// weiteres Beispiel zur abstrakten Klasse

// ( zu Seite 40 )

**public** **class** Abteilung

{

// Attribute

**private** **int** abtnr;

**private** String bezeichnung;

// Konstruktor

**public** Abteilung ( **int** abtnr, String bezeichnung )

{

**this**.abtnr = abtnr;

**this**.bezeichnung = bezeichnung;

}

// weitere Methoden

**public** **int** getAbteilungsNummer ( )

{

**return** **this**.abtnr;

}

**public** String getAbteilungsBezeichnung ( )

{

**return** **this**.bezeichnung;

}

**public** String toString ( )

{

**return** "Abteilungsnummer : " + **this**.abtnr +

" Bezeichnung : " + **this**.bezeichnung + " ";

}

} // end class Abteilung

/////////////////////////////////////////////////////

**public** **abstract** **class** Mitarbeiter // einfuegen

{

// Attribute

**private** **int** persnr;

**private** String name;

// einfuegen

**private** Abteilung abteilung;

// Konstruktor

**public** Mitarbeiter ( **int** persnr, String name, Abteilung abteilung )

{

**this**.persnr = persnr;

**this**.name = name;

// einfuegen

}

// weitere Methoden

**public** **int** getPersNr ( )

{

**return** **this**.persnr;

}

**public** String getName ( )

{

**return** **this**.name;

}

**public** String getAbteilungsBezeichnung ( )

{

**return** **this**.abteilung.getAbteilungsBezeichnung ( );

}

**public** **int** getAbteilungsNummer ( )

{

**return** **this**.abteilung.getAbteilungsNummer( );

}

**public** String toString ( )

{

**return** " PersNr: " + **this**.persnr + " Name : " + **this**.name +

" Abteilung: " + **this**.abteilung;

}

// abstrakte Methode" getAuszahlungsbetrag ( )" einfuegen

} // end class Mitarbeiter

///////////////////////////////////////////////////

**public** **class** Angestellter **extends** Mitarbeiter

{

// Attribut

// einfuegen

// Konstruktor

**public** Angestellter // einfuegen

{

// einfuegen

}

// weitere Methoden

**public** **double** getGehalt ()

{

**return** **this**.gehalt;

}

**public** String getName ( )

{

**return** getName ( );

}

**public** String toString ( )

{

// einfuegen

}

**public** **double** getAuszahlungsbetrag ( )

{

// einfuegen

}

} // end class Angestellter

//////////////////////////////////////////////

**public** **class** Arbeiter **extends** Mitarbeiter

{

// Attribute

**private** **double** stdSatz;

**private** **double** anzStunden;

// Konstruktor

Arbeiter ( **int** persnr, String name, Abteilung abteilung ,

**double** stdSatz, **double** anzStunden )

{

**super** ( persnr, name, abteilung );

**this**.stdSatz = stdSatz;

**this**.anzStunden = anzStunden ;

}

// weitere Methoden

**public** **void** aendereStdSatz ( **double** stdSatz )

{

**this**.stdSatz = stdSatz;

}

**public** **void** aendereAnzStunden ( **double** anzStunden )

{

**this**.anzStunden = anzStunden;

}

**public** String toString ( )

{

// einfuegen

}

**private** **double** lohnBerechnen ( )

{

**return** **this**.stdSatz \* **this**.anzStunden;

}

**public** **double** getAuszahlungsbetrag ( )

{

/7 einfuegen

}

} // end class Arbeiter

//////////////////////////////////////////////////

**import** java.util.\*;

**public** **class** Anw\_abstrakt\_Ma

{

**public** **static** **void** main(String[] args)

{

Abteilung ab1 = **new** Abteilung ( 1, "edv");

Abteilung ab2 = **new** Abteilung ( 2, "verkauf");

Abteilung ab3 = **new** Abteilung ( 3, "werkstatt");

/\*

Von abstrakten Klassen kann kein Objekt gebildet werden

Mitarbeiter a1 = new Mitarbeiter ( 1, "max", ab2 );

\*/

Angestellter a2 = **new** Angestellter ( 2, "eva", ab1, 5000.0 );

Arbeiter a3 = **new** Arbeiter ( 3, "otto", ab3, 25.0, 40.0 );

// Polymorphie

// einfuegen und alle asgebn

}// end main

} // end class Anwendung