

# BAN HỌC TẬP CÔNG NGHỆ PHẦN MỀM

TRAINING CUỐI KỲ HỌC KỲ II NĂM HỌC 2022 – 2023



**Sharing is learning**



 **BAN HỌC TẬP**

*Khoa Công nghệ Phần mềm*

*Trường Đại học Công nghệ Thông tin*

*Đại học Quốc gia thành phố Hồ Chí Minh*

 **CONTACT**

*bht.cnpm.uit@gmail.com*

*fb.com/bhtcnpm*

*fb.com/groups/bht.cnpm.uit*

# TRAINING

## OOP – Cuối kỳ

- ⌚ **Thời gian:** 19:30 thứ 2 ngày 19/06/2023
- 📍 **Địa điểm:** Microsoft Team
- 👤 **Trainers:** Lê Hữu Độ – KTMP2022.1  
Lê Duy Nguyễn – KTPM2022.2



Sharing is learning

# Cấu trúc đề thi

- **Dạng câu 1: Lý thuyết**
- **Dạng câu 2: Thiết kế lớp đơn giản**
- **Dạng câu 3: Thiết kế lớp phức tạp**



Sharing is learning

# Cấu trúc đề thi

- **Dạng câu 1: Lý thuyết**
- **Dạng câu 2: Thiết kế lớp đơn giản**
- **Dạng câu 3: Thiết kế lớp phức tạp**



Sharing is learning

# Câu 1: Lý thuyết

1. Trình bày các đặc điểm quan trọng của lập trình hướng đối tượng.
2. Trình bày sự hiểu biết và cho ví dụ minh họa về khái niệm class (lớp) và object (đối tượng) trong lập trình hướng đối tượng.
3. Trình bày những đặc điểm của tính đóng gói (Encapsulation) trong lập trình hướng đối tượng. Trường hợp nào có thể vi phạm tính đóng gói? Cho ví dụ minh họa.
4. Nêu khái niệm constructor và destructor. Phân biệt các kiểu constructor.



Sharing is learning

# Câu 1: Lý thuyết

5. Trình bày những ưu điểm của kế thừa trong lập trình hướng đối tượng và cho ví dụ minh họa?
6. Phân biệt các kiểu kế thừa private, protected, public.
7. Phân biệt các phạm vi truy cập private, protected, public.
8. Phân biệt khái niệm overload (nạp chồng) và override (ghi đè).
9. Hàm thuần ảo là gì? Lớp trừu tượng là gì? Cho ví dụ minh họa.



Sharing is learning

# Lời giải chi tiết



Sổ tay kiến thức OOP



Sharing is learning

# Cấu trúc đề thi

- **Dạng câu 1: Lý thuyết**
- **Dạng câu 2: Thiết kế lớp đơn giản**
- **Dạng câu 3: Thiết kế lớp phức tạp**



Sharing is learning



## Câu 2: Thiết kế lớp đơn giản

- Các kiến thức hay sử dụng:
  - Constructor
  - Destructor
  - Nạp chồng toán tử
  - Cấp phát động
  - ...
- Các lớp đối tượng hay có trong đề: phân số, đơn thức, đa thức, ngày, thời gian, mảng động, ...



Sharing is learning

## Câu 2: Thiết kế lớp đơn giản

**2.1. (Đề 2018-2019)** Xây dựng lớp Thời gian (giờ, phút, giây). Định nghĩa các phép toán:

- ++ để tăng thời gian thêm 1 giây.
- >> và << để nhập, xuất dữ liệu thời gian.
- + để cộng 2 mốc thời gian



Sharing is learning

## Câu 2: Thiết kế lớp đơn giản

**2.1.** Xây dựng lớp Thời gian (giờ, phút, giây). Định nghĩa các phép toán:

- ++ để tăng thời gian thêm 1 giây.
- >> và << để nhập, xuất dữ liệu thời gian.
- + để cộng 2 mốc thời gian

- Có 2 cách biểu diễn thuộc tính của lớp thời gian:
- **Cách 1:** sử dụng 3 biến số nguyên tương ứng với giờ, phút, giây.
- **Cách 2:** chỉ sử dụng 1 biến số nguyên tượng trưng cho giây.



Sharing is learning

## Câu 2: Thiết kế lớp đơn giản

- Các hàm bên ngoài và phương thức cần khai báo:

```
friend istream& operator>> (istream& is, Time& rhs);
```

```
friend ostream& operator<< (ostream& os, const Time& rhs);
```

```
Time& operator++(); // tiền tố
```

```
Time operator++(int); // hậu tố
```

```
Time operator+ (const Time& rhs)
```



Sharing is learning

## Câu 2: Thiết kế lớp đơn giản

**2.2. (Đề 2022-2023, HK1)** Xây dựng class `IntArr` để hàm main hoạt động đúng như mong đợi.

```
class IntArr {  
private:  
    int count; // tổng số lượng phần tử có trong values  
    int* values; // mảng động các số nguyên  
public:  
    ...  
};
```



Sharing is learning

## Câu 2: Thiết kế lớp đơn giản

- **Yêu cầu đề:**  
`IntArr l1;`
- Tạo mảng không chứa bất kì phần tử nào
- **Phương thức cần khai báo:**
- Phương thức thiết lập mặc định:  
`IntArr()`



