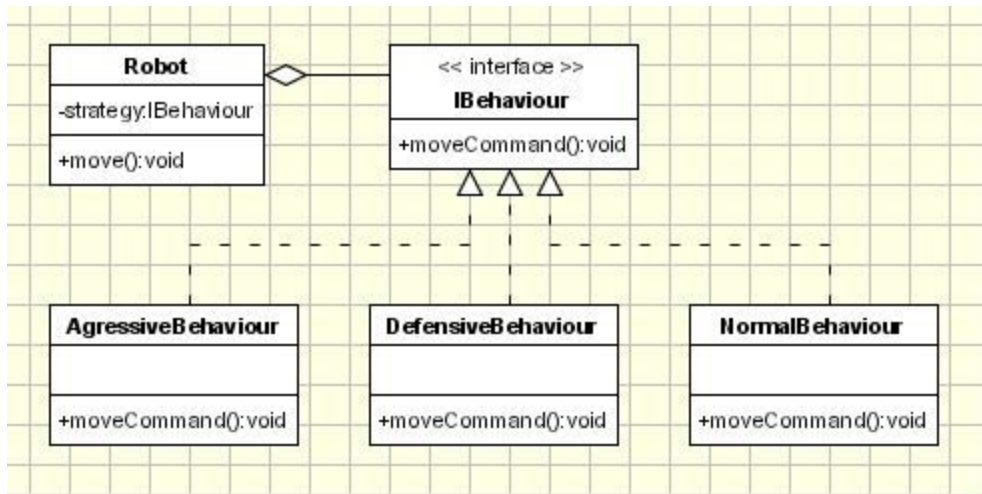


Example - Robots Application



Let's consider an application used to simulate and study robots interaction. For the beginning a simple application is created to simulate an arena where robots are interacting. We have the following classes:

IBehaviour (Strategy) - an interface that defines the behavior of a robot

Concrete Strategies: **AgressiveBehaviour**, **DefensiveBehaviour**, **NormalBehaviour**; each of them defines a specific behavior. In order to decide the action this class needs information that is passed from robot sensors like position, close obstacles, etc.

Robot - The robot is the context class. It keeps or gets context information such as position, close obstacles, etc, and passes necessary information to the Strategy class.

In the main section of the application the several robots are created and several different behaviors are created. Each robot has a different behavior assigned: 'Big Robot' is an aggressive one and attacks any other robot found, 'George v.2.1' is really scared and run away in the opposite direction when it encounter another robot and 'R2' is pretty calm and ignore any other robot. At some point the behaviors are changed for each robot.

```
public interface IBehaviour {
    public int moveCommand();
}

public class AgressiveBehaviour implements IBehaviour{
    public int moveCommand()
    {
        System.out.println("\tAgressive Behaviour: if find another robot attack it");
        return 1;
    }
}

public class DefensiveBehaviour implements IBehaviour{
    public int moveCommand()
```

```

    {
        System.out.println("\tDefensive Behaviour: if find another robot run from it");
        return -1;
    }
}

public class NormalBehaviour implements IBehaviour{
    public int moveCommand()
    {
        System.out.println("\tNormal Behaviour: if find another robot ignore it");
        return 0;
    }
}

public class Robot {
    IBehaviour behaviour;
    String name;

    public Robot(String name)
    {
        this.name = name;
    }

    public void setBehaviour(IBehaviour behaviour)
    {
        this.behaviour = behaviour;
    }

    public IBehaviour getBehaviour()
    {
        return behaviour;
    }

    public void move()
    {
        System.out.println(this.name + ": Based on current position" +
                           "the behaviour object decide the next move:");
        int command = behaviour.moveCommand();
        // ... send the command to mechanisms
        System.out.println("\tThe result returned by behaviour object " +
                           "is sent to the movement mechanisms " +
                           "for the robot " + this.name + "");
    }

    public String getName() {
        return name;
    }
}

```

```

    public void setName(String name) {
        this.name = name;
    }
}

public class Main {

    public static void main(String[] args) {

        Robot r1 = new Robot("Big Robot");
        Robot r2 = new Robot("George v.2.1");
        Robot r3 = new Robot("R2");

        r1.setBehaviour(new AgressiveBehaviour());
        r2.setBehaviour(new DefensiveBehaviour());
        r3.setBehaviour(new NormalBehaviour());

        r1.move();
        r2.move();
        r3.move();

        System.out.println("\n\nNew behaviours: " +
            "\n\t'Big Robot' gets really scared" +
            "\n\t, 'George v.2.1' becomes really mad because" +
            "\n\tit's always attacked by other robots" +
            "\n\tand R2 keeps its calm\n\n");

        r1.setBehaviour(new DefensiveBehaviour());
        r2.setBehaviour(new AgressiveBehaviour());

        r1.move();
        r2.move();
        r3.move();
    }
}

```