```
package robotLearning1;

import java.util.Random;

public class RobotLearning1 {
    static double EPSILON = 0.2;
    static double STEP = 0.05;

    private class Arm {
        private int a, b, count;
        private double q, sumReward;
        private Random rand = new Random();

        Arm(int a, int b){
            this.a = a;
            this.b = b;
            this.sumReward = 0;
            this.q = 0.0;
            this.count = 0;
        }

        double expected() {
            return (double)(a + b)/2.0;
        }

        double reward(){
            // reward is uniformly distributed from [a,b)
            return (rand.nextDouble()*(b-a)+a);
        }

        void reset(double initialQ) {
            this.sumReward = 0;
            this.q = initialQ;
            this.count = 0;
        }
    }

    private Arm[] a;
    private Random r = new Random();
    public RobotLearning1(){
        a = new Arm[4];
        a[0] = new Arm(2,3);
        a[1] = new Arm(-2,1);
        a[2] = new Arm(1,3);
        a[3] = new Arm(0,5);
    }

    public Arm chooseAction(){
        int act = r.nextInt(4);
        return a[act];
    }

    public void reset(double initial) {
        for(int i=0;i<4;i++)
            a[i].reset(initial);
    }

    public double getReward(Arm a){
        return a.reward();
```

```
}

public double getExpected(Arm a){
    return a.expected();
}

public double Q(Arm a){
    double reward = a.reward();
    a.sumReward += reward;
    a.count++;
    a.q = a.q + (1.0/a.count)*(reward - a.q);
    return a.q;
}

public double Q(Arm a, double step){
    double reward = a.reward();
    a.sumReward += reward;
    a.q = a.q + step*(reward - a.q);
    a.count++;
    return a.q;
}

public Arm maxQ(){
    if(a[0].q >= a[1].q && a[0].q >= a[2].q && a[0].q >= a[3].q) {
        return a[0];
    }
    if(a[1].q >= a[0].q && a[1].q >= a[2].q && a[1].q >= a[3].q) {
        return a[1];
    }
    if(a[2].q >= a[0].q && a[2].q >= a[1].q && a[2].q >= a[3].q) {
        return a[2];
    }
    if(a[3].q >= a[0].q && a[3].q >= a[1].q && a[3].q >= a[2].q) {
        return a[3];
    }
    return a[0];
}

public int getCount(Arm a){
    return a.count;
}

public double getSumReward(){
    return a[0].sumReward + a[1].sumReward + a[2].sumReward + a[3].sumReward;
}

public void makeCycle(boolean withLearningStep, double initial) {
    reset(initial);
    Random epsilon = new Random();
    double percent;
    for(int i=1; i<1001; i++) {
        if(i == 1) {
            if (withLearningStep)
                Q(chooseAction(), STEP);
            else
                Q(chooseAction());
            continue;
        }

        if(epsilon.nextDouble() < EPSILON)
```

```
                if (withLearningStep)
                    Q(chooseAction(), STEP);
                else
                    Q(chooseAction());
            else
                Q(maxQ());

            if(i%100 == 0) {
                percent = i;
                System.out.println("Step "+i);
                for(int j=0;j<4;j++)
                    System.out.format("a%d: %.4f%%%n",j+1,
getCount(a[j])*100/percent);
                System.out.format("resulting average reward: %.4f%n",
getSumReward()/i);
            }
        }
    }


    public static void main(String[] args){
        RobotLearning1 rl = new RobotLearning1();

        System.out.println("Task 1:");
        double expVal = 0.0;
        expVal = rl.getExpected(rl.a[0]) + rl.getExpected(rl.a[1]) +
                    rl.getExpected(rl.a[2]) + rl.getExpected(rl.a[3]);
        System.out.format("Expected value: %.4f%n", 0.25*expVal);
        System.out.println("-------------------------------------------------");

        System.out.println("Task 2:");
        double allRew = 0.0;
        for(int i=0; i<1000; i++) {
            allRew += rl.getReward(rl.chooseAction());
        }
        System.out.format("Average reward: %.4f%n", allRew/1000.0);
        System.out.println("-------------------------------------------------");

        System.out.println("Task 3:");
        rl.makeCycle(false, 0);
        System.out.println("-------------------------------------------------");

        System.out.println("Task 4:");
        rl.makeCycle(true, 0);
        System.out.println("-------------------------------------------------");

        System.out.println("Task 5:");
        rl.makeCycle(true, 5);
    }
}
```