Sharing is learning

## Câu 2: Thiết kế lớp đơn giản

- Yêu cầu đề:

`IntArr` 12(3, 2);

- Tạo một mảng với 3 phần tử, tất cả phần tử đều có giá trị là 2

- Phương thức cần khai báo:

- Phương thức thiết lập nhận 2 tham số đầu vào:

`IntArr(int size, int init)`



Sharing is learning

## Câu 2: Thiết kế lớp đơn giản

- **Yêu cầu đề:**  
`IntArr l3(2);`
- Tạo một mảng với 2 phần tử, tất cả phần tử đều có giá trị là 0
- **Phương thức cần khai báo:**
- Phương thức thiết lập nhận 1 tham số đầu vào:  
`IntArr(int size)`



Sharing is learning



## Câu 2: Thiết kế lớp đơn giản

- Yêu cầu đề:
- `IntArr l4 = l2.concat(l3);`
- `l2.push(3);`
- Phương thức cần khai báo:
- `IntArr concat(const IntArr&);`
- `void push(int val);`



Sharing is learning

## Câu 2: Thiết kế lớp đơn giản

- Yêu cầu đề:
- `cin >> l2;`
- `cout << l2;`
- Các hàm cần khai báo:
- Hàm nạp chồng toán tử nhập, xuất

```
friend istream& operator>> (istream& is, IntArr&);
```

```
friend ostream& operator<< (ostream& os, const IntArr&);
```



Sharing is learning

## Câu 2: Thiết kế lớp đơn giản

- **Yêu cầu đề:**
- Khi vượt quá phạm vi sử dụng cần huỷ tất cả các vùng nhớ được cấp phát cho các values của IntArr
- **Phương thức cần khai báo:**
- Phương thức phá huỷ:  
`~IntArr();`



Sharing is learning

## Câu 2: Thiết kế lớp đơn giản

- Ngoài ra nên định nghĩa thêm các phương thức thiết lập sao chép và toán tử gán thích hợp:
- `IntArr(const IntArr& rhs);`
- `IntArr& operator= (const IntArr & rhs);`



Sharing is learning

## Câu 2: Thiết kế lớp đơn giản

**2.3. (Đề 2017-2018, HK1)** Hãy định nghĩa lớp Date thích hợp để chương trình dưới đây không bị lỗi biên dịch và chạy đúng. Lưu ý rằng không được chỉnh sửa hàm main và sinh viên cần viết cả các lệnh `#include` thích hợp.



Sharing is learning

## Câu 2: Thiết kế lớp đơn giản

**2.3.** Hãy định nghĩa lớp Date thích hợp để chương trình dưới đây không bị lỗi biên dịch và chạy đúng.

- Biểu diễn thuộc tính của lớp Date bằng **3 biến** số nguyên tượng trưng cho ngày, tháng, năm.



Sharing is learning

## Câu 2: Thiết kế lớp đơn giản

- **Hàm main:**
- `Date ng1;`
- `Date ng2(2017,1);`
- `Date ng3(2017,1,7);`
- **Phương thức cần khai báo:**
- `Date();`
- `Date(int nam, int thang);`
- `Date(int nam, int thang, int ngay);`



Sharing is learning

## Câu 2: Thiết kế lớp đơn giản

- Hàm main:
- `ng1++;`
- `if (ng1 < ng2)`
- ...
- Phương thức cần khai báo:
- `Date operator++(int);`
- `bool operator< (const CDate&);`



Sharing is learning



## Câu 2: Thiết kế lớp đơn giản

- **Hàm main:**
  - `cin >> ng1;`
  - `cout << ng1;`
- **Các hàm bên ngoài cần khai báo:**
  - `friend istream& operator>> (istream& is, Date&);`
  - `friend ostream& operator<< (ostream& os, const Date);`



Sharing is learning

# Lời giải chi tiết



Sổ tay kiến thức OOP



Sharing is learning

## Câu 3: Thiết kế lớp phức tạp

- Các kiến thức được áp dụng
  - Kiến thức cơ bản về lập trình
  - Các tính chất của OOP: Đóng gói, trừu tượng, kế thừa, đa hình

## Câu 3: Thiết kế lớp phức tạp

**(Đề thi lập trình hướng đối tượng cuối kỳ I 2022 – 2023)**

**Câu 3:** Một tổ chức chuyên trưng bày, mua bán các sản phẩm liên quan đến nghệ thuật đang muốn xây dựng một ứng dụng để quản lý các hoá đơn khi bán các sản phẩm. Mỗi lần bán sản phẩm thành công, cửa hàng sẽ lưu trữ các hoá đơn chứa thông tin sản phẩm liên quan. Mỗi hoá đơn sẽ có thông tin: mã hoá đơn, thông tin khách hàng, ngày lập hoá đơn, danh sách sản phẩm, tổng giá (tổng giá trị các sản phẩm trong đơn hàng)...

