*Sinh viên có thể tự thêm các hàm, tự chọn danh sách tham số, tự đặt chế độ private protected public hợp lý để đảm bảo tính hướng đối tượng (chỉ trừ các thành viên đã nêu rõ private hay protected thì cần giữ nguyên)*

Cho lớp Point minh hoạ một điểm trên mặt phẳng với toạ độ như sau:

class Point

{

public:

double x, y;

Point (double x, double y) : x(x), y(y) {}

};

Cho lớp trừu tượng PhepBienDoi, gồm các thành viên sau:

* Hàm ảo thuần tuý bienDoi nhận một biến tham chiếu Point và biến đổi biến Point đó.
* Hàm bienDoiMang nhận vào một mảng Point và kích thước mảng, sau đó gọi hàm bienDoi với từng phần tử mảng, cuối cùng in ra toạ độ từng phần tử mảng.

Lớp LatNgang (lật theo chiều ngang) kế thừa lớp PhepBienDoi, có :

* Thuộc tính private: một biến axis kiểu double
* Override hàm bienDoi để đổi biến Point truyền vào từ toạ độ (x, y) thành toạ độ (2\*axis-x, y)

Lớp LatDoc (lật theo chiều dọc) kế thừa PhepBienDoi, có :

* Thuộc tính private: một biến axis kiểu double
* Override hàm bienDoi để đổi biến Point truyền vào từ toạ độ (x, y) thành toạ độ (x, 2\*axis - y)

Lớp PhepDich (phép dịch) kế thừa lớp PhepBienDoi, có:

* Thuộc tính private: hai biến shift\_x, shift\_y
* Override hàm bienDoi để đổi biến Point truyền vào từ toạ độ (x, y) thành toạ độ (x + shift\_x, y + shift\_y)

Hàm main tạo một mảng P có 3 phần tử Point, rồi tạo một mảng A có 10 phần tử PhepBienDoi gồm cả LatNgang, LatDoc và PhepDich, sau cùng duyệt mảng A và gọi hàm bienDoiMang trên mảng P.