầu sau:

1. Định nghĩa **class giảng viên**: (3 đ)
 - Khai báo các biến thành viên và các phương thức get|set cho các thuộc tính mã giảng viên, tên giảng viên, lương một tiết và số tiết giảng
 - Tạo constructor không có tham số, nạp chồng constructor có các tham số là mã giảng viên, tên giảng viên, lương một tiết và số tiết giảng
 - Viết phương thức để tính tiền lương của giảng viên (tiền lương= lương một tiết* số tiết giảng).
 - Viết phương thức nạp chồng ToString để xuất thông tin giảng viên.
2. Sử dụng class đã định nghĩa để quản lý các giảng viên (7 đ)
 - 2.1. **Xây dựng menu** của chương trình gồm các chức năng: Nhập thông tin giảng viên; Hiển thị danh sách giảng viên; Tìm giảng viên; Xóa giảng viên; Thoát. Chương trình chỉ kết thúc khi người dùng chọn thoát. Nếu người dùng nhập sai thì thông báo cho người dùng biết và cho phép chọn lại. (2 đ)
 - 2.2. Chức năng **Nhập thông tin giảng viên**: cho phép người sử dụng nhập vào thông tin của các giảng viên. (2 đ)
 - 2.3. Chức năng **Hiển thị danh sách giảng viên**: thông tin hiển thị gồm có mã giảng viên, tên giảng viên, tiền lương. (1 đ)
 - 2.4. Chức năng **Tìm giảng viên**: cho phép tìm giảng viên theo mã được nhập vào. Hiển thị thông tin của giảng viên tìm được gồm: tên giảng viên, lương một tiết, số tiết giảng, tiền lương (1 đ)
 - 2.5. Chức năng xóa giảng Viên với mã giảng viên được nhập vào (1đ)

Họ và tên sinh viên:Mã sinh viên: Số máy:...

Chú ý:

1. Xóa toàn bộ ổ D. Đặt tên solution là tên_sinh_viên_3_số_cuối_mã_sinh_viên (ví dụ : **Anh003**), lưu solution vào ổ D
2. Chương trình không biên dịch được (có lỗi cú pháp) không được đánh giá (1 điểm)

Viết chương trình OOP quản lý hàng hóa, thông tin cần quản lý của một mặt hàng gồm: mã hàng, số lượng và giá. Chương trình có các yêu cầu sau:

1. Định nghĩa **class hàng hóa**: (3 đ)
 - Khai báo các biến thành viên và các phương thức get|set cho các thuộc tính mã hàng, số lượng và giá
 - Tạo constructor không có tham số, nạp chồng constructor có các tham số là mã hàng, số lượng và giá
 - Viết phương thức để tính tổng tiền (tổng tiền = số lượng * giá).
 - Viết phương thức nạp chồng ToString để xuất thông tin Hàng hóa.
2. Sử dụng class đã định nghĩa để quản lý các mặt hàng (7 đ)
 - 2.1. **Xây dựng menu** của chương trình gồm các chức năng: Nhập thông tin mặt hàng; Hiện thị danh sách hàng; Xóa mặt hàng; tìm mặt hàng; Thoát. Chương trình chỉ kết thúc khi người dùng chọn thoát. Nếu người dùng nhập sai thì thông báo cho người dùng biết và cho phép chọn lại (2 đ)
 - 2.2. Chức năng **Nhập thông tin mặt hàng**: cho phép người sử dụng nhập vào thông tin của mặt hàng mới. (2 đ)
 - 2.3. Chức năng **Hiện thị danh sách hàng**: thông tin hiện thị gồm có mã hàng, số lượng, giá và tổng tiền(1 đ)
 - 2.4. Chức năng **Xóa mặt hàng**: cho phép xóa mặt hàng theo mã được nhập vào (1 đ)
 - 2.5. Chức năng Tìm mặt hàng với mã mặt hàng được nhập vào (1 đ)

Họ và tên sinh viên:Mã sinh viên: Số máy:...

Chú ý:

1. Xóa toàn bộ ổ D. Đặt tên solution là tên_sinh_viên_3_số_cuối_mã_sinh_viên (ví dụ : **Anh003**), lưu solution vào ổ D
2. Chương trình không biên dịch được (có lỗi cú pháp) không được đánh giá (1 điểm)

Viết chương trình OOP quản lý hàng hóa, thông tin cần quản lý của một mặt hàng gồm: tênhàng, số lượng và giá. Chương trình có các yêu cầu sau:

1. Định nghĩa **class hàng hóa**: (3 đ)
 - Khai báo các biến thành viên và các phương thức get|set cho các thuộc tính tênhàng, số lượng và giá
 - Tạo constructor không có tham số, nạp chồng constructor có các tham số là tênhàng, số lượng và giá
 - Viết phương thức để tính tổng tiền (tổng tiền =số lượng*giá).
 - Viết phương thức nạp chồng ToString để xuất thông tin hàng hóa.
2. Sử dụng class đã định nghĩa để quản lý các mặt hàng (7 đ)
 - 2.1. **Xây dựng menu** của chương trình gồm các chức năng: Nhập thông tin mặt hàng; Hiển thị danh sách hàng; Cập nhật thông tin mặt hàng; Xóa mặt hàng; Thoát. Chương trình chỉ kết thúc khi người dùng chọn thoát. Nếu người dùng nhập sai thì thông báo cho người dùng biết và cho phép chọn lại (2 đ)
 - 2.2. Chức năng **Nhập thông tin mặt hàng**: cho phép người sử dụng nhập vào thông tin của mặt hàng mới. (2 đ)
 - 2.3. Chức năng **Hiển thị danh sách hàng**: thông tin hiển thị gồm có tênhàng và tổng tiền(1 đ)
 - 2.4. Chức năng **Cập nhật thông tinmặt hàng**: cho phép sửa thông tin mặt hàng theo mã được nhập vào (1 đ)
 - 2.5. Chức năng Xóa mặt hàng với mã mặt hàng được nhập vào (1đ)