## Câu 3: Thiết kế lớp phức tạp

... Tổ chức này hiện tại chỉ kinh doanh 2 loại sản phẩm: tranh ảnh và CD âm nhạc (tương lai có thể thay đổi sản phẩm kinh doanh khác). Mỗi sản phẩm sẽ có thông tin chung cần quản trị: mã số, tiêu đề, giá bán. Ngoài thông tin chung, các sản phẩm tranh ảnh cần thêm thông tin kích thước của bức tranh (chiều rộng, chiều cao), tên họa sĩ. Sản phẩm CD âm nhạc sẽ có thêm tên ca sĩ, tên đơn vị sản xuất. Mỗi khách hàng sẽ được lưu trữ các thông tin: mã khách hàng, tên khách hàng, số điện thoại.

## Câu 3: Thiết kế lớp phức tạp

Một tổ chức chuyên trưng bày, mua bán các sản phẩm liên quan đến nghệ thuật đang muốn xây dựng một ứng dụng để quản lý các hoá đơn khi bán các sản phẩm. Mỗi lần bán sản phẩm thành công, cửa hàng sẽ lưu trữ các hoá đơn chứa thông tin sản phẩm liên quan. Mỗi hoá đơn sẽ có thông tin: mã hoá đơn, thông tin khách hàng, ngày lập hoá đơn, danh sách sản phẩm, tổng giá (tổng giá trị các sản phẩm trong đơn hàng)...

## Câu 3: Thiết kế lớp phức tạp

... Tổ chức này hiện tại chỉ kinh doanh 2 loại sản phẩm: tranh ảnh và CD âm nhạc (tương lai có thể thay đổi sản phẩm kinh doanh khác). Mỗi sản phẩm sẽ có thông tin chung cần quản trị: mã số, tiêu đề, giá bán. Ngoài thông tin chung, các sản phẩm tranh ảnh cần thêm thông tin kích thước của bức tranh (chiều rộng, chiều cao), tên họa sĩ. Sản phẩm CD âm nhạc sẽ có thêm tên ca sĩ, tên đơn vị sản xuất. Mỗi khách hàng sẽ được lưu trữ các thông tin: mã khách hàng, tên khách hàng, số điện thoại.

## Câu 3: Thiết kế lớp phức tạp

- Yêu cầu đề bài:
  - Thiết kế và vẽ sơ đồ lớp cho ứng dụng theo bài toán được mô tả
  - Cài đặt chi tiết theo sơ đồ lớp đã thiết kế và cũng như thành phần cần thiết khác để xây dựng chương trình thực hiện các tính năng :
    - Nhập và xuất danh sách các hoá đơn bán hàng
    - Tính tổng thu nhập của cửa hàng
    - Tìm các khách hàng mua nhiều nhất ở cửa hàng (dựa vào tổng giá trị các hoá đơn khách hàng đã mua)



## Câu 3: Thiết kế lớp phức tạp

- Sơ bộ các lớp cần thiết kế:
  - Lớp cửa hàng
  - Lớp hóa đơn
  - Lớp sản phẩm
  - Lớp khách hàng
  - Lớp ngày
  - Lớp tranh ảnh
  - Lớp CD

## Câu 3: Thiết kế lớp phức tạp

- Xác định thuộc tính các lớp
- Xác định phương thức các lớp

## Câu 3: Thiết kế lớp phức tạp

- Xác định thuộc tính của các lớp

**Giả thuyết:** "...cửa hàng sẽ lưu trữ các hoá đơn chứa thông tin sản phẩm liên quan..."

- Lớp cửa hàng
  - Số lượng hóa đơn
  - Danh sách hóa đơn

## Câu 3: Thiết kế lớp phức tạp

- Xác định thuộc tính của các lớp

**Giả thuyết:** "...cửa hàng sẽ lưu trữ các hoá đơn chứa thông tin sản phẩm liên quan..."

- Lớp cửa hàng
  - Số lượng hóa đơn
  - Danh sách hóa đơn

| CuaHang            |
|--------------------|
| # size: int        |
| # hd: HoaDon*[100] |

## Câu 3: Thiết kế lớp phức tạp

- Xác định thuộc tính của các lớp

**Giả thuyết:** "... Mỗi hoá đơn sẽ có thông tin: mã hoá đơn, thông tin khách hàng, ngày lập hoá đơn, danh sách sản phẩm, tổng giá ..."

- Lớp hóa đơn
  - Mã hóa đơn
  - Thông tin khách hàng
  - Ngày tạo
  - Số lượng sản phẩm
  - Danh sách sản phẩm
  - Tổng giá

## Câu 3: Thiết kế lớp phức tạp

- Xác định thuộc tính của các lớp

**Giả thuyết:** "... Mỗi hoá đơn sẽ có thông tin: mã hoá đơn, thông tin khách hàng, ngày lập hoá đơn, danh sách sản phẩm, tổng giá ..."

