

# Programování v jazyce C#

## 3. cvičení

<http://d3s.mff.cuni.cz/~jezek>

Department of  
Distributed and  
Dependable  
Systems



*Pavel Ježek*

*pavel.jezek@d3s.mff.cuni.cz*



CHARLES UNIVERSITY IN PRAGUE

faculty of mathematics and physics

# Inheritance and Contracts

```
interface I2 {  
    public void f();  
    public void g();  
}
```

```
class D1 : I2 {  
    public void f() {  
        Console.WriteLine("D1.f");  
    }  
  
    public void g() {  
        Console.WriteLine("D1.g");  
    }  
}
```

D1.f machine code

D1.g machine code

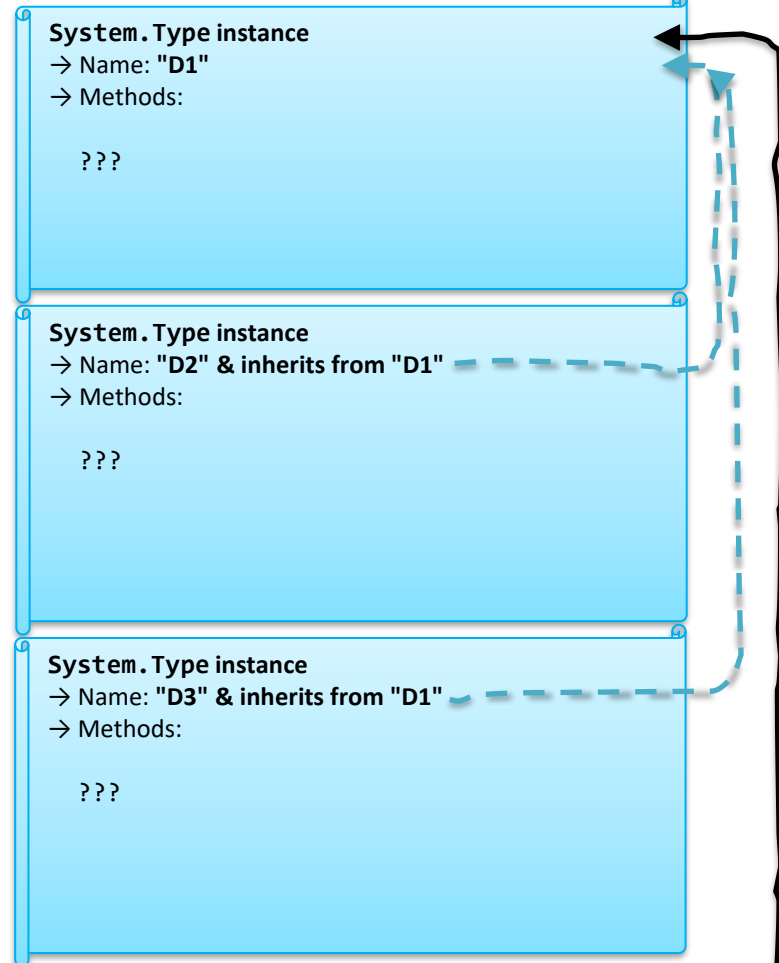
```
class D2 : D1 {  
    public new void g() {  
        Console.WriteLine("D2.g");  
    }  
}
```

D2.g machine code

```
class D3 : D1, I2 {  
    public new void g() {  
        Console.WriteLine("D3.g");  
    }  
}
```

D3.g machine code

```
I2 i2 = new D1();  
i2.f();  
i2.g();
```



# Inheritance and Contracts

```
interface I2 {  
    public void f();  
    public void g();  
}
```

```
class D1 : I2 {  
    public void f() {  
        Console.WriteLine("D1.f");  
    }  
  
    public void g() {  
        Console.WriteLine("D1.g");  
    }  
}
```

```
class D2 : D1 {  
    public new void g() {  
        Console.WriteLine("D2.g");  
    }  
}
```

```
class D3 : D1, I2 {  
    public new void g() {  
        Console.WriteLine("D3.g");  
    }  
}
```

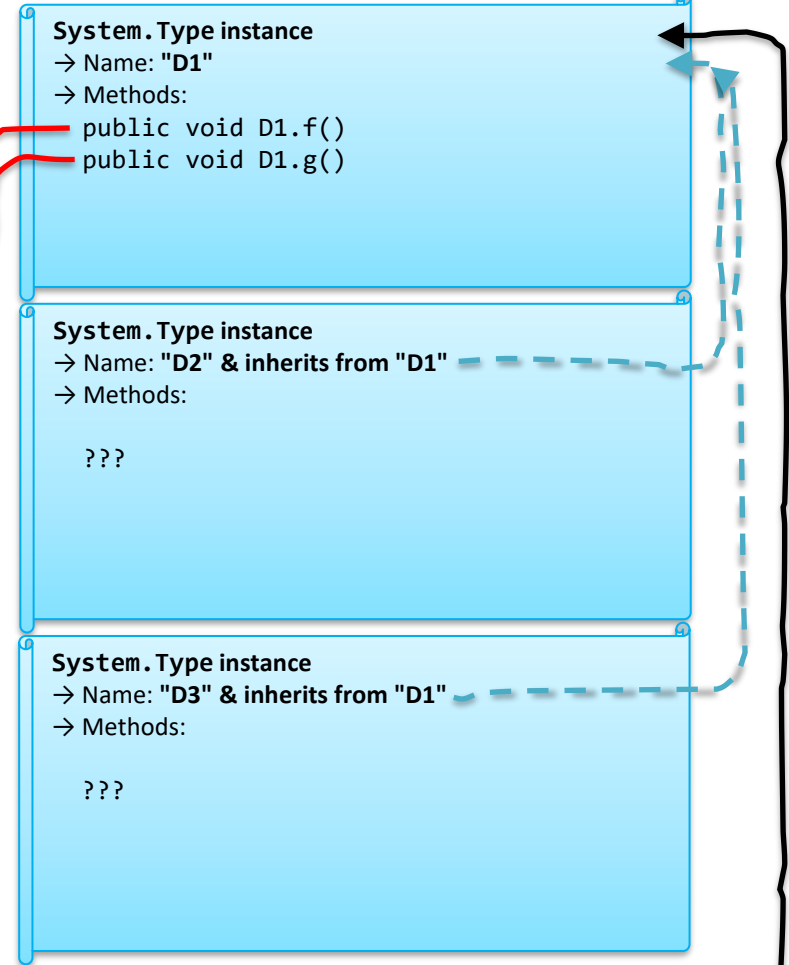
```
I2 i2 = new D1();  
i2.f();  
i2.g();
```

