Die folgenden 3 Kapitel objektorientiertes Programmieren mit JAVA, PHP und JAVASCRIPT zeigen, dass es sich sozusagen um das gleiche Prinzip handelt. Damit dies klarer wird wurde möglichst auf eigene Konstruktoren und Zugriffsspezifizierer verzichtet.
Der Hauptgrund für diese Technik ist eine verbesserte Wiederverwendbarkeit und Sicherheit von Funktionen. Allerdings erscheint dies zunächst etwas umständlich, da die Funktionen nicht direkt erstellt werden können.
Man muss zunächst eine Klasse bilden, die dann einen eigenen Variablentyp darstellt. Dieser enthält dann nicht nur die Funktionen, die jetzt Methoden heißen sondern auch die Variablen der Funktionen, die jetzt Attribute heißen. Innerhalb der Methoden kann man lokale Variable benutzen.
Im Hauptprogramm werden dann Objekte dieses Typs gebildet. Jetzt kann man die Attribute ändern und die Methoden anwenden.
Dass man die mit der objektorientierten Programmierung die Wirklichkeit besonders gut abbilden kann ist ein wichtiger Nebenaspekt, aber nicht der Grund für objektorientiertes Programmieren.
Wozu solll man 3 Programmiersprachen lernen?
Keine der Programmiersprachen kann so ohne weiteres angewandt werden (z.B. per Doppelklick). Generell handelt es sich zunächst um Texte, die bei JAVA unter *.java, bei PHP unter *.php und bei JAVASCRIPT unter *.html gespeichert werden.
Bei JAVA wandelt ein compiler die *.java Datei in eine *.class Datei um, die dann über ein spezielles Programm gestartet werden kann.
Bei PHP benötigt man einen webserver, der die Datei interpretiert und an einen browser sendet.
JAVASCRIPT wird direkt vom browser ausgeführt.
PHP und JAVASCRIPT werden für webseiten verwendet, JAVA kann sozusagen alles und ist auch am genauesten als Programmiersprache und ideal zum Erlernen des objektorientierten Programmierens, insbesondere, wenn man das Entwicklungssystem eclipse verwendet.
JAVASCRIPT kann keine Dateioperationen ausführen, eignet sich aber für interaktive Spiele.
Wer das JAVA-Kapitel verstanden hat, versteht auch die anderen Kapitel. Im Wesentlichen handelt es sich um Schreibweisen.