- Lớp hóa đơn
  - Mã hóa đơn
  - Thông tin khách hàng
  - Ngày tạo
  - Số lượng sản phẩm
  - Danh sách sản phẩm
  - Tổng giá

| HoaDon              |
|---------------------|
| # maDon: string     |
| # khach: Nguoi      |
| # ngayCap: Ngay     |
| # n: int            |
| # sp: SanPham*[100] |
| # tongGia: long     |

## Câu 3: Thiết kế lớp phức tạp

- Xác định thuộc tính của các lớp

**Giả thuyết:** "...Mỗi khách hàng sẽ được lưu trữ các thông tin: mã khách hàng, tên khách hàng, số điện thoại..."

- Lớp khách hàng
  - Mã khách hàng
  - Tên khách hàng
  - Số điện thoại

## Câu 3: Thiết kế lớp phức tạp

- Xác định thuộc tính của các lớp

**Giả thuyết:** "...Mỗi khách hàng sẽ được lưu trữ các thông tin: mã khách hàng, tên khách hàng, số điện thoại..."

- Lớp khách hàng
  - Mã khách hàng
  - Tên khách hàng
  - Số điện thoại

| Người          |
|----------------|
| - maKhach: int |
| - ten: string  |
| - SDT: string  |



## Câu 3: Thiết kế lớp phức tạp

- Xác định thuộc tính của các lớp
  - Lớp ngày
    - Ngày
    - Tháng
    - Năm

| Ngày                                      |
|---|
| - ngay: int<br>- thang: int<br>- nam: int |

## Câu 3: Thiết kế lớp phức tạp

- Xác định thuộc tính của các lớp

**Giả thuyết:** "...Mỗi sản phẩm sẽ có thông tin chung cần quản trị: mã số, tiêu đề, giá bán..."

- Lớp sản phẩm
  - Mã số
  - Tiêu đề
  - Giá bán

## Câu 3: Thiết kế lớp phức tạp

- Xác định thuộc tính của các lớp

**Giả thuyết:** "...Mỗi sản phẩm sẽ có thông tin chung cần quản trị: mã số, tiêu đề, giá bán..."

- Lớp sản phẩm
  - Mã số
  - Tiêu đề
  - Giá bán

| SanPham  |
|--|
| #gia: long<br>#maSo: string<br>#tieuDe: string |

## Câu 3: Thiết kế lớp phức tạp

- Xác định thuộc tính của các lớp

**Giả thuyết:** "...các sản phẩm tranh ảnh cần thêm thông tin **kích thước** của bức tranh (chiều rộng, chiều cao), **tên họa sĩ**..."

- Lớp tranh ảnh
  - Kích thước (chiều dài, chiều rộng)
  - Tên họa sĩ

## Câu 3: Thiết kế lớp phức tạp

- Xác định thuộc tính của các lớp

**Giả thuyết:** "...các sản phẩm tranh ảnh cần thêm thông tin **kích thước** của bức tranh (chiều rộng, chiều cao), **tên họa sĩ**..."

- Lớp tranh ảnh
  - Kích thước (chiều dài, chiều rộng)
  - Tên họa sĩ

| Tranh           |
|-----------------|
| # cao: int      |
| # rong: int     |
| # hoaSi: string |

## Câu 3: Thiết kế lớp phức tạp

- Xác định thuộc tính của các lớp

**Giả thuyết:** "...CD âm nhạc sẽ có thêm tên ca sĩ, tên đơn vị sản xuất..."

- Lớp CD
  - Tên ca sĩ
  - Tên đơn vị sản xuất

## Câu 3: Thiết kế lớp phức tạp

- Xác định thuộc tính của các lớp

**Giả thuyết:** "...CD âm nhạc sẽ có thêm **tên ca sĩ**, **tên đơn vị sản xuất**..."

- Lớp CD
  - Tên ca sĩ
  - Tên đơn vị sản xuất

| CD                              |
|---------------------------------|
| #caSi: string<br>#donVi: string |

## Câu 3: Thiết kế lớp phức tạp

- Xác định các phương thức của lớp (dựa theo yêu cầu đề bài)
  - Yêu cầu:
    - Nhập và xuất danh sách các hoá đơn bán hàng
    - Tính tổng thu nhập của cửa hàng
    - Tìm các khách hàng mua nhiều nhất ở cửa hàng (dựa vào tổng giá trị các hoá đơn khách hàng đã mua)



## Câu 3: Thiết kế lớp phức tạp

- Xác định các phương thức của lớp (dựa theo yêu cầu đề bài)
  - Lớp cửa hàng
    - Nhập
    - Xuất
    - Tổng tiền
    - Xuất tiền khách hàng lớn nhất

## Câu 3: Thiết kế lớp phức tạp

- Xác định các phương thức của lớp (dựa theo yêu cầu đề bài)
  - Lớp cửa hàng
    - Nhập
    - Xuất
    - Tổng tiền
    - Xuất tiền khách hàng lớn nhất

| CuaHang   |
|---|
| # size: int<br># hd: HoaDon*[100]   |
| + Nhap(): void<br>+ Xuat(): void<br>+ ThuNhap(): long<br>+ TimKiemMax(): void |

## Câu 3: Thiết kế lớp phức tạp

- Xác định các phương thức của lớp (dựa theo yêu cầu đề bài)
  - Lớp hóa đơn
    - Nhập
    - Xuất
    - Tính tiền
    - Lấy thông tin

