Class A

```
package ec.edu.espe.conjunta.model;

import java.util.ArrayList;
import java.util.List;

/**
 *
 @author Luis Haro
 */
public class A {
    private List<A> as = new ArrayList<>();

    public A() {
        System.out.println("A::A()");
    }

    public void addA(A a) {
        as.add(a);
    }

    public List<A> getAs() {
        return as;
    }

    public void setAs(List<A> as) {
        this.as = as;
    }

    @Override
    public String toString() {
```

```
        return "A{" + "as=" + as + '}';
    }
}
```

Class B

```
package ec.edu.espe.conjunta.model;

import java.util.ArrayList;
import java.util.List;

/**
 *
@author Luis Haro
 */
public class B extends A {
    private final List<H> hs = new ArrayList<>();

    public B() {
        System.out.println("B::B()");
        hs.add(new G(new J()));
    }

    @Override
    public String toString() {
        return "B{" + '}';
    }
}
```

Class C

```java
package ec.edu.espe.conjunta.model;

import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author Luis Haro
 */
public class C extends A {
    private List<E> es = new ArrayList<>();

    public C() {
        System.out.println("C::C()");
        es.add(new E());
    }

    public List<E> getEs() {
        return es;
    }

    public void setEs(List<E> es) {
        this.es = es;
    }

    @Override
    public String toString() {
        return "C{" + '}';
    }
}
```

Class D

```
package ec.edu.espe.conjunta.model;

import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author Luis Haro
 */
public class D extends A {
    private List<F> fs = new ArrayList<>();
    private E es[] = new E[5];

    public List<F> getFs() {
        return fs;
    }

    public void setFs(List<F> fs) {
        this.fs = fs;
    }

    public E[] getEs() {
        return es;
    }

    public void setEs(E[] es) {
        this.es = es;
    }

    public D() {
```

```
        System.out.println("D::D()");
        fs.add(new F());

        es[0] = new E();
        es[1] = new E();
        es[2] = new E();
        es[3] = new E();
        es[4] = new E();
    }
}
```

Class E

```
package ec.edu.espe.conjunta.model;

/**
 *
 * @author Luis Haro
 */
public class E {
    public E() {
        System.out.println("E::E()");
    }

    @Override
    public String toString() {
        return "E{" + '}';
    }
}
```

Class F

```
package ec.edu.espe.conjunta.model;
```

```
/**
 *
 *@author Luis Haro
 */
public class F {
    public F() {
        System.out.println("F::F()");
    }

    @Override
    public String toString() {
        return "F{" + '}';
    }
}
```

Class G

```
package ec.edu.espe.conjunta.model;

/**
 *
 * @author Luis Haro
 */
public class G implements H {
    public G(J j) {
        System.out.println("G::G(" + j + ")");
    }

    public G() {
    }
```

```
    @Override
    public String toString() {
        return "G{" + '}';
    }
}
```

Class H

```
package ec.edu.espe.conjunta.model;

/**
 *
 * @author Luis Haro
 */
public interface H {

}
```

