**BÁO CÁO THỰC HÀNH**

1. **Thông tin chung**
2. *Họ tên:* Các sinh thuộc nhóm 15 gồm:
   * + 1. *Vi Ngọc Đông*
       2. *Vũ Tuấn Dũng*
       3. *Lưu Thành Đạt*
       4. *Vũ Đức Cường*
       5. *Nguyễn Thành Dương*
3. *Lớp:* KTPM3  *Khóa:* K14
4. *Học phần:…………… Bài số:*1 *Phiếu thực hành số:*5
5. *Ngày thực hành:* 16/01/2022  *Phòng thực hành:…………………….*
6. *Thời gian:*

*Bắt đầu từ ……giờ…….phút đến ….. giờ……..phút*

1. **Nội dung thực hành**

**Bài tập phiếu 5**

**Ý 1:**

* Quan hệ giữa interface ChuyenNghanh và Class SinhVien đang được thể hiện sai. Bởi đó là kí hiệu C:\Users\win\Downloads\1.png thể hiện sự kế thừa giữ lớp cha và lớp con. Như vậy, ta cần sử dụng kí hiệu nàyC:\Users\win\Downloads\2.png để thể hiện mối quan hệ giữa interface ChuyenNghanh và Class SinhVien.

**Ý 2:**



* Quan hệ kế thừa Đối tượng cụ thể (concrete) sẽ kế thừa các thuộc tính và phương thức của đối tượng tổng quát (general).

Giải thích : Liên kết được hình thành để SinhVien có thể kế thừa được thuộc tính của chuyên ngành

* setChuyenNganh() và getChuyenNganh() .

****

* + SinhVien và Diachi có quan hệ *Association* với nhau
  + Object của SinhVien có chứa object của DiaChi
  + Object của SinhVien và Object của DiaChi có vòng đời riêng biệt. Nghĩa là SinhVien bị hủy thì DiaChi sẽ không bị hủy theo.
  + Còn gọi là : Whole SinhVien – Part DiaChi . Nghĩa là SinhVien được tạo từ nhiều DiaChi kết hợp lại , và DiaChi có thể tạo ra độc lập , không cần phải tạo ra SinhVien , DiaChi có thể cùng thuộc 1 whole khác SinhVien.
  + Giải thích : Liên kết được hình thành để sử dụng object của DiaChi
  + Dựa vào thuộc tính DiaChiNha

****

* + Đây là một "mối quan hệ ĐIỂM YẾU". Các đối tượng có thể sống độc lập và thường có các bộ thiết lập hoặc các cách khác để đưa các đối tượng phụ thuộc vào
  + Dựa vào các thuộc tính của SinhVien

**Ý 3:**

* + Lớp sinh viên là lớp cụ thể

**Ý 4:**

* + Khi cài đặt thì thứ tự :
    - 1. Tạo lớp DiaChi
    - 2. Tạo inteface ChuyenNganh
    - 3. Tạo lớp SinhVien
    - 4. Tạo lớp XuatSinhVien chứa main

**Ý 5:**

* + Để nhóm các lệnh thực hiện nhiệm vụ trong hàm được gọn gàng, dễ tìm lỗi và dùng được nhiều lần:

+ Các câu lệnh dễ bị lặp trong quá trình thực hiện dự án nên tạo ra trong 1 hàm riêng để tránh phải viết lại và có thể tái sử dụng nhiều lần.

+ Mỗi một dòng chỉ nên viết 1 dòng code, muốn viết thêm nên xuống dòng và viết tiếp.

+ Thụt đầu dòng sao cho hợp lý cũng là một cách rất tốt trong việc khiến code dễ đọc hơn.

+ Nên tách ra thành các hàm con, mỗi hàm thực hiện một công việc riêng rẽ.

**Ý 6:**

* + Đã hoành thành, vui lòng xem trong tệp đính kèm.

1. **Các vấn đề khác**
2. ***Nội dung đã thực hành***
   * *Các ý: 1, 2, 3, 4, 5, 6.*
3. ***Các nội dung chưa thực hiện và lý do chưa hoàn thành***
4. ***Ngày hoàn tất nội dung và nộp báo cáo bổ sung***
5. ***Các ý kiến đề nghị***

***Hà Nội, ngày 14 tháng 1 năm 2022***

**Nhóm 15**

**Đã ký**