## Câu 3: Thiết kế lớp phức tạp

- Xác định các phương thức của lớp (dựa theo yêu cầu đề bài)
  - Lớp hóa đơn
    - Nhập
    - Xuất
    - Tính tiền
    - Lấy thông tin

| HoaDon   |
|--|
| # maDon: string<br># khach: Nguoi<br># ngayCap: Ngay<br># n: int<br># hd: SanPham*[100]<br># tongGia: long |
| + Nhap(): void<br>+ Xuat(): void<br>+ getTongGia(): long<br>+ getKhach(): Nguoi                            |

## Câu 3: Thiết kế lớp phức tạp

- Xác định các phương thức của lớp (dựa theo yêu cầu đề bài)
  - Lớp sản phẩm
    - Nhập
    - Xuất
    - Tính tiền

## Câu 3: Thiết kế lớp phức tạp

- Xác định các phương thức của lớp (dựa theo yêu cầu đề bài)
  - Lớp sản phẩm
    - Nhập
    - Xuất
    - Tính tiền

| SanPham  |
|--|
| # gia: long<br># maSo: string<br># tieuDe: string    |
| + Nhap(): void<br>+ Xuat(): void<br>+ getGia(): long |

## Câu 3: Thiết kế lớp phức tạp

- Xác định các phương thức của lớp (dựa theo yêu cầu đề bài)
  - Lớp khách hàng
    - `Operator>>(overload)`
    - `Operator<<(overload)`
    - Lấy mã khách

## Câu 3: Thiết kế lớp phức tạp

- Xác định các phương thức của lớp (dựa theo yêu cầu đề bài)
  - Lớp khách hàng
    - `Operator>>(overload)`
    - `Operator<<(overload)`
    - Lấy mã khách

| <b>Nguoi</b>  |
|---|
| - maKhach: int<br>- ten: string<br>- SDT: string        |
| + Nhap(): void<br>+ Xuat(): void<br>+ getMaKhach(): int |



## Câu 3: Thiết kế lớp phức tạp

- Xác định các phương thức của lớp (dựa theo yêu cầu đề bài)
  - Lớp ngày
    - `Operator>>(friend)`
    - `Operator<<(friend)`

## Câu 3: Thiết kế lớp phức tạp

- Xác định các phương thức của lớp (dựa theo yêu cầu đề bài)
  - Lớp ngày
    - `Operator>>(friend)`
    - `Operator<<(friend)`

| Ngày                                      |
|---|
| - ngay: int<br>- thang: int<br>- nam: int |
| + Nhap(): void<br>+ Xuat(): void          |

## Câu 3: Thiết kế lớp phức tạp

- Xác định các phương thức của lớp (dựa theo yêu cầu đề bài)
  - Lớp tranh ảnh (kế thừa từ lớp sản phẩm)
    - Nhập
    - Xuất
    - Tính tiền

## Câu 3: Thiết kế lớp phức tạp

- Xác định các phương thức của lớp (dựa theo yêu cầu đề bài)
  - Lớp tranh ảnh (kế thừa từ lớp sản phẩm)
    - Nhập
    - Xuất
    - Tính tiền

| Tranh  |
|--|
| # cao: int<br># rong: int<br># hoaSi: string |
| + Nhap(): void<br>+ Xuat(): void             |

## Câu 3: Thiết kế lớp phức tạp

- Xác định các phương thức của lớp (dựa theo yêu cầu đề bài)
  - Lớp CD (kế thừa từ lớp sản phẩm)
    - Nhập
    - Xuất
    - Tính tiền