D1.f machine code

D1.g machine code

D2.g machine code

D3.g machine code



# Inheritance and Contracts

```
interface I2 {  
    public void f();  
    public void g();  
}
```

```
class D1 : I2 {  
    public void f() {  
        Console.WriteLine("D1.f");  
    }  
    public void g() {  
        Console.WriteLine("D1.g");  
    }  
}
```

```
class D2 : D1 {  
    public new void g() {  
        Console.WriteLine("D2.g");  
    }  
}
```

```
class D3 : D1, I2 {  
    public new void g() {  
        Console.WriteLine("D3.g");  
    }  
}
```

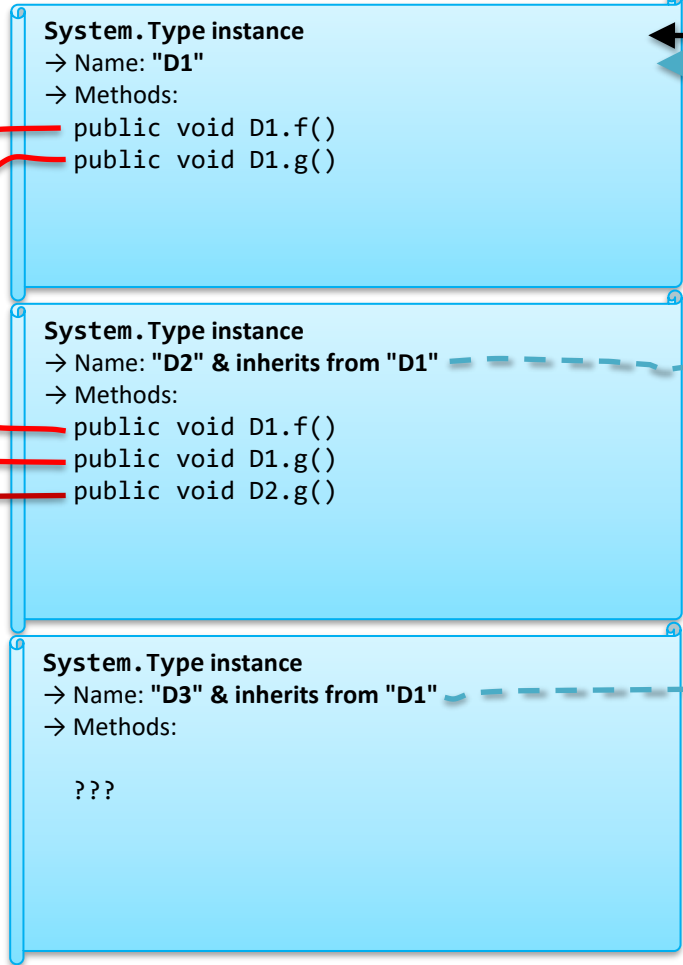
```
I2 i2 = new D1();  
i2.f();  
i2.g();
```

