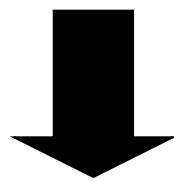
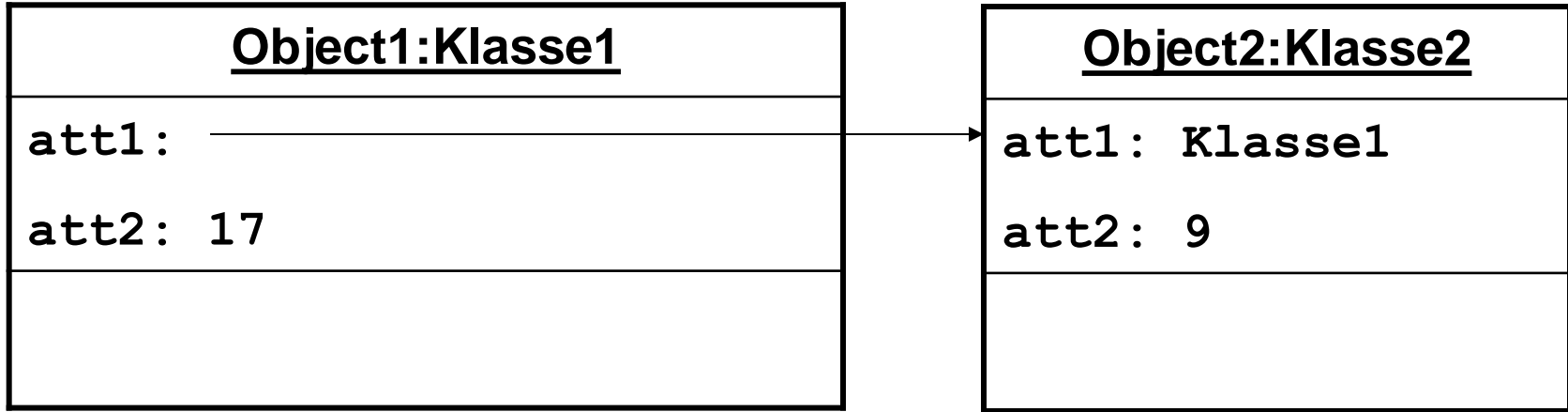


Problemstellung

Objekt mit aggregierten Objekten:



serialisiert als Datei:



- Speicherung von Objekten
- Netzwerkübertragung von Objekten

write-Methoden in ObjectOutputStream

```
public final void writeObject(Object obj)
    throws IOException
public void writeBoolean(boolean data)
    throws IOException
public void writeByte(int data)
    throws IOException
public void writeShort(int data)
    throws IOException
public void writeChar(int data)
    throws IOException
public void writeInt(int data)
    throws IOException
public void writeLong(long data)
    throws IOException
public void writeFloat(float data)
    throws IOException
public void writeDouble(double data)
    throws IOException
```

Bsp.: Anwendung einer write-Methode

gegeben sei folgender ObjectOutputStream:

<code>true</code>	<code>17.3</code>	<code>'H'</code>			
-------------------	-------------------	------------------	--	--	--

Ausführung der Methode `writeBoolean(true)` liefert:

<code>true</code>	<code>17.3</code>	<code>'H'</code>	<code>true</code>		
-------------------	-------------------	------------------	-------------------	--	--

Bsp.: Die Klasse Zeit

```
import java.io.*;

public class Zeit implements Serializable {

    private int stunde;
    private int minute;

    public Zeit(int stunde, int minute) {
        this.stunde = stunde;
        this.minute = minute;
    }

    public String toString() {
        return stunde + ":" + minute;
    }
}
```

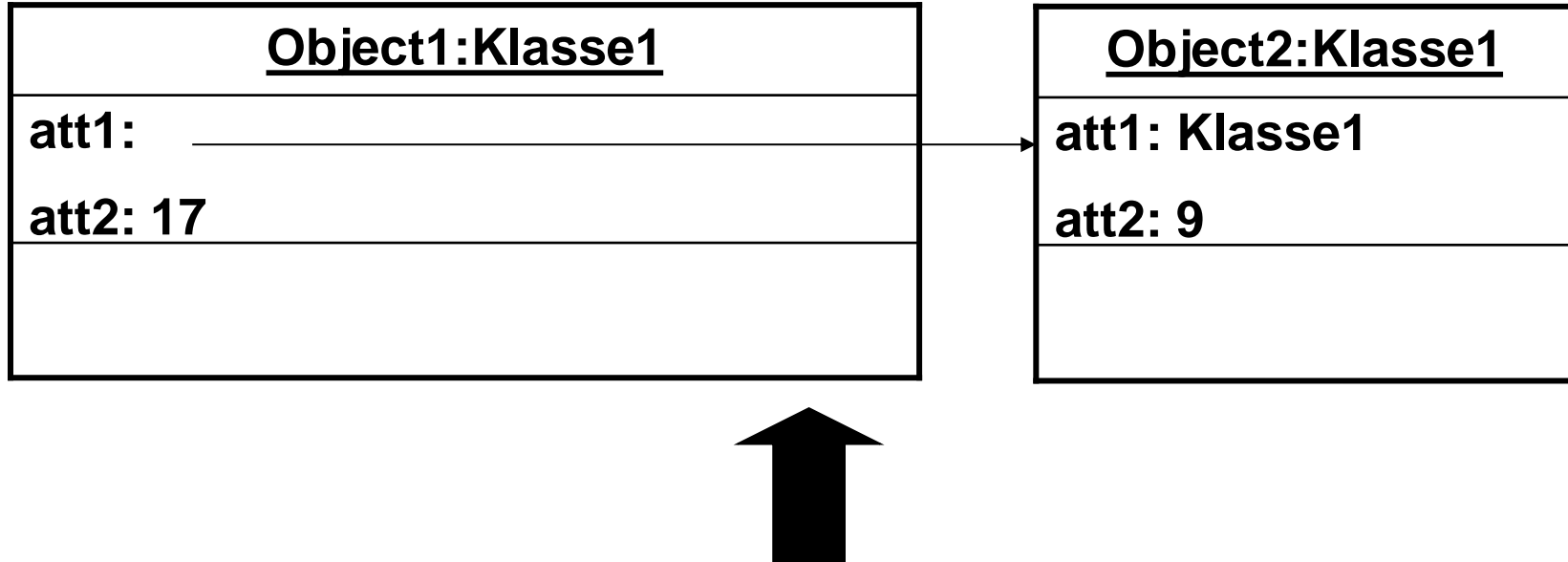
Serialisierung eines Objekts von Zeit

```
import java.io.*;

public class Serialisiere {
    public static void main(String[] args) {
        try {
            FileOutputStream fs = new FileOutputStream("fos.s");
            ObjectOutputStream os = new ObjectOutputStream(fs);
            Zeit zeit = new Zeit(10,20);
            os.writeObject(zeit);
            os.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Problemstellung

Objekt mit aggregierten Objekten:



Datei mit serialisierten Objekten:

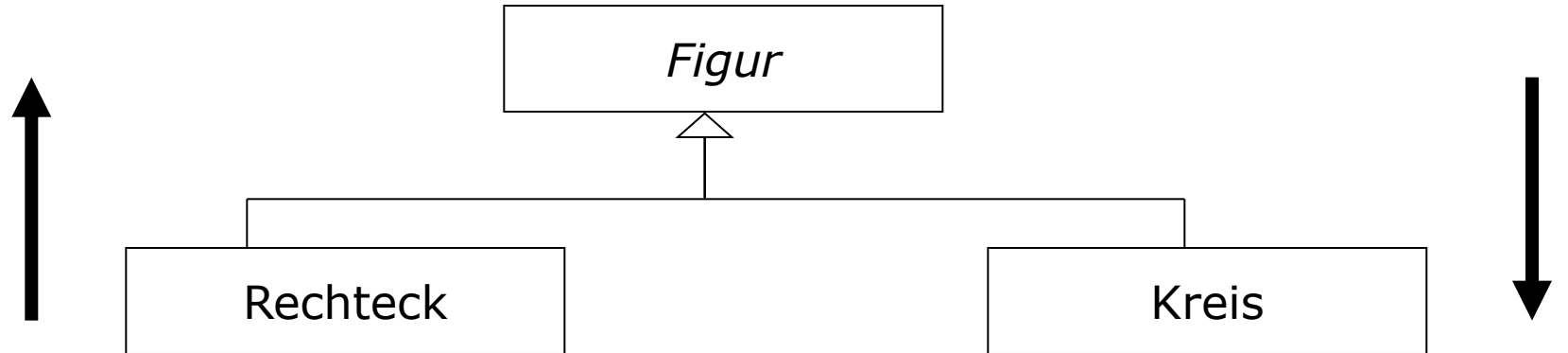
Klasse1	att1	att2	Klasse2	att1	att2
---------	------	------	---------	------	------

- Rekonstruktion von Objekten aus Dateien
- Netzwerkübertragung von Objekten

read-Methoden in ObjectInputStream

```
public final Object readObject()
    throws IOException, OptionalDataException,
           ClassNotFoundException
public boolean readBoolean()
    throws IOException
public byte readByte()
    throws IOException
public short readShort()
    throws IOException
public char readChar()
    throws IOException
public int readInt()
    throws IOException
public long readLong()
    throws IOException
public float readFloat()
    throws IOException
public double readDouble()
    throws IOException
```

Casting



```
Figur[] figuren = new Figur[4];
```

UpCast: Zuweisung eines Rechtecks (Kreises) an Figur

```
figuren[1] = new Rechteck(1,1,1,1);
```

```
figuren[2] = new Kreis(1,1,1);
```

DownCast: Zuweisung einer Figur an Rechteck (Kreis)

```
Rechteck rechteck = (Rechteck) figuren[1];
```

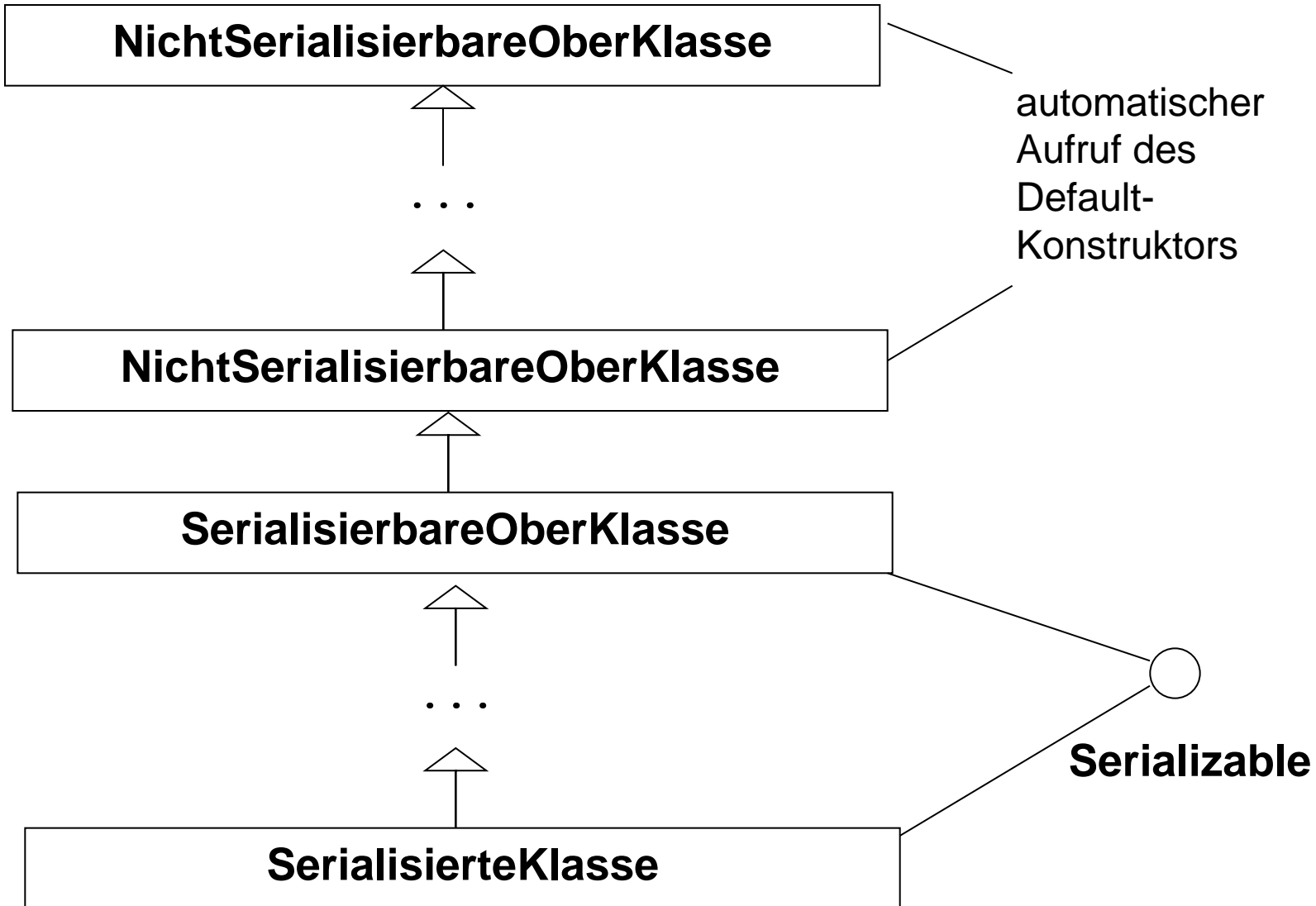
```
Kreis kreis = (Kreis) figuren[2];
```


Deserialisierung eines Objekts von Zeit

```
import java.io.*;

public class Deserialisiere {
    public static void main(String[] args) {
        try {
            FileInputStream fs = new FileInputStream("fos.s");
            ObjectInputStream is = new ObjectInputStream(fs);
            Zeit zeit = (Zeit) is.readObject();
            System.out.println(zeit);
            is.close();
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Deserialisieren in Vererbungshierarchien



Überlagerung von equals

- **bisher: equals überladen**

```
boolean equals(Zeit zeit) {  
    return(    this.stunde == zeit.stunde  
            &&  this.minute == zeit.minute );  
}
```

- **muss aber überlagert werden, da von Java-Bibs aufgerufen**

```
boolean equals(Object zeit) {  
    return(    zeit != null  
            && zeit instanceof Zeit  
            && this.stunde == ((Zeit) zeit).stunde  
            && this.minute == ((Zeit) zeit).minute);  
}
```