OBJECT-ORIENTED LANGUAGE AND THEORY

BÀI TẬP TỔNG HỢP

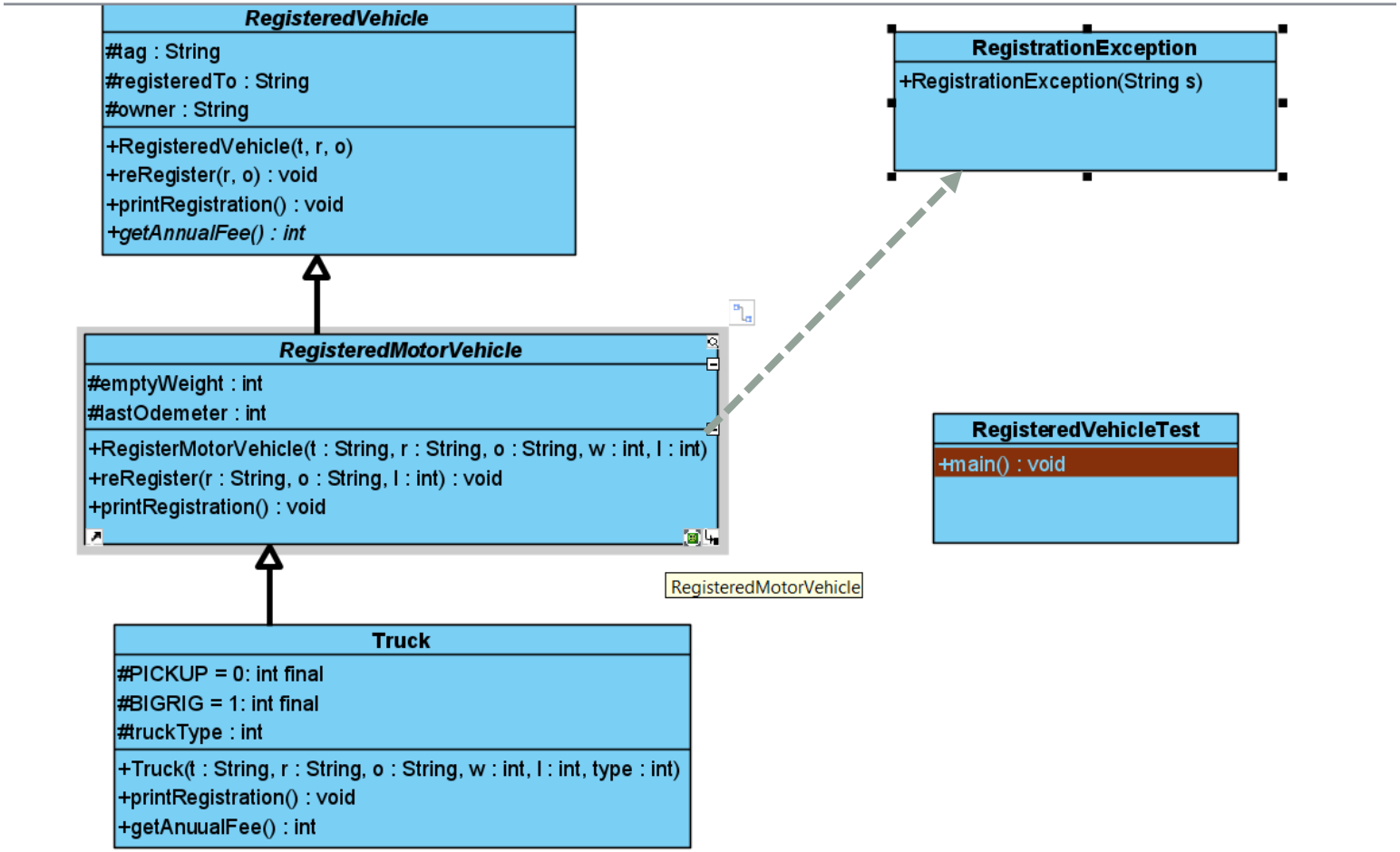
Quản lý đăng ký phương tiện đơn giản

- Điền vào chỗ trống:

A	public RegisteredVehicle	**I**	truckType
B	extends Exception	**J**	case PICKUP
C	abstract	**K**	main (String args[])
D	RegisteredVehicle	**L**	RegisteredMotorVehicle
E	super.reRegister(r,o)	**M**	RegistrationException
F	printRegistration	**N**	vs[i].printRegistration()
G	RegisteredMotorVehicle	**O**	vs[i].getAnnualFee()
H	super(t, r, o, w, l)		

Quản lý đăng ký phương tiện đơn giản

- Sơ đồ lớp:



Câu hỏi lý thuyết

- Câu 1: Hãy nêu điểm khác nhau giữa lớp trừu tượng và lớp trong java?

- Lớp và lớp trừu tượng đều thể hiện các khái niệm trừu tượng hoá, đóng gói các thông tin về thuộc tính và phương thức

Lớp (class)	Lớp trừu tượng (abstract class)
- Tạo các đối tượng từ lớp thông qua khởi tạo	- Không tạo các thể hiện của lớp do lớp chưa hoàn thiện. Khai báo lớp với từ khoá **abstract**
- Tất cả các phương thức đều có phần cài đặt đầy đủ	- Chứa các phương thức trừu tượng, chỉ có chữ kí chứ không có phần thân
- Lớp con khi kế thừa có thể sử dụng các dữ liệu của lớp cha theo nguyên lý kế thừa	- Khi một lớp con kế thừa từ 1 lớp trừu tượng có 2 trường hợp xảy ra: + Lớp con sẽ là lớp thông thường nếu như các phương thức trừu tượng chưa hoàn chỉnh trong lớp cha được ghi đè và hoàn thiện. + Lớp con vẫn sẽ là lớp trừu tượng nếu nó chưa hoàn thiện các phương thức trừu tượng của lớp cha.

Câu hỏi lý thuyết

- Câu 2: Nếu lớp cha chỉ có phương thức khởi dựng mặc định thì lớp con không cần dùng từ khóa super trong phương thức khởi dựng của nó. Nhưng tại sao nếu lớp cha chỉ có một phương thức khởi dựng có tham số (và không có phương thức khởi tạo mặc định) thì trong phương thức khởi dựng của lớp con, ta vẫn phải dùng từ khóa super?

- Trong quá trình khởi tạo đối tượng thì phần dữ liệu của lớp cha sẽ được khởi tạo trước các phần dữ liệu của lớp con → thể hiện qua: câu lệnh đầu tiên trong contructor của lớp con sẽ là lời gọi đến contructor của lớp cha với cú pháp: ***super()*** hoặc ***super(danh sách tham số)***
- Lời gọi này có thể được cung cấp theo 2 cách:
  - Khi lớp cha có contructor mặc định (default constructor – phương thức khởi tạo không tham số) → nếu không có lời gọi super tường minh → Java sẽ TỰ ĐỘNG bổ sung lời gọi super().
  - Khi lớp cha không xây dựng constructor mặc định mà chỉ có 1 contructor chứa tham số → Java sẽ ko tự động bổ sung lời gọi trong lớp con mà lập trình viên sẽ phải tự viết gọi nó một cách tường minh.

Câu hỏi lý thuyết

- Câu 3: Đặc tính sử dụng lại mã được thể hiện ở các nguyên lí nào của lập trình hướng đối tượng. Hãy phân tích sự khác nhau trong việc sử dụng lại các nguyên lí này và cho ví dụ minh họa.

- Đặc tính tái sử dụng mã nguồn được thể hiện ở 2 nguyên lí của lập trình hướng đối tượng là: Kết tập và Kế thừa.