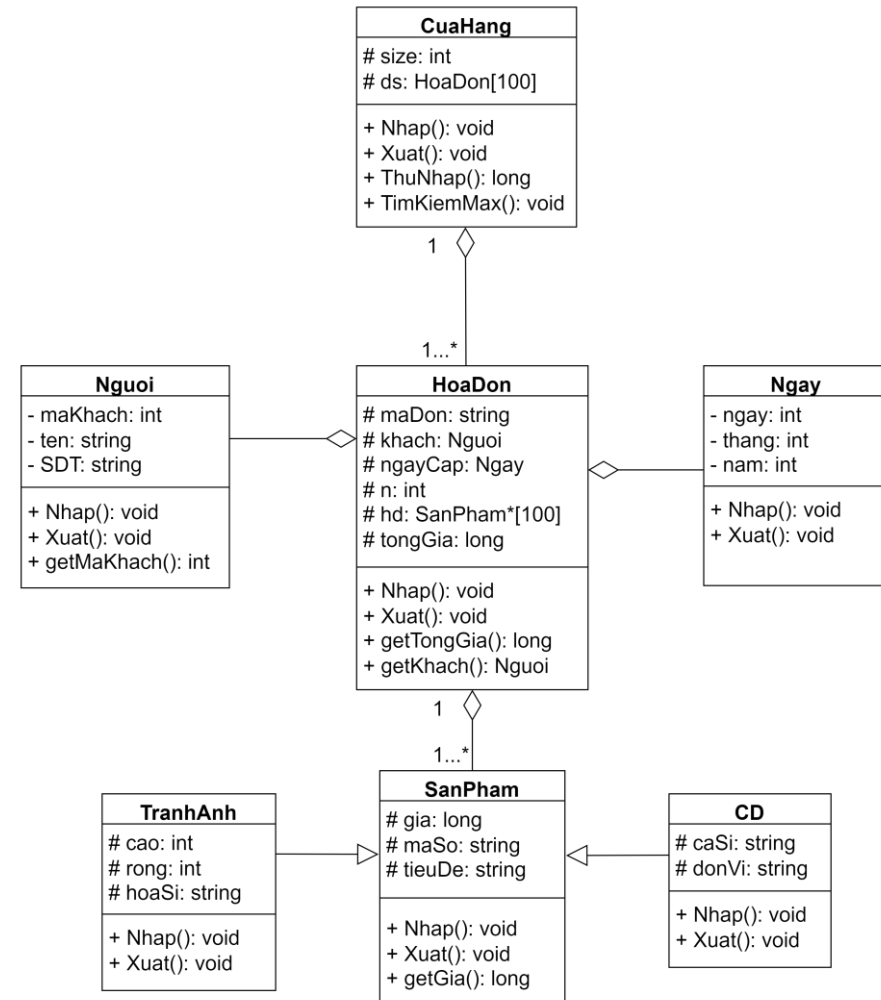
## Câu 3: Thiết kế lớp phức tạp

- Xác định các phương thức của lớp (dựa theo yêu cầu đề bài)
  - Lớp CD (kế thừa từ lớp sản phẩm)
    - Nhập
    - Xuất
    - Tính tiền

| CD                                |
|-----------------------------------|
| # caSi: string<br># donVi: string |
| + Nhap(): void<br>+ Xuat(): void  |

# Câu 3: Thiết kế lớp phức tạp

- Chú thích:
  - # : protected
  - + : public
  - - : private
  - 1-> 1...\* : quan hệ 1 – nhiều



## Câu 3: Thiết kế lớp phức tạp

- Coding:
  - Khai báo thư viện:

```
#include<iostream>
#include<string>
#include<map>
using namespace std;
```



## Câu 3: Thiết kế lớp phức tạp

- Coding:
  - Lớp Nguoi:

| Nguoi   |
|---|
| - maKhach: int<br>- ten: string<br>- SDT: string        |
| + Nhap(): void<br>+ Xuat(): void<br>+ getMaKhach(): int |

```
class Nguoi {  
    private:  
        int maKhach;  
        string ten;  
        string SDT;  
    public:  
        void Nhap();  
        void Xuat();  
        int getMaKhach();  
};
```

## Câu 3: Thiết kế lớp phức tạp

- Coding:
  - Lớp Ngay

| Ngay                                      |
|---|
| - ngay: int<br>- thang: int<br>- nam: int |
| + Nhap(): void<br>+ Xuat(): void          |

```
class Ngay {  
    private:  
        int ngay;  
        int thang;  
        int nam;  
    public:  
        void Nhap();  
        void Xuat();  
};
```

# Câu 3: Thiết kế lớp phức tạp

- Coding:
  - Lớp SanPham

| SanPham  |
|--|
| # gia: long<br># maSo: string<br># tieuDe: string    |
| + Nhap(): void<br>+ Xuat(): void<br>+ getGia(): long |

```
class SanPham {  
    protected:  
        string maSo;  
        string tieuDe;  
        long gia;  
    public:  
        virtual void Nhap();  
        virtual void Xuat();  
        long getGia();  
};
```

## Câu 3: Thiết kế lớp phức tạp

- Coding:
  - Lớp Tranh

| Tranh  |
|--|
| # cao: int<br># rong: int<br># hoaSi: string |
| + Nhap(): void<br>+ Xuat(): void             |

```
class Tranh :public SanPham {  
    protected:  
        int rong;  
        int cao;  
        string hoaSi;  
    public:  
        void Nhap();  
        void Xuat();  
};
```

## Câu 3: Thiết kế lớp phức tạp

- Coding:
  - Lớp CD

| CD                                |
|-----------------------------------|
| # caSi: string<br># donVi: string |
| + Nhap(): void<br>+ Xuat(): void  |

```
class CD :public SanPham {  
    protected:  
        string caSi;  
        string donVi;  
    public:  
        void Nhap();  
        void Xuat();  
};
```

# Câu 3: Thiết kế lớp phức tạp

- Coding:
  - Lớp hóa đơn

| HoaDon   |
|--|
| # maDon: string<br># khach: Nguoi<br># ngayCap: Ngay<br># n: int<br># hd: SanPham*[100]<br># tongGia: long |
| + Nhap(): void<br>+ Xuat(): void<br>+ getTongGia(): long<br>+ getKhach(): Nguoi                            |

```
class HoaDon {  
    protected:  
        string maDon;  
        Nguoi khach;  
        Ngay ngayCap;  
        int n;  
        SanPham* hd[100];  
        long tongGia;  
    public:  
        void Nhap();  
        void Xuat();  
        long getTongGia();  
        Nguoi getKhach();  
};
```

