# BÀI 4 – CÁC KỸ THUẬT KẾ THỪA

*Mục tiêu:*

- *Nắm được khái niệm về ghi đè, lớp trừu tượng và giao diện*

*Thời gian: 90 phút*

## 1. LÝ THUYẾT

***Ghi đè***

Phương thức ở lớp con trùng tên, trùng danh sách tham số với phương thức ở lớp cha.

+ Trình biên dịch sẽ quyết định gọi phương thức nào tùy vào kiểu đối tượng được gọi

***Lớp trừu tượng***

Là một lớp không thể thể hiện hóa trực tiếp

+ Cú pháp: sử dụng từ khóa abstract
+ Một lớp trừu tượng phải có ít nhất một phương thức trừu tượng
+ Các lớp con kế thừa phải có nhiệm vụ ghi đè phương thức trừu tượng này
+ Nếu một lớp đã có một hoặc nhiều phương thức trừu tượng thì bắt buộc phải là lớp trừu tượng

***Giao diện***

Là một cấu trúc quy định những phương thức mà lớp thực thi nó phải thực hiện được.

+ Cú pháp: sử dụng từ khóa interface
+ Chỉ chứa các chữ ký phương thức mà không có phần thực thi.
+ Để thực thi: sử dụng từ khóa implements
+ Lớp thực thi giao diện phải cài đặt tất cả các phương thức khai báo trong giao diện, nếu không phải là lớp trừu tượng

## 2. BÀI TẬP

1. Lớp Tài khoản (Account) sửa thành lớp trừu tượng. Phương thức rút tiền và gửi tiền trở thành phương thức trừu tượng.

2. Xây dựng một giao diện là Tài khoản sinh lãi (HasInterest), trong giao diện này khai báo phương thức tính lãi.

3. Xây dựng lớp Tài khoản vãng lai (CheckingAccount) kế thừa lớp Tài khoản, có các phương thức rút tiền và gửi tiền. Phương thức rút tiền không cho phép rút quá số dư trong tài khoản. Tài khoản vãng lai không có lãi suất.

4. Sửa lại lớp Tài khoản tiết kiệm (SavingAccount) ở bài trước. Lớp này thực thi giao diện Tài khoản sinh lãi. Bổ sung thêm thuộc tính Ngày tạo tài khoản (dateCreated) cho lớp Tài khoản tiết kiệm. Sửa lại phương thức rút tiền và gửi tiền sao cho việc rút tiền và gửi tiền chỉ có thể thực hiện được vào đúng ngày tạo tài khoản hàng năm.

5. Xây dựng lớp Tài khoản tín dụng (CreditAccount) kế thừa lớp Tài khoản. Bổ sung thêm thuộc tính Hạn mức (limit), lãi suất gửi (debitInterest) và lãi suất ghi nợ (creditInterest) và Ngày tạo tài khoản (dateCreated). Phương thức rút tiền cho phép người dùng rút quá số dư trong tài khoản nhưng không vượt quá hạn mức.

Phương thức tính lãi của tài khoản tín dụng phụ thuộc vào số dư trong tài khoản là lớn hơn hay nhỏ hơn 0. Nếu số dư lớn hơn 0 thì khi tính lãi, số dư sẽ được cộng thêm theo lãi suất gửi, còn nếu số dư nhỏ hơn 0 thì số dư sẽ bị tính thêm theo lãi suất ghi nợ.

## 3. GIỚI THIỆU VỀ ROBOCODE

Robocode là một thư viện lập trình cho phép lập trình viên có thể tạo ra các xe tăng để đấu nhau với các xe tăng của các lập trình viên khác.

Tải Robocode tại đây http://robocode.sourceforge.net/

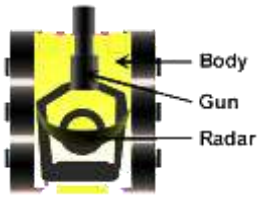
***Thêm thư viện Robocode vào NetBeans***

Ta chọn New Project như bình thường, sau đó vào Properties rồi chọn Add JAR. Tiếp theo đó tìm đến thư mục **libs** của Robocode và chọn *robocode.jar*

Để kết hợp với Javadocs của Robocode, ấn vào Edit và chọn thư mục *Javadoc* của Robocode

Đối với Eclipse làm tương tự

***Cấu trúc của Robot***




Robot có 3 phần: Thân, súng và radar. Ba phần này có thể di chuyển độc lập. Thân có thể quay được và di chuyển tiến hoặc lùi và chuyển động chậm nhất. Súng có thể quay được và chuyển động nhanh hơn thân. Radar cũng có thể quay được và chuyển động nhanh nhất.

Robot sử dụng một nguồn năng lượng chung cho cả tấn công và phòng thủ. Khi bị bắn trúng hay khi robot đâm vào tường, năng lượng của robot sẽ bị giảm. Súng sẽ không bắt được nếu bị overheat (gọi đến phương thức getGunHeat() để xác định giá trị này)

***Viết mã nguồn cho Robot***

Chọn New Class như bình thường. Lớp này kế thừa từ robocode.Robot hoặc có thể import robocode.* và kế thừa lớp Robot.

Sau đó ta có thể ghi đè (override) lại các phương thức của lớp Robot.

```
public class MyFirstRobot extends Robot {
   public void run() {
        while (true) {
            ahead(100);
            turnGunRight(360);
            back(100);
            turnGunRight(360);
        }
    }
    public void onScannedRobot(ScannedRobotEvent e) {
        fire(1);
    }
}
```

***Chạy thử robot trong Robocode***

Chọn menu Options | Preferences, vào tab Development  rồi add thư mục đến gói chứa các package của mình (thư mục ***classes*** trong NetBeans hoặc thư mục *bin* trong Eclipse).

***Một số phương thức quan trọng***

- `setColors` - đổi màu robot
- `setAdjustRadarForGunTurn(true);` - cho phép camera chuyển động độc lập với súng
- `getHeading()` - trả về góc hiện tại của robot
- `getBearing()` - phương thức của event (ví dụ khi scan được một enemy thì đối tượng đó là một event), trả về góc của robot so với đối tượng đó.
- `getDistance()` - trả về khoảng cách từ robot đến đối tượng

*Ví dụ: Đuổi theo một xe tăng khác*

```
public class NewBot extends Robot {
    public void run() {
        setAdjustRadarForGunTurn(true);
        while (true) {
            turnRadarRight(10000); // Quay radar đến lúc phát hiện kẻ địch
        }
     }

    public void onScannedRobot(ScannedRobotEvent e) {
            turnRight(e.getBearing()); // Quay về phía địch
            fire(3); // bắn về phía kẻ địch
            ahead(e.getDistance()); // lao về phía địch
    }
}
```

## 4. BÀI TẬP VỀ NHÀ

Nghiên cứu các phương thức của Robot trong Robocode.

Sử dụng Robocode viết mã nguồn cho một chiếc xe tăng có thể tuần tra và bắn khi gặp kẻ địch.

(Nâng cao) Viết xe tăng có khả năng tránh đạn bắn ra của một số xe tăng mẫu, có khả năng tính trước đường đi của kẻ địch để tấn công.