D1.f machine code

D1.g machine code

D2.g machine code

D3.g machine code



# Inheritance and Contracts

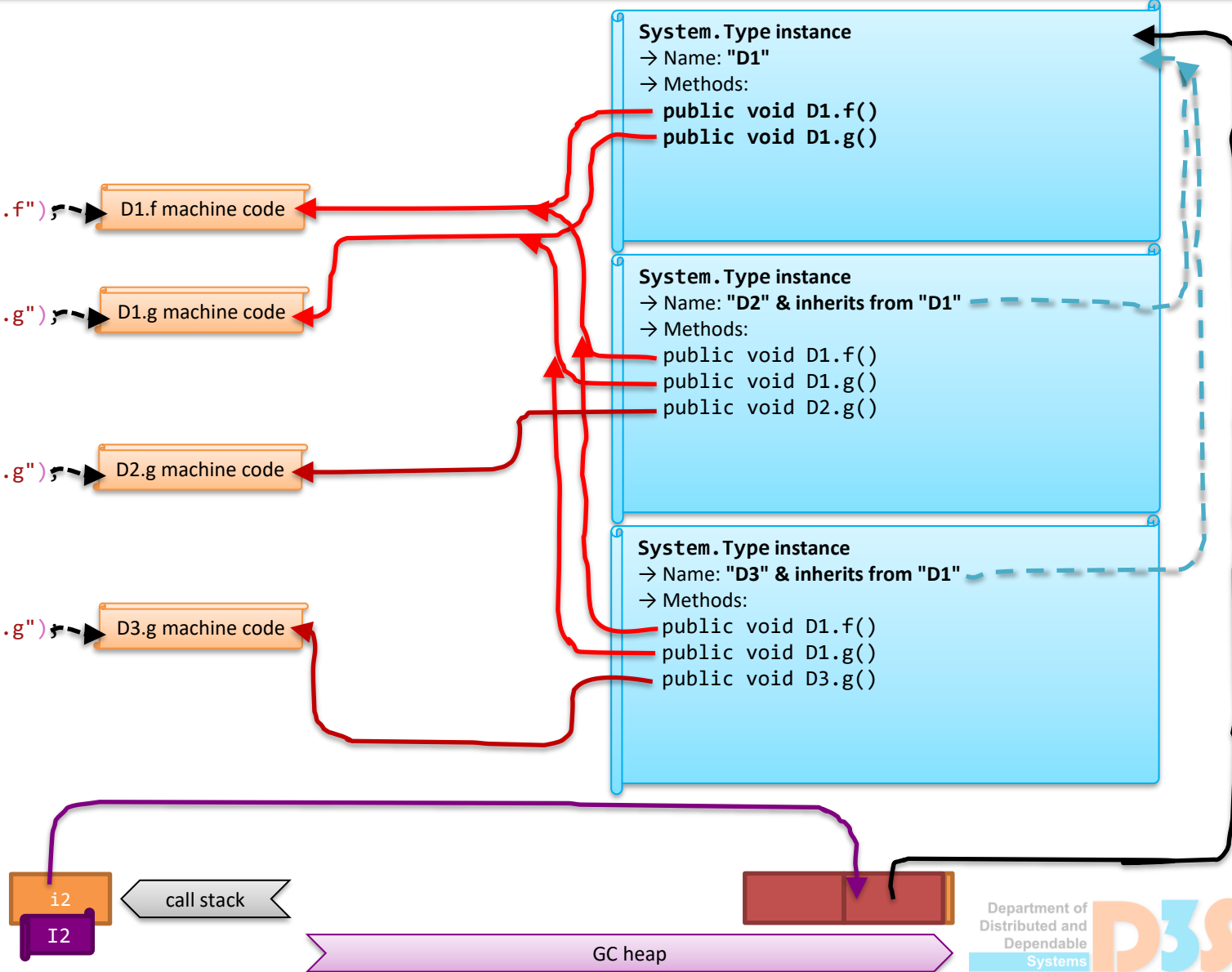
```
interface I2 {  
    public void f();  
    public void g();  
}
```

```
class D1 : I2 {  
    public void f() {  
        Console.WriteLine("D1.f");  
    }  
    public void g() {  
        Console.WriteLine("D1.g");  
    }  
}
```

```
class D2 : D1 {  
    public new void g() {  
        Console.WriteLine("D2.g");  
    }  
}
```

```
class D3 : D1, I2 {  
    public new void g() {  
        Console.WriteLine("D3.g");  
    }  
}
```

```
I2 i2 = new D1();  
i2.f();  
i2.g();
```



# Inheritance and Contracts

```
interface I2 {  
    public void f();  
    public void g();  
}
```

```
class D1 : I2 {  
    public void f() {  
        Console.WriteLine("D1.f");  
    }  
    public void g() {  
        Console.WriteLine("D1.g");  
    }  
}
```

```
class D2 : D1 {  
    public new void g() {  
        Console.WriteLine("D2.g");  
    }  
}
```

```
class D3 : D1, I2 {  
    public new void g() {  
        Console.WriteLine("D3.g");  
    }  
}
```

```
I2 i2 = new D1();  
i2.f();  
i2.g();
```

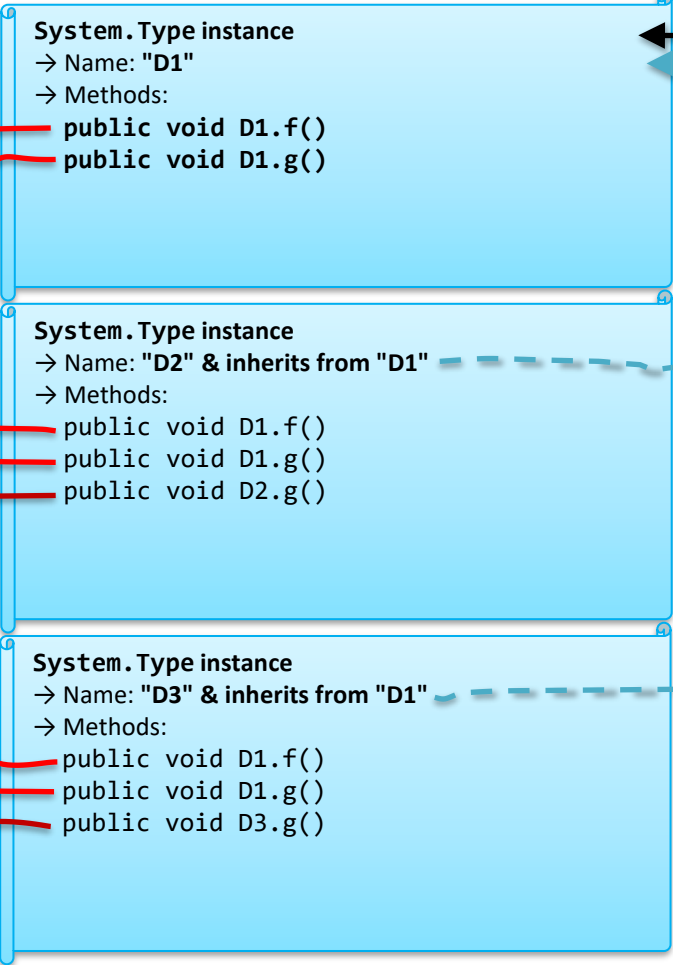
void f()  
void g()