## Câu 3: Thiết kế lớp phức tạp

- Coding:
  - Lớp cửa hàng

| CuaHang   |
|---|
| # size: int<br># ds: HoaDon[100]  |
| + Nhap(): void<br>+ Xuat(): void<br>+ ThuNhap(): long<br>+ TimKiemMax(): void |

```
class CuaHang {  
    protected:  
        int size;  
        HoaDon ds[100];  
    public:  
        void Nhap();  
        void Xuat();  
        long ThuNhap();  
        void TimKiemMax();  
};
```

## Câu 3: Thiết kế lớp phức tạp

- Coding:
  - Định nghĩa phương thức của lớp Ngay

| Ngay                                      |
|---|
| - ngay: int<br>- thang: int<br>- nam: int |
| + Nhap(): void<br>+ Xuat(): void          |

```
void Ngay::Nhap() {  
    cout << "Nhap ngay: ";  
    cin >> ngay;  
    cout << "Nhap thang: ";  
    cin >> thang;  
    cout << "Nhap nam: ";  
    cin >> nam;  
}  
  
void Ngay::Xuat() {  
    cout << ngay << "/" << thang <<  
    "/" << nam << endl;  
}
```



## Câu 3: Thiết kế lớp phức tạp

- Coding:
  - Định nghĩa phương thức của lớp Nguoi

| Nguoi   |
|---|
| - maKhach: int<br>- ten: string<br>- SDT: string        |
| + Nhap(): void<br>+ Xuat(): void<br>+ getMaKhach(): int |

```
void Nguoi::Nhap() {  
    cout << "Nhap ma khách hàng: ";  
    cin >> maKhach;  
    cout << "Nhap họ tên: ";  
    cin >> ten;  
    cout << "Nhap số điện thoại: ";  
    cin >> SDT;  
}  
  
void Nguoi::Xuat() {  
    cout << "Mã khách hàng: " << maKhach <<  
    endl;  
    cout << "Họ tên: " << ten << endl;  
    cout << "Số điện thoại: " << SDT <<  
    endl;  
}  
  
int Nguoi::getMaKhach() {  
    return maKhach;  
}
```

## Câu 3: Thiết kế lớp phức tạp

- Coding:
  - Định nghĩa phương thức của lớp SanPham

| SanPham  |
|--|
| # gia: long<br># maSo: string<br># tieuDe: string    |
| + Nhap(): void<br>+ Xuat(): void<br>+ getGia(): long |

```
void SanPham::Nhap() {
    cout << "Nhap ma so: ";
    cin >> maSo;
    cout << "Nhap tieu de: ";
    cin >> tieuDe;
    cout << "Nhap gia ban: ";
    cin >> gia;
}

void SanPham::Xuat() {
    cout << "Ma so: " << maSo << endl;
    cout << "Tieu de: " << tieuDe << endl;
    cout << "Gia ban: " << gia << endl;
}

long SanPham::getGia() {
    return gia;
}
```

## Câu 3: Thiết kế lớp phức tạp

- Coding:
  - Định nghĩa phương thức của lớp Tranh

| Tranh  |
|--|
| # cao: int<br># rong: int<br># hoaSi: string |
| + Nhap(): void<br>+ Xuat(): void             |

```
void Tranh::Nhap() {
    SanPham::Nhap();
    cout << "Nhap kích thước: " << endl;
    cout << "Nhap chiều rộng: ";
    cin >> rong;
    cout << "Nhap chiều cao: ";
    cin >> cao;
    cout << "Nhap tên họa sĩ: ";
    cin >> hoaSi;
}

void Tranh::Xuat() {
    SanPham::Xuat();
    cout << "Chiều rộng: " << rong << endl;
    cout << "Chiều cao: " << cao << endl;
    cout << "Tên họa sĩ: " << hoaSi << endl;
}
```

## Câu 3: Thiết kế lớp phức tạp

- Coding:
  - Định nghĩa phương thức của lớp CD

| CD                                |
|-----------------------------------|
| # caSi: string<br># donVi: string |
| + Nhap(): void<br>+ Xuat(): void  |

```
void CD::Nhap() {
    SanPham::Nhap();
    cout << "Nhap ten ca si: ";
    cin >> caSi;
    cout << "Nhap don vi san xuat: ";
    cin >> donVi;
}

void CD::Xuat() {
    SanPham::Xuat();
    cout << "Ten ca si: " << caSi << endl;
    cout << "Don vi san xuat: " << donVi << endl;
}
```

## Câu 3: Thiết kế lớp phức tạp

- Coding:
  - Định nghĩa phương thức của lớp HoaDon

```
void HoaDon::Nhap() {  
    cout << "Nhap ma hoa don: ";  
    cin >> maDon;  
    cout << "Nhap thông tin khách hàng:  
" << endl;  
    khách.Nhap();  
    cout << "Nhap ngày cấp: " << endl;  
    ngayCap.Nhap();  
    cout << "Nhap số lượng sản phẩm: ";  
    cin >> n;
```

```
for (int i = 0; i < n; i++) {  
    cout << "Nhap sản phẩm thứ "  
    << i + 1 << endl;  
    int loai;  
    cout << "Nhap loại (0. Tranh  
anh, 1. CD): ";  
    cin >> loai;  
    if (loai == 0) {  
        sp[i] = new Tranh();  
    }  
    else {  
        sp[i] = new CD();  
    }  
    sp[i]->Nhap();  
}
```