Kết tập	Kế thừa
Kết tập tái sử dụng thông qua đối tượng. Tạo ra lớp mới là tập hợp các đối tượng của các lớp đã có Lớp toàn thể có thể sử dụng dữ liệu và hành vi thông qua các đối tượng thành phần	Kế thừa tái sử dụng thông qua lớp. Tạo lớp mới bằng cách phát triển lớp đã có Lớp con kế thừa dữ liệu và hành vi của lớp cha
Quan hệ "là một phần" ("is a part of")	Quan hệ "là một loại" ("is a kind of")
Không sử dụng từ khoá	Sử dụng từ khoá extends
Biểu diện mối quan hệ giữa 2 lớp trong UML: hình thoi tại phía lớp tổng thể, bội số quan hệ cung cấp thông tin về số lượng các đối tượng của 2 lớp tham gia trong quan hệ	Biểu diện mối quan hệ giữa 2 lớp trong UML: mũi tên tam giác rỗng với đầu mũi tên đặt ở phía lớp cha
Ví dụ: Bánh xe là một phần của Ô tô Viết code minh hoạ	Ví dụ: Ô tô là một loại phương tiện vận tải Viết code minh hoạ

Câu hỏi lý thuyết

- Câu 4: So sánh sự khác nhau giữa phương thức khởi tạo và phương thức thông thường. Tại sao khi xây dựng lớp trong java lại nên xây dựng phương thức khởi tạo mặc định?

Phương thức (method)	Phương thức khởi tạo (constructor)
Phương thức: là thành viên của lớp, mô hình hoá các hoạt động của các đối tượng (đáp ứng cho các thông điệp gửi đến đối tượng)	Constructor: Không được xem là thành viên của lớp, nhiệm vụ khởi tạo giá trị ban đầu cho các thuộc tính khi tạo mới đối tượng
Sử dụng: gọi phương thức thông qua đối tượng và toán tử "." : ***Tên đối tượng.phương thức(danh sách tham số)***	Sử dụng: kết hợp với toán tử new để khởi tạo đối tượng: ***new Constructor(danh sách tham số)***
Cú pháp: có thể đặt tên bất kỳ, khai báo phương thức gồm đầy đủ các thông tin: kiểu trả về, tên phương thức, danh sách tham số và kiểu dữ liệu của tham số	Cú pháp: tên của constructor trùng với tên của lớp, không có kiểu trả về (trả về tham chiếu đến đối tượng được tạo mới) Java: constructor không thể dùng các từ khóa abstract, static, final, native, synchronized.

- Khi xây dựng các lớp trong Java thì chúng ta nên xây dựng phương thức khởi tạo mặc định vì nó là phương thức khởi tạo không tham số, thiết lập các giá trị mặc định cho các thuộc tính của đối tượng → cung cấp cho người lập trình một cách khởi tạo đơn giản, dễ dàng cho các đối tượng

Câu hỏi lý thuyết

- Câu 5: So sánh điểm khác biệt giữa this() và super(). Cho ví dụ cụ thể?

- **this()**: Gọi đến các phương thức khởi tạo khác của lớp.

Ví dụ:

```
public class Ship {
        private double x = 0.0, y = 0.0;
        public String name;
public Ship (String name) {
        this.name = name;
}
public Ship (String name, double x, double y){
        this (name); this.x = x; this.y = y;
}
```

- **super()**: Gọi đến các phương thức khởi tạo của lớp cha (được sử dụng trong các lớp con)

Ví dụ:

```
public class Vehicle {
    protect String name;
    protect int count;
      public Vehicle (String name, int count) {
          this.name = name; this.count = count;
     } }
public class Moto extends Vehicle {
        private String type;
        public Moto (String name, int count, String type){
            super(name, count); this. type = type;
} }
```

Câu hỏi lý thuyết

- Câu 6: Trừu tượng hóa là gì? So sánh đối tượng và lớp?