D1.f machine code

D1.g machine code

D2.g machine code

D3.g machine code



# Inheritance and Contracts

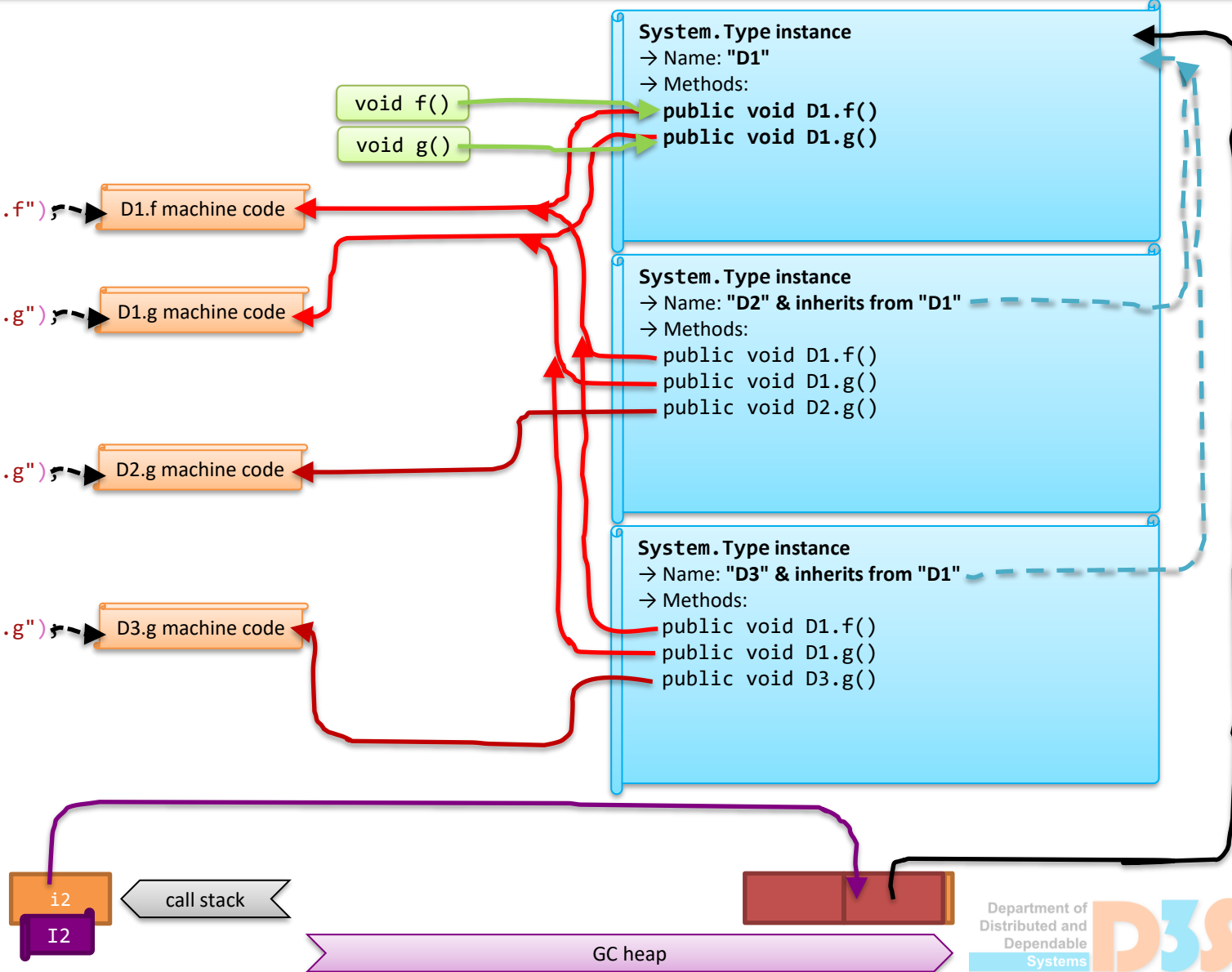
```
interface I2 {  
    public void f();  
    public void g();  
}
```

```
class D1 : I2 {  
    public void f() {  
        Console.WriteLine("D1.f");  
    }  
  
    public void g() {  
        Console.WriteLine("D1.g");  
    }  
}
```

```
class D2 : D1 {  
    public new void g() {  
        Console.WriteLine("D2.g");  
    }  
}
```

```
class D3 : D1, I2 {  
    public new void g() {  
        Console.WriteLine("D3.g");  
    }  
}
```

```
I2 i2 = new D1();  
i2.f();  
i2.g();
```



