

```

interface I1 {
    public void method1();
}

interface I2 {
    public void method1();
}

class Alpha implements I1 {
    String x = "Alpha";

    public void method1() {
        System.out.println("Alpha m1 x = " + x);
    }
}

class Beta extends Alpha implements I2 {
    String x = "Beta";

    public void method1() {
        System.out.println("Beta m1 x = " + x);
    }
}

class Phi implements I1 {
    String x = "Phi";

    public void method1() {
        super.method1();
        System.out.println("Phi m1 x = " + x);
    }
}

```

```

class Gamma extends Alpha {
    String x = "Gamma";

    void method0() {
        System.out.println("Gamma2 m0 x = " + x);
    }
    public void method1() {
        System.out.println("Gamma m1 x = " + x);
        method0();
    }
}

class Iota extends Gamma {
    public void method0() {
        System.out.println("Iota m0 x = " + x);
    }
}

class Eta extends Gamma {
    void method0(String s) {
        System.out.println("Eta m0 s =" + s);
    }
}

```

In einer Klasse Explore im selben Package befindet sich die Methode main.

```
public static void main (String[] args) {  
  
    /* Body */  
  
}
```

Die folgenden Anweisungen sollen als “Body” (Rumpf) anstelle des Kommentars `/* body */` eingefügt werden. Geben Sie für jede Anweisungsfolge an, was für eine Ausgabe erzeugt wird – entweder was gedruckt wird, oder ob ein Laufzeitfehler auftritt (schreiben Sie “Exception”), oder ob der Compiler einen Fehler feststellt (schreiben Sie “Compile-Fehler”). Falls ein gedruckter String Leerzeichen/Umbrüche enthält, dann ist die genaue Anzahl/Weite unwichtig.

1.	<code>I1 a = new Alpha(); a.method1();</code>	Alpha m1 x = Alpha -----
2.	<code>Beta b = new Beta(); b.method1();</code>	Beta m1 x = Beta -----
3.	<code>I1 ba = new Beta(); ba.method1();</code>	Beta m1 x = Beta -----
4.	<code>I2 ab = new Beta(); ((I1) ab).method1();</code>	Beta m1 x = Beta -----
5.	<code>Alpha a = new Beta(); a.method1();</code>	Beta m1 x = Beta -----
6.	<code>Alpha a = new Gamma(); a.method1();</code>	Gamma m1 x = Gamma Gamma2 m0 x = Gamma -----
7.	<code>Phi p = new Phi(); p.method1();</code>	Compile(r)-Fehler / Compile(r)-Error -----
8.	<code>Alpha a = new Iota(); ((Gamma)a).method0();</code>	Iota m0 x = Gamma -----
9.	<code>Iota a = new Iota(); a.method1();</code>	Gamma m1 x = Gamma Iota m0 x = Gamma -----
10.	<code>Eta a = new Eta(); a.method1();</code>	Gamma m1 x = Gamma Gamma2 m0 x = Gamma -----