

Koje klase će biti uspješno kompajlirane:

```
public class X1 {  
    public abstract void x1();  
}
```

```
abstract class X2 {  
    public void X2() {  
    }  
}
```

```
abstract class X3 {  
    public void m3();  
}
```

```
protected class X4 {  
    private final void m4(){};  
}
```

```
final class X5 {  
  
}
```

```
abstract class X6{  
    public final void m6(){};  
}
```

Validni identifikatori su:

`:x` `x#` `$x` `_x` `.x` `this` `super`

Koje klase će biti uspješno kompajlirane?

```
public class T1 {
    public abstract T1 metoda();
}

abstract class T2 {
    public void T2() { }
}

abstract class T3 {
    public void metoda();
}

private class T4 {
    private final void metoda(){};
}

final class T5 {
    final void metoda(){};
    void metoda2() { }
}

abstract class T6 {
    public final void metoda(){};
}

interface TI {
    private void metoda();
}

interface TI2 {
    interface TI3{
        void metoda();
    }
}

class T7{
    public void T7(){};
}

interface TI4 implements TI2{
    void metoda();
}
```

Analizirati kod i utvrditi da li se može kompajlirati i izvršiti. Ako kod nije moguće kompajlirati ili izvršiti, označiti „problematične“ linije koda i navesti razloge. Ako se kod može kompajlirati i izvršiti, napisati izlaz.

```
public class T8 {  
    int a = 10;  
  
    private static void metoda(int a) {  
        a -= 1;  
        System.out.println((a++));  
    }  
    public static void main(String args[]) {  
        T8 t = new T8();  
        T8.metoda(t!=null?3:1);  
    }  
}
```

Analizirati kod i utvrditi da li se može kompajlirati i izvršiti. Ako kod nije moguće kompajlirati ili izvršiti, označiti „problematične“ linije koda i navesti razloge. Ako se kod može kompajlirati i izvršiti, napisati izlaz.

```
public class B1 implements BI1, BI2 {  
    public B1() {  
        super();  
    }  
    public static void main(String[] args) {  
        BI1 b = new B1();  
    }  
    public final void metoda() {  
        System.out.println("1");  
    }  
    public int metoda() {  
        System.out.println("2");  
        return 1;  
    }  
}  
interface BI1 {  
    public abstract void metoda();  
}  
interface BI2 {  
    int metoda();  
}
```