Kapitel 1 JAVA
In eclipse Java-Projekt KREIS und in diesem class CKreis1 anlegen...
Ebenso ist bei anderen *.java Dateien vorzugehen.
Nach // stehen erläuternde Kommentare.

```
//KREIS/CKreis1.java (Klasse mit integrierter main-Funktion)
public class CKreis1 {//mit Zugriffsspezifizierer public
//Attribut(e)
double aradius;//Attribut mit Variablendeklaration double
//ohne eigenen Konstruktor!
//Methoden
double mumfang(){//Methode1
        double umf;//Hilfsvariable, existiert extern nicht!
        umf=2*Math.PI*aradius;
        return umf;
}
double minhalt(){//Methode2
        double inh;//Hilfsvariable
        inh=Math.PI*aradius*aradius;
        return inh;
}
//Test mit internem Aufruf, entfaellt bei externem Aufruf *
public static void main(String[] args) {
                CKreis1 OKreis1=new CKreis1();//Anlage Objekt von CKreis1
OKreis1.aradius=10.0;//Uebergabe Attribut
System.out.println("Umfang="+OKreis1.mumfang());//Aufruf Methoden
System.out.println("Inhalt="+OKreis1.minhalt());
}
//interner Aufruf, entfaellt bei externem Aufruf *
}
//Merke:
//Klassen sind komplexe Variablentypen, Objekte einer Klasse
//sind Variablen vom Typ der Klasse:
//CKlassenName Okn=new CKLassenName();
//Benutzung:
//Okn.a*=wert ist Attributfestlegung
//Okn.m*() ist Funktionsaufruf
//aus: Umfang=62.8 Inhalt=314.159
```

```
//KREIS/CKreis2.java (ohne interne Klasse, benutzt externe Kl.)
import java.util.*;
public class CKreis2 {
	public static void main(String[] args) {
		CKreis1 OKreis=new CKreis1();//Zugriff, da gleiches package
		Scanner sc=new Scanner(System.in);//sc ist Objekt der Klasse Scanner
		System.out.print("Radius=");
		OKreis.aradius=sc.nextDouble();//Dezimalkomma!?
		System.out.println("u="+OKreis.mumfang());
		System.out.println("A="+OKreis.minhalt());
}
}
//ein: 10,0 aus: u=62.8 A=315.9
//man beachte den Zugriff auf CKreis1 (Wiederverwendbarkeit)
//durch Trennung von reinen Klassen und main-Klassen
//wird das Programmieren uebersichtlicher und damit
//sicherer und besser wiederverwendbar.
//Natuerlich muss man in der main-Klasse ein Objekt der
//reinen Klasse bilden, um sie zu verwenden.
//Zu Testzwecken empfiehlt es sich der reinen Klasse
//eine main-Funktion beizugeben, was java erlaubt.

//KREIS/CKreis3.java (ohne interne Klasse, benutzt externe Kl.)
import java.util.*;//Enthaelt Klasse Scanner
public class CKreis3 {
	public static void main(String[] args) {
		Scanner sc=new Scanner(System.in);
		CKreis1 OKreis=new CKreis1();//Zugriff, da gleiches package
		try{
		System.out.print("Radius mit Dezimalkomma");
		OKreis.aradius=sc.nextDouble();
		System.out.println("u="+OKreis.mumfang());
		System.out.println("A="+OKreis.minhalt());
		}
		catch(Exception e){
			System.out.println("Fehler");
		}
	}
}
//ein: 10,0 aus: u=62.8 A=315.9
//Fehlerkontrolliert durch try/catch
```

```
//KREIS/CKreis4.java (ohne interne Klasse, benutzt externe Kl.)
import java.util.*;//Enthaelt Klasse Scanner
public class CKreis4 {
	public static void main(String[] args) {
		Double eingabe;
		boolean my=true;//logische Pruefvariable
		Scanner sc=new Scanner(System.in);//vor while platzieren!
		CKreis1 OKreis=new CKreis1();//Zugriff, da gleiches package
		while(my==true){//Schleifenanfang
		System.out.print("Radius mit Dezimalkomma/x=Ende");
		try{
		eingabe=sc.nextDouble();
		OKreis.aradius=eingabe;
		System.out.println("u="+OKreis.mumfang());
		System.out.println("A="+OKreis.minhalt());
}
		catch(Exception e){
		System.out.println("Fehler/Ende");
		my=false;
		}
		}//Schleifenende
	}
}
//ein: 10,0 aus: u=62.8 A=315.9
```

```
//RECHTECK/CRechteck1.java
public class CRechteck1 {
double alaenge;//Attribut, extern festlegbar und benutzbar
double abreite;
//ohne eigenen Konstruktor!
double mumfang(){//Methode1
        double umf;//Hilfsvariable, existiert extern nicht!
        umf=2*(alaenge+abreite);
        return umf;
}
double minhalt(){//Methode2
        double inh;//Hilfsvariable
        inh=alaenge*abreite;
        return inh;
}
//interner Aufruf, entfaellt bei externem Aufruf
public static void main(String[] args) {
                CRechteck1 ORechteck1=new CRechteck1();//Anlage Objekt von CKreis1
ORechteck1.alaenge=10.0;//Uebergabe Attribut
ORechteck1.abreite=20;
System.out.println("Umfang="+ORechteck1.mumfang());//Aufruf Methoden
System.out.println("Inhalt="+ORechteck1.minhalt());
}
//interner Aufruf
}
//Merke:
//ObjektName.a*=wert ist Attributfestlegung
//ObjektName.m*() ist Funktionsaufruf
//aus: Umfang=62.8 Inhalt=314.159
```

