```java
interface Movement {
    void moveUp();
    void moveDown();
    void moveLeft();
    void moveRight();
}
class RegularPlayer implements Movement {
    @Override
    public void moveUp() {
        System.out.println("Regular Player moved up.");
    }

    @Override
    public void moveDown() {
        System.out.println("Regular Player moved down.");
    }

    @Override
    public void moveLeft() {
        System.out.println("Regular Player moved left.");
    }

    @Override
    public void moveRight() {
        System.out.println("Regular Player moved right.");
    }
}
class JumperPlayer implements Movement {
    @Override
    public void moveUp() {
        System.out.println("Jumper Player jumped up by 5 spaces.");
    }

    @Override
    public void moveDown() {
        System.out.println("Jumper Player jumped down by 5 spaces.");
    }

    @Override
    public void moveLeft() {
        System.out.println("Jumper Player jumped left by 5 spaces.");
    }

    @Override
    public void moveRight() {
        System.out.println("Jumper Player jumped right by 5 spaces.");
    }
}
class OppositePlayer implements Movement {
    @Override
    public void moveUp() {
        System.out.println("Opposite Player moved down (opposite of up).");
    }

    @Override
    public void moveDown() {
        System.out.println("Opposite Player moved up (opposite of down).");
    }

    @Override
    public void moveLeft() {
        System.out.println("Opposite Player moved right (opposite of left).");
    }

    @Override
    public void moveRight() {
        System.out.println("Opposite Player moved left (opposite of right).");
    }
}
public class GameTest {
    public static void main(String[] args) {

        Movement regularPlayer = new RegularPlayer();
        regularPlayer.moveUp();
        regularPlayer.moveLeft();
        regularPlayer.moveRight();
        regularPlayer.moveDown();
        Movement jumperPlayer = new JumperPlayer();
        jumperPlayer.moveUp();
        jumperPlayer.moveLeft();
        jumperPlayer.moveRight();
        jumperPlayer.moveDown();

        Movement oppositePlayer = new OppositePlayer();
        oppositePlayer.moveUp();
        oppositePlayer.moveLeft();
        oppositePlayer.moveRight();
        oppositePlayer.moveDown();
    }
}
```