Class J

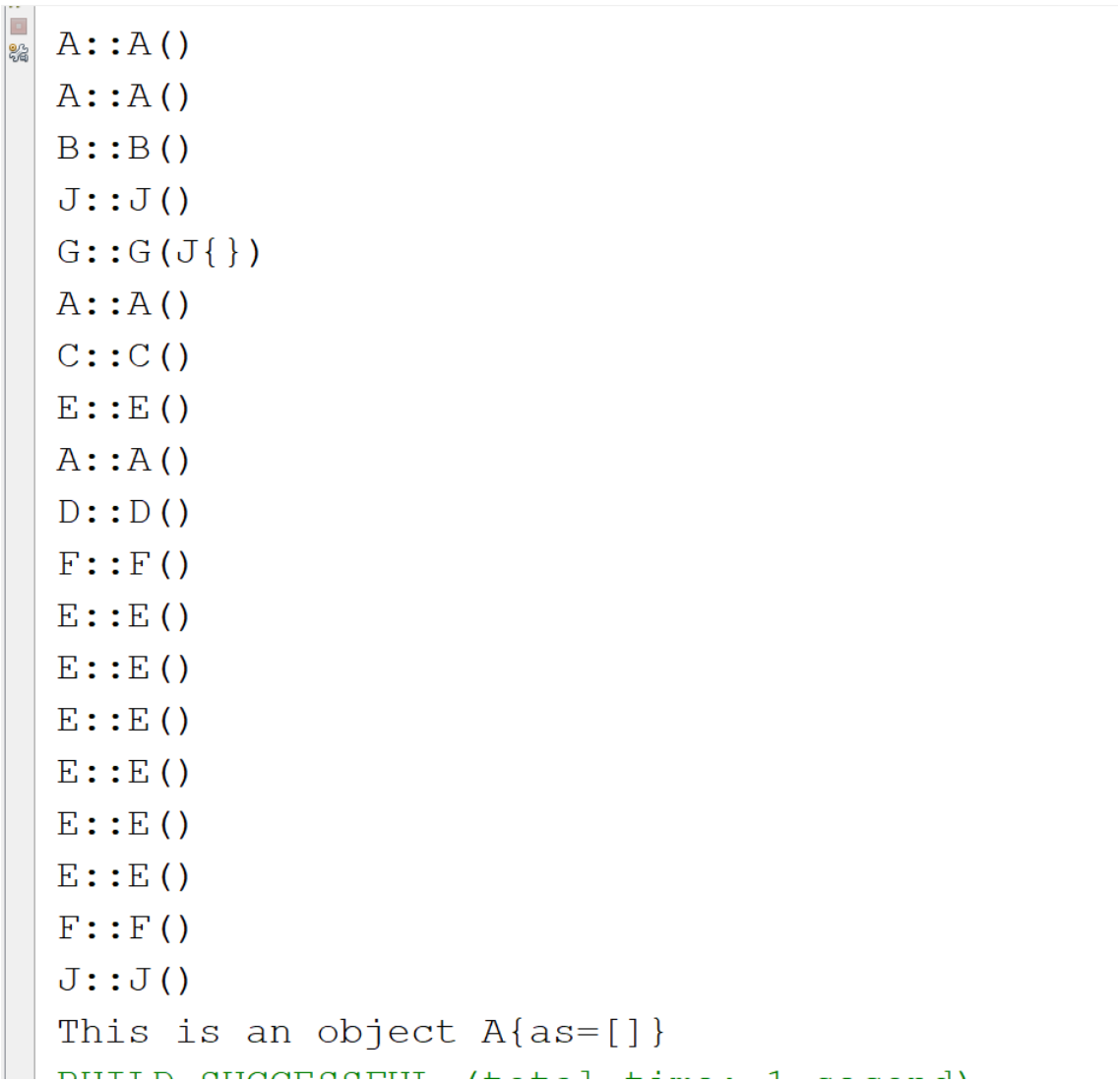
```
package ec.edu.espe.conjunta.model;

/**
 *
 * @author Luis Haro
 */
public class J {
    public J() {
        System.out.println("J::J()");
    }

    @Override
```

```
    public String toString() {
        return "J{" + '}';
    }
}
```

Main

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package ec.edu.espe.conjunta.view;

import ec.edu.espe.conjunta.model.*;
import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author Luis Haro
 */
public class ConjuntaSystem {
    public static void main(String[] args) {
        A a = new A();
        B b = new B();
        C c = new C();
        D d = new D();
        E e = new E();
        F f = new F();
        G g = new G();
        J j = new J();
```

```
        System.out.println("This is an object " + a.toString());

    }

}
```



```
A::A()
A::A()
B::B()
J::J()
G::G(J{})
A::A()
C::C()
E::E()
A::A()
D::D()
F::F()
E::E()
E::E()
E::E()
E::E()
E::E()
E::E()
E::E()
F::F()
J::J()
This is an object A{as=[]}
```