**Part 2: Draw the memory map when the program runs [3 points]**

Explain step by step what happened when the program runs and answer some questions.

* What is stored in the static heap, stack, dynamic heap?
* Why the Organization class is abstract?
* Why must the Colony/University class implement the communicateBytool() method?
* You explain the polymorphism feature in your code.

Describe the difference between an interface and an abstract class.

1 :

Tạo class abstract Organization có chứa trường size ( protected) và phương thức abstract communicateByTool();

Tạo 2 lớp Colony và University kế thừa lớp abstract Organization : 2 lớp này phải ghi đè lại phương thức communicateByTool() vì lớp Organization là lớp abstract;

Tạo interface Role chứa phương thức createWorker() : void

Tạo lớp BeeColony kế thừa lớp Colony và interface Role : lớp BeeColony phải ghi đè lại các phương thức trong interface Role và không cần ghi đè các phương thức trong lớp cha Colony vì lớp Colony không phải lớp astract ;

Tạo lớp FPTUniversity kế thừa lớp University và interface Role : lớp FPTUniversity phải ghi đè lại các phương thức trong interface Role và không cần ghi đè các phương thức trong lớp cha University vì lớp University không phải lớp astract ;

* What is stored in the static heap, stack, dynamic heap?

+ Bộ nhớ stack là một phần của bộ nhớ chứa method, local variable và variable tham chiếu.Bộ nhớ stack luôn được tham chiếu theo last in first out. Local variable thi được tạo trong stack.

+ Bộ nhớ Heap là phần bộ nhớ chưa các Object cũng có thể chưa biến tham chiếu, instance variable được tạo ở đây.

* Why the Organization class is abstract?

+ lớp Origanization là lớp abstract bởi vì nó là 1 lớp trừu tượng , vì không có bất kì thông tin nào để triển khai nó .

* Why must the Colony/University class implement the communicateBytool() method?

+ Bởi vì lớp Colony và lớp University là lớp con của lớp lớp Organization , mà lớp Organization là lớp asbtract nên cần phải implement lại phương thức astract .

* You explain the polymorphism feature in your code.

+ Tính đa hình được thể hiện rõ nhất ở trong @Override . Khi tạo ra lớp FPTUniversity mà phải kế thừa từ lớp cha Organization và implement interface Role thì phải ghi đè lại các phương thức .

Thể hiện được 1 phương thức chúng ta có thể thực hiện được nhiều hành động khác nhau trong từng lớp khác nhau .

* Describe the difference between an interface and an abstract class.

+ Giống:

1. Không cho phép tạo instance

2. Đều hỗ trợ tính đa hình

Khác:

1. Hiện thực thân method:

[interface]: Không cho phép thực hiện phần thân method

[abstract]: Có thể thực hiện phần thân method trong đó

2. Khai báo biến thành viên:

[interface] : Không được

[abstract] : Được

3. Tầm vực

[interface] : Tất cả method là public

[abstract]: Không ràng buộc, các method, biến thành viên có thể là public/ private/ protected, hoặc abstract

4. Từ khóa kế thừa của lớp con

[interface]: Sử dụng từ khóa "implements" để thừa kế

[abstract]: Sử dụng từ khóa "extends" để thừa kế

Để hiểu sâu hơn có thể dịch 2 từ khóa này như sau:

implements nghĩa là ta triển khai 1 phương thức đã được định nghĩa.

extends nghĩa là mở rộng 1 phương thức. (phương thức này đã được định nghĩa và đã có xử lý của nó, ta chỉ mở rộng thêm cho nó mà thôi).

5.Thừa kế

[interface]: Không thể thừa kế từ 1 interface khác

[abstract]: Có thể thừa kế từ 1 lớp abstract khác

[interface]: Không thể thừa kế từ 1 lớp abtract

[abstract]: Có thể thừa kế từ 1 interface

[interface]: Cho phép đa thừa kế

[abstract]: không cho phép đa thừa kế.