

Fachhochschule Regensburg / Continental AG Dr. Christian Flug Dr. Peter Jüttner	
Vorlesung Software Engineering	SS 2008
Trockenübung	Termin 28.03.2008

## Lösung zur Übungsaufgabe Objektorientierte Analyse Mehrfachvererbung

### Beschreibung:

Mehrfachvererbung wird gelegentlich benutzt für die Darstellung von Ein-/Ausgabeverhalten mit gemeinsamen Anteilen. Dies kann eine Reihe von Problemen hervorrufen.

### Aufgabe:

Überlegen Sie sich ein geeignetes Beispiel einer Klassenhierarchie mit mindestens einem Klassenattribut und einer Methode.

Erläutern Sie potentielle Probleme und deren Lösungsansätze.

Erläutern Sie auch die verschiedenen Ansätze mindestens zweier OO-Programmiersprachen (etwa C++ und Java)

## Lösung

### Java, 1. Versuch:

```
class Port
{
    virtual private int port_nr;
    virtual public int get_port_nr();
};

class InputPort extends Port
{
    private int port_nr;
    public int get_port_nr();
};

class OutputPort extends Port
{
    private int port_nr;
    public int get_port_nr();
};
```

```

class InOutPort extends InputPort, OutputPort // geht in Java nicht
// Alternative: 3 Subklassen unter Port oder Realisierung über Interfaces
{
    private int port_nr;
    public int get_port_nr();
};

public static main()
{
    InOutPort myIOPort = new IOPort;
    ...
    myIOPort.get_port_nr();
}

```

### Java, 2. Versuch:

```

class Port
{
    virtual private int port_nr;
    virtual public int get_port_nr();
};

interface InputPort
{
    int get_port_nr();
};

interface OutputPort
{
    int get_port_nr();
};

class InOutPort extends Port
    implements InputPort, OutputPort
{
    private int port_nr;
    public int get_port_nr();
};

public static main()
{
    InOutPort myIOPort = new IOPort;
    ...
    myIOPort.get_port_nr();
}

```

### C++:

```

class Port
{

```

```

    virtual private int port_nr;

    virtual public int get_port_nr();
};

class InputPort : public Port
{
    private int port_nr;

    virtual public int get_port_nr();
};

class OutputPort : public Port
{
    private int port_nr;

    virtual public int get_port_nr();
};

class InOutPort : public InputPort, public OutputPort // Mehrfachvererbung
{
    ...
};

public static main()
{
    InOutPort* myIOPort = new IOPort;
    ...
    myIOPort->get_port_nr(); //error, da nicht eindeutig

    myIOPort::OutputPort->get_port_nr(); //paßt semantisch nicht unbedingt
    ((OutputPort*)myIOPort->get_port_nr(); // Alternative zu oben

    // Lösung: explizite Methode get_port_nr() in InOutPort definieren
}

```