# Inheritance and Contracts

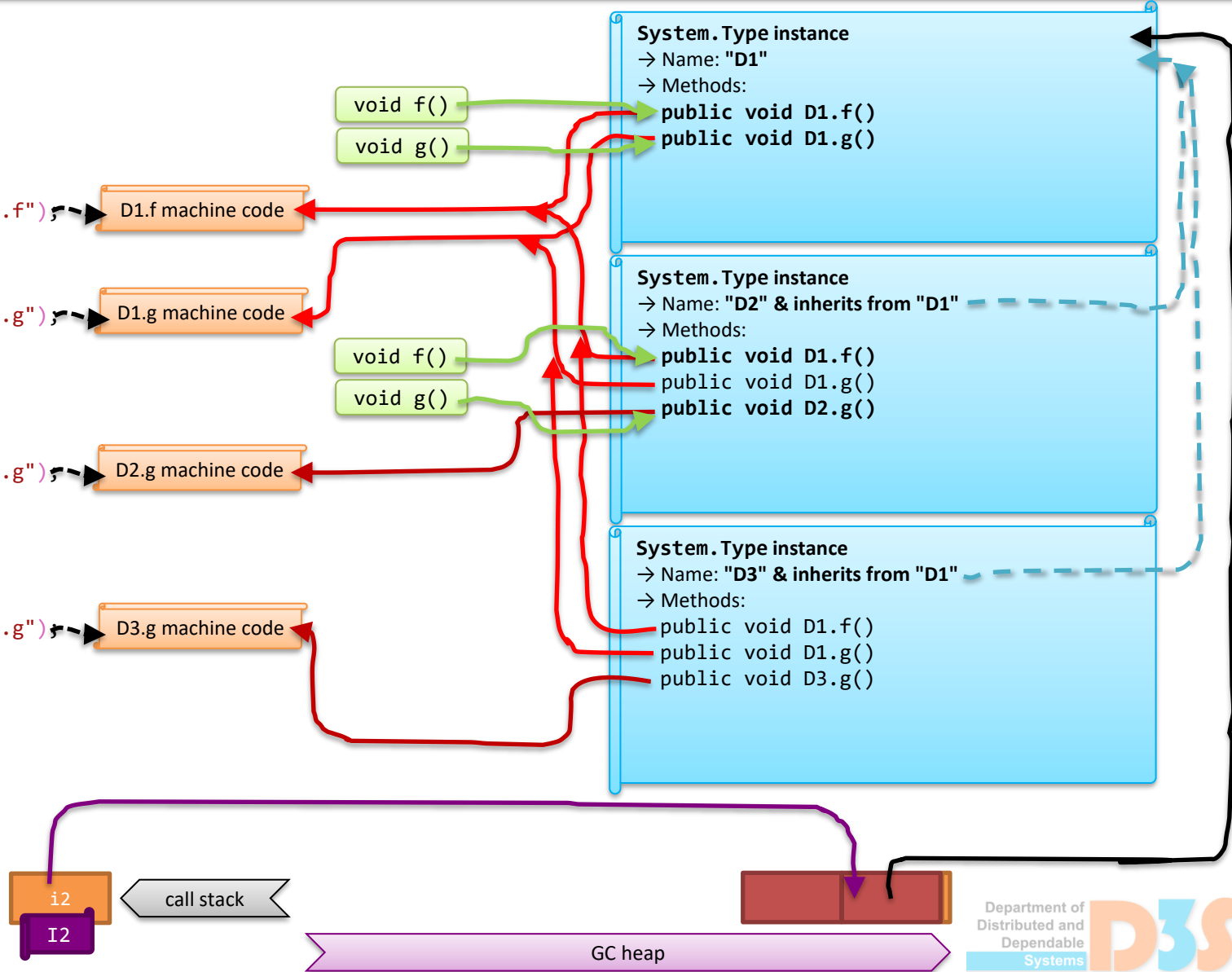
```
interface I2 {  
    public void f();  
    public void g();  
}
```

```
class D1 : I2 {  
    public void f() {  
        Console.WriteLine("D1.f");  
    }  
    public void g() {  
        Console.WriteLine("D1.g");  
    }  
}
```

```
class D2 : D1 {  
    public new void g() {  
        Console.WriteLine("D2.g");  
    }  
}
```

```
class D3 : D1, I2 {  
    public new void g() {  
        Console.WriteLine("D3.g");  
    }  
}
```

```
I2 i2 = new D1();  
i2.f();  
i2.g();
```





# Inheritance and Contracts

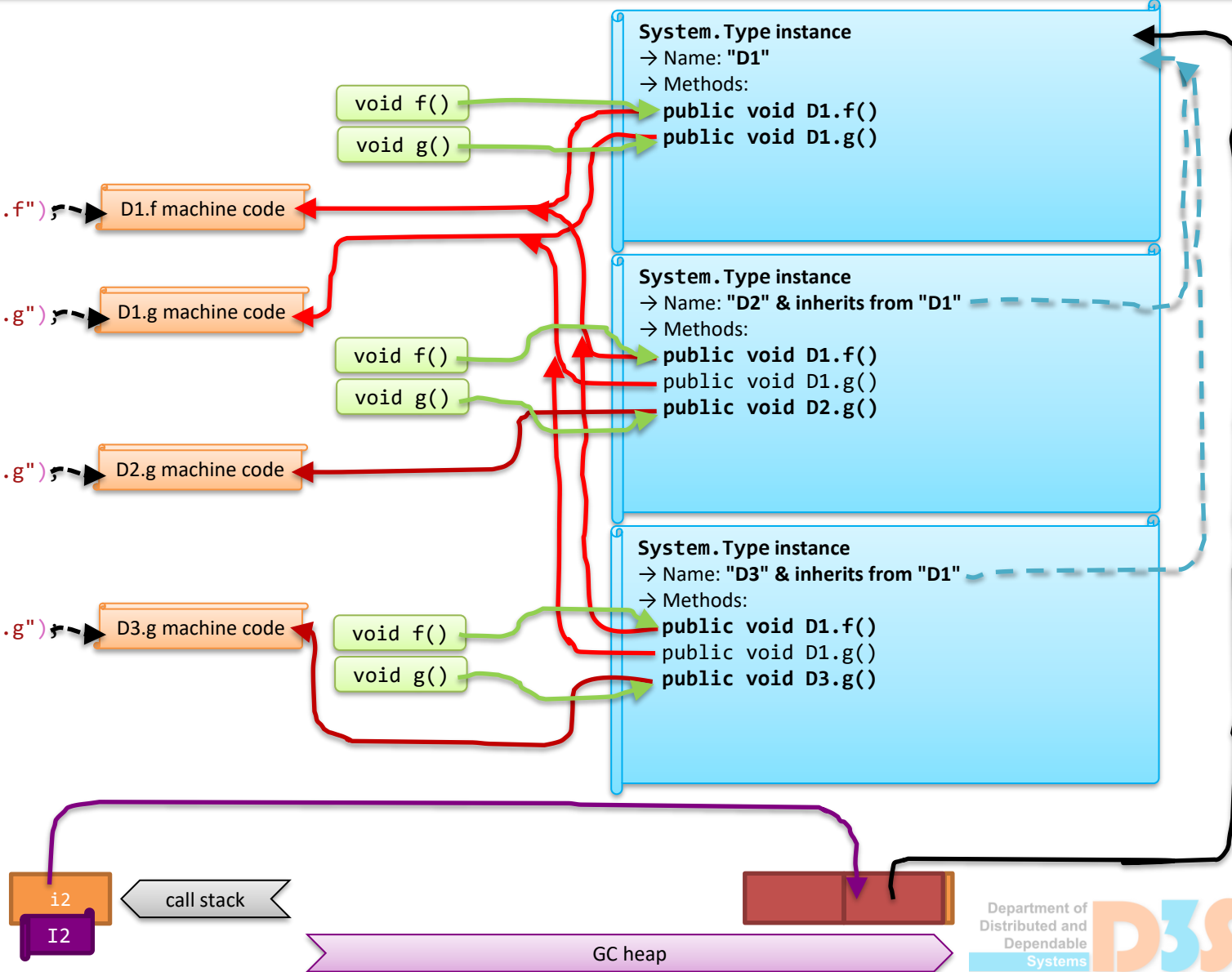
```
interface I2 {  
    public void f();  
    public void g();  
}
```

```
class D1 : I2 {  
    public void f() {  
        Console.WriteLine("D1.f");  
    }  
    public void g() {  
        Console.WriteLine("D1.g");  
    }  
}
```

```
class D2 : D1 {  
    public new void g() {  
        Console.WriteLine("D2.g");  
    }  
}
```

```
class D3 : D1, I2 {  
    public new void g() {  
        Console.WriteLine("D3.g");  
    }  
}
```

```
I2 i2 = new D1();  
i2.f();  
i2.g();
```



# Inheritance and Contracts

```
interface I2 {  
    public void f();  
    public void g();  
}
```

```
class D1 : I2 {  
    public void f() {  
        Console.WriteLine("D1.f");  
    }  
    public void g() {  
        Console.WriteLine("D1.g");  
    }  
}
```

```
class D2 : D1 {  
    public new void g() {  
        Console.WriteLine("D2.g");  
    }  
}
```

```
class D3 : D1, I2 {  
    public new void g() {  
        Console.WriteLine("D3.g");  
    }  
}
```

```
I2 i2 = new D1();  
i2.f();  
i2.g();
```

System.Type instance

→ Name: "I2"

→ Methods:

public void I2.f()  
public void I2.g()

System.Type instance

→ Name: "D1"

→ Methods:

public void D1.f()  
public void D1.g()

System.Type instance

→ Name: "D2" & inherits from "D1"

→ Methods:

public void D1.f()  
public void D1.g()  
public void D2.g()

System.Type instance

→ Name: "D3" & inherits from "D1"

→ Methods:

public void D1.f()  
public void D1.g()  
public void D3.g()

void f()

void g()

void f()

void g()

void f()

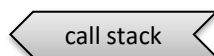
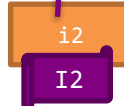
void g()

D1.f machine code

D1.g machine code

D2.g machine code

D3.g machine code



# Inheritance and Contracts

```
interface I2 {
    public void f();
    public void g();
}
```

```
class D1 : I2 {
    public void f() {
        Console.WriteLine("D1.f");
    }
}
```

```
public void g() {
    Console.WriteLine("D1.g");
}
```

```
class D2 : D1 {
    public new void g() {
        Console.WriteLine("D2.g");
    }
}
```

```
class D3 : D1, I2 {
    public new void g() {
        Console.WriteLine("D3.g");
    }
}
```

```
I2 i2 = new D1();
i2.f();
i2.g();
```

**System.Type instance**  
 → Name: "I2"  
 → Methods:  
 public void I2.f()  
 public void I2.g()

**System.Type instance**  
 → Name: "D1"  
 → Methods:  
 public void D1.f()  
 public void D1.g()  
 → Interface I1:  
 public void I2.f() 0:  
 public void I2.g() 1:

**System.Type instance**  
 → Name: "D2" & inherits from "D1"  
 → Methods:  
 public void D1.f()  
 public void D1.g()  
 public void D2.g()  
 → Interface I1:  
 public void I2.f() 0:  
 public void I2.g() 1:

**System.Type instance**  
 → Name: "D3" & inherits from "D1"  
 → Methods:  
 public void D1.f()  
 public void D1.g()  
 public void D3.g()  
 → Interface I1:  
 public void I2.f() 0:  
 public void I2.g() 1:

void f()

void g()

void f()

void g()

void f()

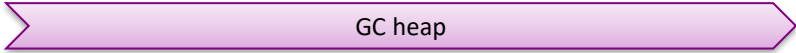
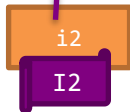
void g()