## Câu 3: Thiết kế lớp phức tạp

- Coding:
  - Định nghĩa phương thức của lớp HoaDon

| HoaDon   |
|--|
| # maDon: string<br># khach: Nguoi<br># ngayCap: Ngay<br># n: int<br># sp: SanPham*[100]<br># tongGia: long |
| + Nhap(): void<br>+ Xuat(): void<br>+ getTongGia(): long<br>+ getKhach(): Nguoi                            |

```
void HoaDon::Xuat() {  
    cout << "Ma hoa don: " << maDon  
    << endl;  
    cout << "Ngay cap: ";  
    ngayCap.Xuat();  
    cout << "Thong tin khách hàng: "  
    << endl;  
    cout << "Tong gia tien: " <<  
    getTongGia() << endl;  
    khach.Xuat();  
    for (int i = 0; i < n; i++) {  
        cout << "San pham thu " << i  
        + 1 << endl;  
        sp[i]->Xuat();  
    }  
}
```

## Câu 3: Thiết kế lớp phức tạp

- Coding:
  - Định nghĩa phương thức của lớp HoaDon

| HoaDon   |
|--|
| # maDon: string<br># khach: Nguoi<br># ngayCap: Ngay<br># n: int<br># sp: SanPham*[100]<br># tongGia: long |
| + Nhap(): void<br>+ Xuat(): void<br>+ getTongGia(): long<br>+ getKhach(): Nguoi                            |

```
long HoaDon::getTongGia() {  
    tongGia = 0;  
    for (int i = 0; i < n; i++) {  
        tongGia += sp[i]->getGia();  
    }  
    return tongGia;  
}
```

```
Nguoi HoaDon::getKhach() {  
    return khach;  
}
```

## Câu 3: Thiết kế lớp phức tạp

- Coding:
  - Định nghĩa phương thức của lớp CuaHang

| CuaHang   |
|---|
| # size: int<br># hd: HoaDon*[100]   |
| + Nhap(): void<br>+ Xuat(): void<br>+ ThuNhap(): long<br>+ TimKiemMax(): void |

```
void CuaHang::Nhap() {
    cout << "Nhap so luong hoa don: ";
    cin >> size;
    for (int i = 0; i < size; i++) {
        cout << "Nhap hoa don thu " <<
            i + 1 << endl;
        hd[i].Nhap();
    }
}

void CuaHang::Xuat() {
    for (int i = 0; i < size; i++) {
        cout << "Hoa don thu " << i +
            1 << endl;
        hd[i].Xuat();
    }
}
```



## Câu 3: Thiết kế lớp phức tạp

- Coding:
  - Định nghĩa phương thức của lớp CuaHang

| CuaHang   |
|---|
| # size: int<br># hd: HoaDon*[100]   |
| + Nhap(): void<br>+ Xuat(): void<br>+ ThuNhap(): long<br>+ TimKiemMax(): void |

```
long CuaHang::ThuNhap() {  
    int s = 0;  
    for (int i = 0; i < size; i++) {  
        s += hd[i].getTongGia();  
    }  
    return s;  
}
```

## Câu 3: Thiết kế lớp phức tạp

- Coding:
  - Định nghĩa phương thức của lớp CuaHang

```
void CuaHang::TimKiemMax() {  
    map<int, long> mp;  
    for (int i = 0; i < size; ++i) {  
        int id =  
            hd[i].getKhach().getMaKhach()  
            ;  
        mp[id] += hd[i].getTongGia();  
    }  
    long max = 0;
```

```
    for (const auto& pair : mp) {  
        if (pair.second > max) {  
            max = pair.second;  
        }  
    }  
    cout << "ID cac khach mua nhieu  
    nhatt: \n";  
    for (const auto& pair : mp) {  
        if (pair.second == max) {  
            cout << pair.first <<  
                endl;  
        }  
    }  
}
```

## Câu 3: Thiết kế lớp phức tạp

- Coding:
  - Hàm main

```
int main() {  
    CuaHang l;  
    l.Nhap();  
    l.Xuat();  
    cout << "Tong thu nhap: " <<  
    l.ThuNhap() << endl;  
    l.TimKiemMax();  
}
```

# Lời giải chi tiết



Sổ tay kiến thức OOP

# Form điểm danh



# BAN HỌC TẬP CÔNG NGHỆ PHẦN MỀM

TRAINING CUỐI KỲ HỌC KỲ II NĂM HỌC 2022 – 2023



**Sharing is learning**

# HẾT

**CẢM ƠN CÁC BẠN ĐÃ THEO DÕI  
CHÚC CÁC BẠN CÓ KẾT QUẢ THI THẬT TỐT!**

 **BAN HỌC TẬP**

*Khoa Công nghệ Phần mềm*

*Trường Đại học Công nghệ Thông tin*

*Đại học Quốc gia thành phố Hồ Chí Minh*

 **CONTACT**

*bht.cnpm.uit@gmail.com*

*fb.com/bhtcnpm*

*fb.com/groups/bht.cnpm.uit*