- Trừu tượng hóa là 1 trong 4 nguyên lí của lập trình hướng đối tượng, là quá trình loại bỏ các thông tin ít quan trọng, chỉ giữ lại những thông tin có nhiều ý nghĩa, quan trọng trong hoàn cảnh cụ thể.

Lớp	Đối tượng
Lớp là mô hình hoá, mô tả các khái niệm.	Đối tượng là các sự vật thật, là thực thể thực sự.
Lớp như 1 bản mẫu, định nghĩa các phương thức và thuộc tính của các đối tượng	Mỗi đối tượng được tạo ra từ một lớp có các dữ liệu cụ thể (thuộc tính) và hành vi (phương thức) của nó.
Một lớp là một sự trừu tượng hóa của các đối tượng	Một đối tượng là một thể hiện (instance) của một lớp

Câu hỏi lý thuyết

- Câu 7: So sánh chồng phương thức và ghi đè lại phương thức.

- Nạp chồng và ghi đè phương thức là các kỹ thuật thể hiện cho nguyên lý đa hình trong lập trình hướng đối tượng

Chồng phương thức (Overloading)	Ghi đè phương thức (Override)
Phạm vi: các phương thức nạp chồng trong cùng lớp	Phạm vi: ghi đè phương thức trong mối quan hệ kế thừa giữa lớp cha và lớp con. Phương thức trong lớp con ghi đè / thay thế / định nghĩa lại phương thức của lớp cha
Cú pháp: các phương thức nạp chồng trùng tên và khác chữ kí (số lượng hay kiểu dữ liệu của đối số)	Cú pháp: phương thức ghi đè hoàn toàn giống về giao diện (chữ kí), giới hạn truy nhập không chặt hơn phương thức của lớp cha, không ném ra các ngoại lệ mới so với phương thức của lớp cha
Các phương thức có thể khác về kiểu dữ liệu trả về	Các phương thức ghi đè phải có cùng kiểu dữ liệu trả về
	Không được ghi đè các phương thức static, private và final trong lớp cha

Câu hỏi lý thuyết

- Câu 8: Phương thức khởi tạo là gì? Chúng có những tác dụng gì? Lấy ví dụ về chồng phương thức khởi tạo?

- Phương thức khởi tạo là phương thức có nhiệm vụ khởi tạo giá trị ban đầu cho các thuộc tính khi tạo mới đối tượng. Tên của constructor trùng với tên của lớp, không có kiểu trả về (trả về tham chiếu đến đối tượng được tạo mới). Sử dụng: kết hợp với toán tử new để khởi tạo đối tượng.

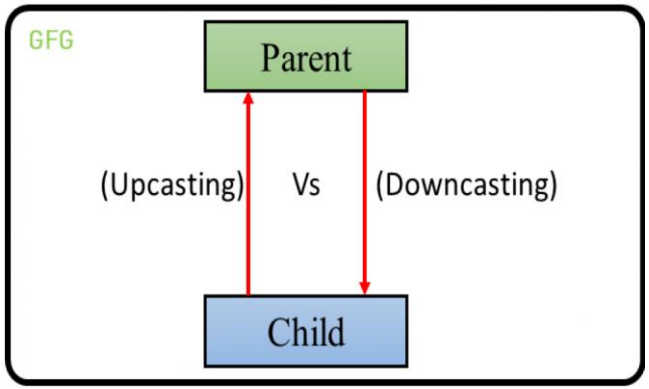
- Ví dụ về chồng phương thức khởi tạo: cung cấp các cách khởi tạo khác nhau cho các đối tượng

```
public class Student {
   private String name;
   private int tuoi;
   public Student() { name = “An”; tuoi = 18; }
   public Student(String name, int tuoi) { this.name = name; this.tuoi = tuoi; }
}
```

Câu hỏi lý thuyết

- Câu 9: Cần chú ý gì khi thực hiện Upcasting và Downcasting các đối tượng của lớp trong lập trình hướng đối tượng?