D1.f machine code

D1.g machine code

D2.g machine code

D3.g machine code



# Inheritance and Contracts

```
interface I2 {
    public void f();
    public void g();
}
```

```
class D1 : I2 {
    public void f() {
        Console.WriteLine("D1.f");
    }
}
```

```
public void g() {
    Console.WriteLine("D1.g");
}
```

```
class D2 : D1 {
    public new void g() {
        Console.WriteLine("D2.g");
    }
}
```

```
class D3 : D1, I2 {
    public new void g() {
        Console.WriteLine("D3.g");
    }
}
```

```
I2 i2 = new D1();
i2.f();
i2.g();
```

**System.Type instance**  
 → Name: "I2"  
 → Methods:  
 public void I2.f()  
 public void I2.g()

**System.Type instance**  
 → Name: "D1"  
 → Methods:  
 public void D1.f()  
 public void D1.g()  
 → Interface I1:  
 public void I2.f() 0:  
 public void I2.g() 1:

**System.Type instance**  
 → Name: "D2" & inherits from "D1"  
 → Methods:  
 public void D1.f()  
 public void D1.g()  
 public void D2.g()  
 → Interface I1:  
 public void I2.f() 0:  
 public void I2.g() 1:

**System.Type instance**  
 → Name: "D3" & inherits from "D1"  
 → Methods:  
 public void D1.f()  
 public void D1.g()  
 public void D3.g()  
 → Interface I1:  
 public void I2.f() 0:  
 public void I2.g() 1:

void f()

void g()

void f()

void g()

void f()

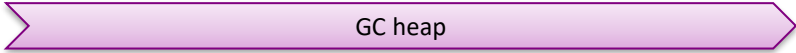
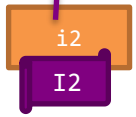
void g()

D1.f machine code

D1.g machine code

D2.g machine code

D3.g machine code



# Inheritance and Contracts

```
interface I2 {
    public void f();
    public void g();
}
```

```
class D1 : I2 {
    public void f() {
        Console.WriteLine("D1.f");
    }
}
```

```
public void g() {
    Console.WriteLine("D1.g");
}
```

```
class D2 : D1 {
    public new void g() {
        Console.WriteLine("D2.g");
    }
}
```

```
class D3 : D1, I2 {
    public new void g() {
        Console.WriteLine("D3.g");
    }
}
```

```
I2 i2 = new D1();
i2.f();
i2.g();
```

**System.Type instance**  
 → Name: "I2"  
 → Methods:  
 public void I2.f()  
 public void I2.g()

**System.Type instance**  
 → Name: "D1"  
 → Methods:  
 public void D1.f()  
 public void D1.g()  
 → Interface I1:  
 public void I2.f() 0:  
 public void I2.g() 1:

**System.Type instance**  
 → Name: "D2" & inherits from "D1"  
 → Methods:  
 public void D1.f()  
 public void D1.g()  
 public void D2.g()  
 → Interface I1:  
 public void I2.f() 0:  
 public void I2.g() 1:

**System.Type instance**  
 → Name: "D3" & inherits from "D1"  
 → Methods:  
 public void D1.f()  
 public void D1.g()  
 public void D3.g()  
 → Interface I1:  
 public void I2.f() 0:  
 public void I2.g() 1:

void f()  
 void g()

void f()  
 void g()

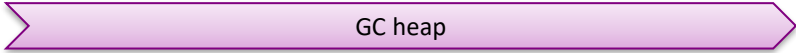
void f()  
 void g()

D1.f machine code

D1.g machine code

D2.g machine code

D3.g machine code



# Inheritance and Contracts

```
interface I2 {
    public void f();
    public void g();
}
```

```
class D1 : I2 {
    public void f() {
        Console.WriteLine("D1.f");
    }
}
```

```
public void g() {
    Console.WriteLine("D1.g");
}
```

```
class D2 : D1 {
    public new void g() {
        Console.WriteLine("D2.g");
    }
}
```

```
class D3 : D1, I2 {
    public new void g() {
        Console.WriteLine("D3.g");
    }
}
```

```
I2 i2 = new D1();
i2.f();
i2.g();
```

**System.Type instance**

→ Name: "I2"

→ Methods:  
 public void I2.f()  
 public void I2.g()

void f()

void g()

D1.f machine code

D1.g machine code

D2.g machine code

D3.g machine code

void f()

void g()

void f()

void g()

**System.Type instance**

→ Name: "D1"

→ Methods:  
 public void D1.f()  
 public void D1.g()

→ Interface I1:  
 public void I2.f() 0:  
 public void I2.g() 1:

**System.Type instance**

→ Name: "D2" & inherits from "D1"

→ Methods:  
 public void D1.f()  
 public void D1.g()  
 public void D2.g()

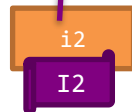
→ Interface I1:  
 public void I2.f() 0:  
 public void I2.g() 1:

**System.Type instance**

→ Name: "D3" & inherits from "D1"

→ Methods:  
 public void D1.f()  
 public void D1.g()  
 public void D3.g()

→ Interface I1:  
 public void I2.f() 0:  
 public void I2.g() 1:



# Inheritance and Contracts

```
interface I2 {
    public void f();
    public void g();
}
```

```
class D1 : I2 {
    public void f() {
        Console.WriteLine("D1.f");
    }
}
```

```
public void g() {
    Console.WriteLine("D1.g");
}
```

```
class D2 : D1 {
    public new void g() {
        Console.WriteLine("D2.g");
    }
}
```

```
class D3 : D1, I2 {
    public new void g() {
        Console.WriteLine("D3.g");
    }
}
```

```
I2 i2 = new D1();
i2.f();
i2.g();
```

```
D2 d2 = new D2();
d2.f();
d2.g();
```

**System.Type instance**  
 → Name: "I2"  
 → Methods:  
 public void I2.f()  
 public void I2.g()

**System.Type instance**  
 → Name: "D1"  
 → Methods:  
 public void D1.f()  
 public void D1.g()  
 → Interface I1:  
 public void I2.f() 0:  
 public void I2.g() 1:

**System.Type instance**  
 → Name: "D2" & inherits from "D1"  
 → Methods:  
 public void D1.f()  
 public void D1.g()  
 public void D2.g()  
 → Interface I1:  
 public void I2.f() 0:  
 public void I2.g() 1:

**System.Type instance**  
 → Name: "D3" & inherits from "D1"  
 → Methods:  
 public void D1.f()  
 public void D1.g()  
 public void D3.g()  
 → Interface I1:  
 public void I2.f() 0:  
 public void I2.g() 1:

void f()  
void g()

void f()  
void g()

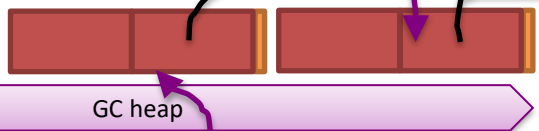
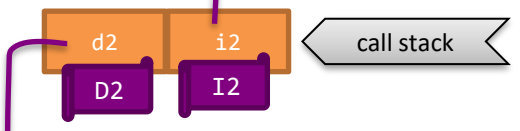
void f()  
void g()

D1.f machine code

D1.g machine code

D2.g machine code

D3.g machine code





# Inheritance and Contracts

```
interface I2 {
    public void f();
    public void g();
}
```

```
class D1 : I2 {
    public void f() {
        Console.WriteLine("D1.f");
    }
}
```

```
public void g() {
    Console.WriteLine("D1.g");
}
```

```
class D2 : D1 {
    public new void g() {
        Console.WriteLine("D2.g");
    }
}
```

```
class D3 : D1, I2 {
    public new void g() {
        Console.WriteLine("D3.g");
    }
}
```

```
I2 i2 = ...;
...
```

```
D2 d2 = new D2();
d2.f();
d2.g();
```

```
i2 = d2;
i2.f();
i2.g();
```

**System.Type instance**  
 → Name: "I2"  
 → Methods:  
 public void I2.f()  
 public void I2.g()

**System.Type instance**  
 → Name: "D1"  
 → Methods:  
 public void D1.f()  
 public void D1.g()  
 → Interface I1:  
 public void I2.f() 0:  
 public void I2.g() 1:

**System.Type instance**  
 → Name: "D2" & inherits from "D1"  
 → Methods:  
 public void D1.f()  
 public void D1.g()  
 public void D2.g()  
 → Interface I1:  
 public void I2.f() 0:  
 public void I2.g() 1:

**System.Type instance**  
 → Name: "D3" & inherits from "D1"  
 → Methods:  
 public void D1.f()  
 public void D1.g()  
 public void D3.g()  
 → Interface I1:  
 public void I2.f() 0:  
 public void I2.g() 1:

void f()  
 void g()

void f()  
 void g()

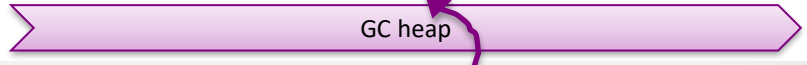
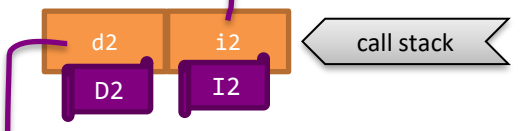
void f()  
 void g()

D1.f machine code

D1.g machine code

D2.g machine code

D3.g machine code





# Inheritance and Contracts

```
interface I2 {
    public void f();
    public void g();
}
```

```
class D1 : I2 {
    public void f() {
        Console.WriteLine("D1.f");
    }
}
```

```
public void g() {
    Console.WriteLine("D1.g");
}
```

```
class D2 : D1 {
    public new void g() {
        Console.WriteLine("D2.g");
    }
}
```

```
class D3 : D1, I2 {
    public new void g() {
        Console.WriteLine("D3.g");
    }
}
```

```
I2 i2 = ...;
...
```

```
D2 d2 = new D2();
...
```

```
i2 = new D3();
i2.f();
i2.g();
```

**System.Type instance**  
 → Name: "I2"  
 → Methods:  
 public void I2.f()  
 public void I2.g()

**System.Type instance**  
 → Name: "D1"  
 → Methods:  
 public void D1.f()  
 public void D1.g()  
 → Interface I1:  
 public void I2.f() 0:  
 public void I2.g() 1:

**System.Type instance**  
 → Name: "D2" & inherits from "D1"  
 → Methods:  
 public void D1.f()  
 public void D1.g()  
 public void D2.g()  
 → Interface I1:  
 public void I2.f() 0:  
 public void I2.g() 1:

**System.Type instance**  
 → Name: "D3" & inherits from "D1"  
 → Methods:  
 public void D1.f()  
 public void D1.g()  
 public void D3.g()  
 → Interface I1:  
 public void I2.f() 0:  
 public void I2.g() 1:

void f()  
 void g()

void f()  
 void g()

void f()  
 void g()

D1.f machine code

D1.g machine code

D2.g machine code

D3.g machine code

