

Main.java

```
public class Main {  
    public static void main(String[] args) {  
        Producer dani = new Producer("Dani");  
  
        dani.startAct(70);  
        dani.addAudience();  
        dani.startAct(3);  
    }  
}
```

Producer.java

```
public class Producer extends Actor {

    private Juggler josh;
    private SpinningAcrobat charlene;
    private SpinningAcrobat camille;
    private DancingAcrobat mustafa;
    private AudienceDirector liban;

    public Producer(String name) {
        super(name);

        System.out.println("the producer has been created.");

        josh = new Juggler("Josh", 3);
        System.out.println("you are a juggler, here are 3 balls");

        charlene = new SpinningAcrobat("Charlene");
        System.out.println("you are spinning acrobat");

        camille = new SpinningAcrobat("Camille");
        System.out.println("you are spinning acrobat");

        mustafa = new DancingAcrobat("Mustafa");
        System.out.println("you are a dancing acrobat");

        liban = new AudienceDirector("Liban");
        System.out.println("you are an audience director");

        System.out.println("you are an audience director with 2 audience members.");
        liban.appoint(2);

        System.out.println("return");
    }

    public void startAct(int money) {
        if (money > 5) {
            int choice = (int)(Math.random() * 2);
            if (choice == 0) {
                System.out.println("juggle!");
                josh.juggle();
            } else {
                int ballsToJuggle = (int)(Math.random() * 4) + 1;
                System.out.println("juggle " + ballsToJuggle + " balls!");
                josh.juggle(ballsToJuggle);
            }

            System.out.println("act!");
            charlene.act();

            System.out.println("act!");
            camille.act();

            System.out.println("act!");
            mustafa.act();

            System.out.println("make clap");
            liban.makeClap();

            System.out.println("return");
        } else {
            System.out.println("not enough money to perform");
            System.out.println("return");
        }
    }

    public boolean addAudience() {
        System.out.println("add member");
        boolean ok = liban.addMember();
        System.out.println("return");
        return ok;
    }
}
```


Juggler.java

```
public class Juggler extends Actor {
    private int balls;

    public Juggler(String name, int balls) {
        super(name);
        this.balls = balls;
    }

    public int getBalls() {
        return balls;
    }

    public void juggle() {
        juggle(3);
    }

    public void juggle(int amount) {
        if (amount > balls) {
            System.out.println("I cant do that, I don't have enough juggling balls.");
            System.out.println("return");
            return;
        }
        System.out.println("Juggling " + amount + " balls!");
        System.out.println("return");
    }
}
```

SpinningAcrobat.java

```
public class SpinningAcrobat extends Acrobat {  
    public SpinningAcrobat(String name) {  
        super(name);  
    }  
  
    @Override  
    public void act() {  
        System.out.println("Dice->roll");  
        int spins = Dice.roll();  
        System.out.println(spins);  
  
        System.out.print("Spinning " + spins + " times! ");  
        for (int i = 1; i <= spins; i++) {  
            System.out.print(i + (i < spins ? "... " : "!"));  
        }  
        System.out.println();  
  
        System.out.println("return");  
    }  
}
```

DancingAcrobat.java

```
public class DancingAcrobat extends Acrobat {  
    public DancingAcrobat(String name) {  
        super(name);  
    }  
  
    @Override  
    public void act() {  
        System.out.println("Dice->roll");  
        int seconds = Dice.roll();  
        System.out.println(seconds);  
  
        System.out.print("Dancing for " + seconds + " seconds! ");  
        for (int i = 1; i <= seconds; i++) {  
            System.out.print(i + (i < seconds ? "... " : "!"));  
        }  
        System.out.println();  
  
        System.out.println("return");  
    }  
}
```

AudienceDirector.java

```
import java.util.ArrayList;

public class AudienceDirector extends Actor {
    private ArrayList<AudienceMember> audience;

    public AudienceDirector(String name) {
        super(name);
        audience = new ArrayList<>();
    }

    public void appoint(int numMembers) {
        if (numMembers <= 0) {
            System.out.println("How many audience members am I supposed to manage?!");
            System.out.println("return");
            return;
        }

        for (int i = 1; i <= numMembers; i++) {
            AudienceMember member = new AudienceMember("audience" + i);
            audience.add(member);
            System.out.println("Appointing " + member.getName() + ".");
        }
        System.out.println("return");
    }

    public int getAudienceCount() {
        return audience.size();
    }

    public void makeClap() {
        for (AudienceMember member : audience) {
            System.out.println(member.getName() + ", clap!");
            member.clap();
        }
        System.out.println("return");
    }

    public boolean addMember() {
        int next = audience.size() + 1;
        AudienceMember member = new AudienceMember("audience" + next);
        audience.add(member);
        System.out.println("Appointing " + member.getName() + ".");
        System.out.println("true");
        System.out.println("return");
        return true;
    }
}
```

AudienceMember.java

```
public class AudienceMember extends Actor {  
    public AudienceMember(String name) {  
        super(name);  
    }  
  
    public void clap() {  
        System.out.println("*clap clap clap*");  
        System.out.println("return");  
    }  
}
```


Actor.java

```
public abstract class Actor {  
    protected String name;  
  
    public Actor(String name) {  
        this.name = name;  
    }  
  
    public String getName() {  
        return name;  
    }  
}
```

Acrobat.java

```
public abstract class Acrobat extends Actor {  
    public Acrobat(String name) {  
        super(name);  
    }  
  
    public abstract void act();  
}
```

Dice.java

```
public abstract class Dice {  
    public static int roll() {  
        return (int)(Math.random() * 6) + 1;  
    }  
}
```