- Chuyển đổi kiểu của các đối tượng thuộc các lớp có mối quan hệ kế thừa.
- Thể hiện tính đa hình trong lập trình hướng đối tượng.



Upcasting	Downcasting
Đối tượng của lớp cha được gán tham chiếu tới đối tượng của lớp con	Chuyển kiểu đối tượng là một thể hiện của lớp cha xuống thành đối tượng là thể hiện của lớp con
Một biến kiểu dữ liệu supperclass có thể tham chiếu đến tất cả các đối tượng subclass → sử dụng như một kiểu tổng quát	Sử dụng downcasting khi cần sử dụng các tính chất riêng của subclass mà ở supperclass không có
Tự động chuyển đổi kiểu	Phải ép kiểu
Có thể xảy ra lỗi Compile error (lỗi biên dịch)	Có thể xảy ra lỗi Runtime error (lỗi khi chạy chương trình – ngoại lệ ClassCastException) Kiểm tra kiểu trước khi chuyển đổi: toán tử **instanceof**

Câu hỏi lý thuyết

- Câu 10: Hãy nêu sự giống nhau và khác nhau giữa giao diện và lớp trừu tượng trong java.

- Giao diện và lớp trừu tượng: đều chứa các phương thức chưa hoàn thiện (chỉ có phần khai báo, không có phần thân phương thức)
- **Abstract class (Lớp trừu tượng):** có thể hiểu là một class cha cho tất cả các Class có cùng bản chất.
  - Mỗi lớp con chỉ có thể kế thừa từ một lớp trừu tượng (đơn kế thừa).
  - Abstract class không cho phép tạo các thể hiện (instance), vì vậy không thể khởi tạo một đối tượng trực tiếp từ một Abstract class.
- **Interface (Giao diện):** có thể được xem như một mặt nạ cho tất cả các lớp cùng cách thức hoạt động nhưng có thể khác nhau về bản chất.
  - Lớp con có thể thực thi nhiều Interface để bổ sung đầy đủ cách thức hoạt động của mình (đa kế thừa).

Câu hỏi lý thuyết

- Câu 10: Hãy nêu sự giống nhau và khác nhau giữa giao diện và lớp trừu tượng trong java.

Lớp trừu tượng	Giao diện
Abstract Class là "khuôn mẫu" cho các lớp con kế thừa (quan hệ “là một loại” → cùng bản chất)	Interface chứa các "khuôn mẫu" cho phương thức → các lớp con khi thực thi giao diện phải cài đặt cụ thể (các lớp này có thể khác bản chất)
Có thể chứa các phương thức instance (các phương thức đã được triển khai với phần thân cụ thể)	Chỉ có thể chứa danh sách chữ ký phương thức
Phương thức có thể sử dụng những access modifiers như private, protected, default, public, static	Phương thức của interface không cần khai báo access modifiers vì chúng chỉ sử dụng public, asbtract và đã được mặc định
Có thể chứa các thuộc tính final và non-final	Chỉ có thể chứa các thuộc tính hằng
Lớp con phải định nghĩa lại (override) các phương thức trừu tượng	Các lớp thực thi Interface sẽ phải định nghĩa lại toàn bộ các phương thức có trong Interface
Một lớp con chỉ được thừa kế 1 lớp cha là lớp trừu tượng (Dùng từ khóa extends)	Một lớp có thể thực thi nhiều Interface cùng một lúc (Dùng từ khóa implements)

Câu hỏi lý thuyết

- Câu 11: Hãy nêu điều kiện cần và đủ để lập trình viên có thể tiến hành việc: Khai báo đối tượng là đối tượng của lớp A nhưng lại khởi tạo đối tượng của lớp B như sau: A obj = new B();

- Lớp B là con của lớp A
- Lớp B có phương thức khởi tạo mặc định

Một game online đơn giản

- Vẽ biểu đồ lớp mô tả hệ thống: