**Lý thuyết ngôn ngữ hướng đối tượng**

**Mã số sinh viên: Họ tên sinh viên:**

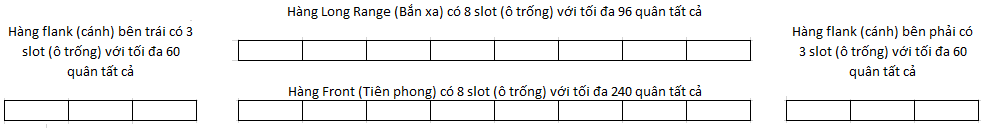
**Email:**

**Bài tập tổng hợp:**

\* \* \* \* \*

Một game online có chia quân bộ thành hai loại: quân có đạn (**Powder**) và quân không có đạn (**NonMunition**). Cả hai đều có chung một số thông số như *damage* (sức công phá), *blood* (máu), *armor* (giáp) là số nguyên. Quân (có đạn) sẽ có thêm thông tin về *munition* (số lượng đạn). Quân không có đạn đương nhiên không có thông tin này.

Quân không có đạn được chia làm: **Hop** (Lính cầm thương), **Sword** (Kiếm thủ) và **Spear** (Ném lao). Khi mỗi trận đánh bắt đầu, tất cả quân sẽ được sắp xếp theo đội hình như sau:



Mỗi **Line** (hàng) sẽ có các thông số về *capacity* (tổng số lính tối đa trong khu vực); *slot* (số lượng các slot trong hàng). Ta có 3 loại hàng khác nhau: **Front** (tiên phong), **LongRange** (Bắn xa) và **Flank** (cánh). Hàng **Flank** có thông tin *isLeft* (kiểu **boolean**) để biết cánh này có ở bên trái hay không?. Mỗi lớp trong số ba lớp trên đều có thuộc tính là mảng hai chiều mang tên *all* có kiểu dữ liệu tùy thuộc vào loại quân có thể chứa trong các *slot*.

Hàng **Front** chỉ chứa các quân không có đạn. Chỉ có quân **Sword** và **Spear** đứng được ở hai hàng **Flank**. Hàng **LongRange** chỉ chứa quân Powder.

Một **Formation** (đội hình) như vậy chứa hai **Flank**, một **Front** và một **LongRange**.

a) Vẽ biểu đồ lớp mô tả hệ thống kể trên. Chú ý rằng không cần thiết phải khai báo lớp đại diện cho Game. Chỉ khai báo các lớp cần thiết nhất.

Trả lời

b) Theo em liệu có cách thức nào đơn giản và hiệu quả nhất để thuộc tính *slot* của lớp **Front** không thể nào bị thay đổi ở bên ngoài lớp **Front,** biết rằng thuộc tính này vẫn được gán giá trị trong phương thức khởi dựng mặc định của lớp **Front**.

Trả lời

c) Khai báo các thuộc tính của lớp Flank và cài đặt phương thức khởi dựng có tham số tên *isLeft* kiểu **boolean** ở lớp **Flank** sao cho trong phương thức đó khởi tạo và/hoặc gán các giá trị phù hợp cho các thuộc tính của nó. Chú ý rằng kích thước các chiều của mảng *all* phải phù hợp số lượng *slot* và số lượng tối đa các quân có thể chứa trong tất cả các *slot*. *Slot* ngoài cùng bên trái được đại diện bởi cột đầu tiên của mảng *all*. (Giả định rằng mảng X[3][5] sẽ đại diện cho ma trận X có 3 hàng và 5 cột). Giả sử rằng tất cả các lớp khác đã được cài đặt.

d) Lớp **Flank** có phương thức *arrange* nhận đầu vào là một mảng một chiều *group* mà chứa các đối tượng của lớp **Sword** và **Spear**. Chú ý rằng không có ràng buộc nào yêu cầu phần tử ở chỉ số j của mảng *group* phải là đối tượng của **Sword** hay **Spear**. Phương thức này sẽ sắp xếp các phần tử của *group* vào các ô trống (từ trái sang phải). Nhớ rằng **Sword** có ưu tiên cao hơn trong việc sắp xếp và không ô *slot* nào chứa cả hai loại quân. Chẳng hạn với *group* có 32 đối tượng **Sword** và 12 đối tượng của **Spear** và khi đó chưa có quân nào trong hàng **Flank** thì slot bên trái chứa 20 **Sword**, slot giữa chứa 12 **Sword** và slot bên phải chứa 12 **Spear**. Giả định rằng: (i) không phần tử nào của *group* là *null* và (ii) khi gọi phương thức *arrange*, có thể các slot đã chứa quân bên trong. Để đơn giản, SV không cần quan tâm đến việc lưu trữ các quân bị thừa ra (khi các slot không thể chứa hết). Hãy cài đặt phương thức này.

**./HẾT/.**