Ein einfacher Fall von Vererbung.

```
//RECHTECK/CQuader1.java
public class CQuader1 extends CRechteck1 {
double ahoehe;//Attribut, extern festlegbar und benutzbar
//ohne eigenen Konstruktor!
double minhalt(){//Ueberschreibt Methode der Superklasse
        double inh;//Hilfsvariable
        inh=alaenge*abreite*ahoehe;
        return inh;
}
double mmantel(){//Methode2
        double mantel;//Hilfsvariable
        double l=super.alaenge;//Zugriff auf Attribut Superkl
        double b=super.abreite;
        double h=ahoehe;
        mantel=2*(l*b+l*h+b*h);
        return mantel;
}
```

```
//interner Aufruf, entfaellt bei externem Aufruf
public static void main(String[] args) {
                CQuader1 OQuader1=new CQuader1();//Anlage Objekt von CKreis1
                OQuader1.alaenge=10.0;//Uebergabe Attribut
OQuader1.abreite=20;
OQuader1.ahoehe=30;
System.out.println("Inhalt="+OQuader1.minhalt());
System.out.println("Grundflaeche="+OQuader1.mmantel());
}
//interner Aufruf
}
//Merke:
//ObjektName.a*=wert ist Attributfestlegung
//ObjektName.m*() ist Funktionsaufruf
//aus: Umfang=62.8 Inhalt=314.159
```

Es folgen Beispiele zu Datenfeldern (Variablenlisten).

```
//FELD/FELD0.java
public class FELD0 {
int azufall1=(int)(Math.random()*49+1);
int azufall2=(int)(Math.random()*49+1);;
int azufall3=(int)(Math.random()*49+1);;
int azufall4=(int)(Math.random()*49+1);;
int azufall5=(int)(Math.random()*49+1);;
int azufall6=(int)(Math.random()*49+1);;
public void mprint(){
        System.out.print(azufall1+"-");
        System.out.print(azufall2+"-");
        System.out.print(azufall3+"-");
        System.out.print(azufall4+"-");
        System.out.print(azufall5+"-");
        System.out.print(azufall6+"-");
}
        public static void main(String[] args) {
FELD0 feld0=new FELD0();
feld0.mprint();
        }
}
//praktisch nicht sortierbar, Doppelte moeglich
```

```
//FELD/FELD1.java
public class FELD1 {
	int[] azufall=new int[6];//->azufall[0],...,azufall[5]
	void mprint(){
		for(int i=0;i<6;i++){//for-Anfang
		azufall[i]=(int)(Math.random()*49+1);//(int):ganz abrunden
		System.out.println(azufall[i]);
		}//for-Ende
	}
	public static void main(String[] args) {
	FELD1 feld1=new FELD1();
	feld1.mprint();
	}
}
//nicht sortiert, Doppelte moeglich

//FELD/FELD2.java
import java.util.Arrays;
public class FELD2 {
	int[] azufall=new int[6];//->azufall[0],...,azufall[5]
	FELD2(){//Konstruktor, muss wie Klasse heissen
		boolean my=false;//Algoritmus fuer Verschiedene
		while(my==false){
			my=true;
		for(int i=0;i<6;i++){//for-Anfang
			azufall[i]=(int)(Math.random()*49+1);//(int):ganz abrunden
			}//for-Ende
	Arrays.sort(azufall);//sortiert Liste

	if(azufall[0]==azufall[1])my=false;
	if(azufall[1]==azufall[2])my=false;
	if(azufall[2]==azufall[3])my=false;
	if(azufall[3]==azufall[4])my=false;
	if(azufall[4]==azufall[5])my=false;
	}
	}
	void mprint(){
		for(int i=0;i<6;i++){//for-Anfang
		System.out.println(azufall[i]);
		}//for-Ende
	}
	public static void main(String[] args) {
	FELD2 feld1=new FELD2();
	feld1.mprint();
	}
}
//sortiert, keine Doppelte moeglich
//ein Konstruktor legt die Attribute fest
//if-Anweisungen koennen durch for-Anweisung reduziert werden
```

```
//HAUPTSTADT/CHS0
import java.util.*;
public class CHS0 {
String[] namen=new String[32];
CHS0(){//Konstruktor, Name wie Klasse
	this.namen[0]="BW";
	this.namen[1]="S";
	this.namen[2]="BY";
	this.namen[3]="M";
	this.namen[4]="BE";
	this.namen[5]="B";
	this.namen[6]="BB";
	this.namen[7]="P";
	this.namen[8]="HB";
	this.namen[9]="HB";
	this.namen[10]="HH";
	this.namen[11]="HH";
	this.namen[12]="HE";
	this.namen[13]="WI";
	this.namen[14]="MV";
	this.namen[15]="SN";
	this.namen[16]="NI";
	this.namen[17]="H";
	this.namen[18]="NW";
	this.namen[19]="D";
	this.namen[20]="RP";
	this.namen[21]="MZ";
	this.namen[22]="SL";
	this.namen[23]="SB";
	this.namen[24]="SN";
	this.namen[25]="DD";
	this.namen[26]="ST";
	this.namen[27]="MD";
	this.namen[28]="SH";
	this.namen[29]="KI";
	this.namen[30]="TH";
	this.namen[31]="EF";
	}
```

```
public static void main(String[] args){
	int z;
	int r=0;
	String eingabe;
	Scanner sc=new Scanner(System.in);
	CHS0 hsq=new CHS0();
	for(int i=1;i<=10;i++){
	z=(int)(Math.random()*32);
	System.out.println(hsq.namen[z]);
	eingabe=sc.next();
	if(z%2==0){
		System.out.println("richtig: "+hsq.namen[z+1]);
		if(eingabe.equals(hsq.namen[z+1]))
			r=r+1;
	}
	if(z%2==1){
		System.out.println("richtig: "+hsq.namen[z-1]);
		if(eingabe.equals(hsq.namen[z-1]))
	r=r+1;
	}
	}
	System.out.println(r+" richtige bei 10 Aufgaben");
}
}
```

Es folgen Beispiele für Dateien, bei denen Variable auf die Festplatte gespeichert werden.

```
//MEMO/MEMO0.java
import java.util.*;
import java.io.*;
public class MEMO0 {
String awann;
String awas;
String adnm="memo0";
FileWriter afw;
void mspeichern(){
	try{
	afw=new FileWriter(adnm,true);//ohne true:neu
	afw.write(awann+"\n"+awas+"\n");
	afw.close();
	}
	catch(Exception e){
	System.out.println("Fehler!");
	}
}
void mlesen(){
	FileReader fr;
		String s;
	try{
	fr=new FileReader(adnm);
	BufferedReader br=new BufferedReader(fr);
	while((s=br.readLine())!=null)
			System.out.println(s);
	}
	catch(Exception e){
		System.out.println("Fehler");
	}
}
	public static void main(String[] args) {
	MEMO0 memo0=new MEMO0();
	Scanner sc=new Scanner(System.in);
	System.out.print("wann? (empfohlen: jjmmtt)");
memo0.awann=sc.nextLine();
System.out.print("was?");
memo0.awas=sc.nextLine();
sc.close();
memo0.mspeichern();
memo0.mlesen();
	}
}
//s. Ordner MEMO
```

```
//MEMO/MEMO1.java
import java.util.*;
import java.io.*;
public class MEMO1 {
String awann;
String awas;
String adnm="memo1.txt";
FileWriter afw;
void mspeichern(){
	if(!awann.equals("#") && !awas.equals("#")){
		try{
	afw=new FileWriter(adnm,true);//ohne true:neu
	afw.write(awann+"\n"+awas+"\n");
	afw.close();
	}
	catch(Exception e){
	System.out.println("Fehler!");
	}
	}
	}
void mlesen(){
	FileReader fr;
		String s;
		Integer counter=0;
	try{
	fr=new FileReader(adnm);
	BufferedReader br=new BufferedReader(fr);
	while((s=br.readLine())!=null){
		counter++;
			System.out.println(counter+":"+s);
	}
	}
	catch(Exception e){
		System.out.println("Fehler");
	}
}
	public static void main(String[] args) {
	MEMO1 memo1=new MEMO1();
	Scanner sc=new Scanner(System.in);
	System.out.print("wann? (jjmmtt)");
memo1.awann=sc.nextLine();
System.out.print("was?)");
memo1.awas=sc.nextLine();
sc.close();
memo1.mspeichern();
memo1.mlesen();
	}
}
//s. Ordner MEMO
```

```
//MEMO/MEMO2.java
import java.util.*;
import java.io.*;
public class MEMO2 {
String awann;
String awas;
String adnm="memo2.txt";
FileWriter afw;
void mspeichern(){
	if(!awann.equals("#") && !awas.equals("#")){
		try{
	afw=new FileWriter(adnm,true);//ohne true:neu
	afw.write(awann+">"+awas+"\n");
	afw.close();
	}
	catch(Exception e){
	System.out.println("Fehler!");
	}
	}
	}
void mlesen(){
	FileReader fr;
		String s;
		try{
	fr=new FileReader(adnm);
	BufferedReader br=new BufferedReader(fr);
	while((s=br.readLine())!=null){
		if((awann.equals("#") || awas.equals("#")) && (s.contains(awann) ||
s.contains(awas)))
		System.out.println(s);
	}
	}
	catch(Exception e){
		System.out.println("Fehler");
	}
}
	public static void main(String[] args) {
	MEMO2 memo2=new MEMO2();
	Scanner sc=new Scanner(System.in);
	System.out.print("wann? (jjmmtt, #=Suche)");
memo2.awann=sc.nextLine();
System.out.print("was? (Woerter, #=Suche)");
memo2.awas=sc.nextLine();
sc.close();
memo2.mspeichern();
memo2.mlesen();
	}
}
//s. Ordner MEMO
```

Kapitel 2 PHP

Es folgen jetzt zu obigen java-Beispielen analoge php-Beispiele.

Zur Ausführung der Dateien wird ein webserver (hier xampp) und ein browser (hier firefox) benützt. Der Dateiname steht jeweils im Formular und muss dort auch stehen.

Datei CKreis0.php in ...htdocs/KREIS gespeichert,

start: ff mit localhost:8080/KREIS/ CKreis0.php

```
<html>
<head>
<!- Paragraph stylen -->
<style>
p {
   border: 2px solid;
   width: 400px;
   height: 200px;
   resize: both;
   overflow: auto;
}
</style>
<!- Paragraph stylen -->
</head>
<body>
<h3>Bitte ausf&uuml;llen!</h3>
<form action="CKreis0.php" method="post">
<input id="vradius" name="vradius" required>Radius<br>
<input type="submit" name="sbmt" value="eingaben"><br>
</form>
<p>
<?php
class CKreis0{//Klasse
public $aradius;//Attribute
//function __construct($pradius){//Konstruktor, iniziiert die Klassenvariablen
//$this->aradius=$pradius;
//}
function mumfang(){//Methode
$ret=6.28*$this->aradius;
return $ret;
}
}
//main
if(isset($_POST['vradius']) && $_POST['vradius']>0){
$vradius=$_POST['vradius'];
//$OKreis0 = new CKreis0($vradius);//Objekt vom Typ CGruss wird angelegt
$OKreis0 = new CKreis0();
$OKreis0->aradius=$vradius;
$ausgabe=$OKreis0->mumfang();//Übergabe von Methode an Variable
echo "Umfang=".$ausgabe;
}
?>
</p>
</body>
</html>
```

Die Datei CKreis.php repräsentiert die Klasse und wird in Kreis.php per include eingebunden.

```
<?php
class CKreis{//Klasse
public $aradius;//Attribute
//function __construct($pradius){//Konstruktor, iniziiert die Klassenvariablen
//$this->aradius=$pradius;
//}
function mumfang(){//Methode
$ret=6.28*$this->aradius;
return $ret;
}
function minhalt(){//Methode
$ret=3.14*$this->aradius*$this->aradius;
return $ret;
}
}
?>
```

Datei Kreis.php

```
<html>
<head>
<!- Paragraph stylen -->
<style>
p {
   border: 2px solid;
   width: 400px;
   height: 200px;
   resize: both;
   overflow: auto;
}
</style>
<!- Paragraph stylen -->
</head>
<body>
<h3>Bitte ausf&uuml;llen!</h3>
<form action="Kreis.php" method="post">
<input id="vradius" name="vradius" required>Radius<br>
<input type="submit" name="sbmt" value="eingaben"><br>
</form>
<p>
<?php
include 'CKreis.php';//Einbinden einer externen Klasse
//main
if(isset($_POST['vradius']) && $_POST['vradius']>0){
$vradius=$_POST['vradius'];
//$OKreis = new CKreis($vradius);//Objekt vom Typ CGruss wird angelegt
$OKreis = new CKreis();
$OKreis->aradius=$vradius;
$ausgabe=$OKreis->mumfang();//Übergabe von Methode an Variable
echo "Umfang=".$ausgabe;
echo "<br>";
$ausgabe=$OKreis->minhalt();//Übergabe von Methode an Variable
echo "Inhalt=".$ausgabe;

}
?>
</p>
</body>
</html>
```

Es folgt jetzt ein sogenannter Vererbungsfall, bei dem eine bestehende Klasse (Superklasse) durch eine Subklasse erweitert wird. Dabei kann die Subklasse auf die Superklasse zugreifen.

```
<html>
<head>
<!- Paragraph stylen -->
<style>
p {
   border: 2px solid;
   width: 400px;
   height: 200px;
   resize: both;
   overflow: auto;
}
</style>
<!- Paragraph stylen -->
</head>
<body>
<h3>Bitte ausf&uuml;llen!</h3>
<form action="Quader.php" method="post">
<input id="vlaenge" name="vlaenge" required>Laenge<br>
<input id="vbreite" name="vbreite" required>Breite<br>
<input id="vhoehe" name="vhoehe" required>hoehe<br>
<input type="submit" name="sbmt" value="eingaben"><br>
</form>
<p>
<?php
class CRechteck{//Klasse
public $alaenge;//Attribute
public $abreite;
function mumfang(){//Methode
$ret=2*($this->alaenge+$this->abreite);
return $ret;
}
function minhalt(){//Methode
$ret=$this->alaenge*$this->abreite;
return $ret;
}
}
class CQuader extends CRechteck{
public $ahoehe;
function minhalt(){//Methode
$ret=$this->alaenge*$this->abreite*$this->ahoehe;
return $ret;
}
function mmantel(){//Methode
$ret=parent::minhalt();
return $ret;
}

}
```

```
//main
if(isset($_POST['vlaenge']) && $_POST['vlaenge']>0){
$vlaenge=$_POST['vlaenge'];
$vbreite=$_POST['vbreite'];
$vhoehe=$_POST['vhoehe'];
$OQuader = new CQuader();
$OQuader->alaenge=$vlaenge;
$OQuader->abreite=$vbreite;
$OQuader->ahoehe=$vhoehe;
echo "V=".$OQuader->minhalt()."<br>";
echo "G=".$OQuader->mmantel();
}
?>
</p>
</body>
</html>
```

Es folgen Beispiele zu Datenfeldern (Variablenlisten).

```
<html>
<head>
<!- Paragraph stylen -->
<style>
p {
   border: 2px solid;
   width: 400px;
   height: 200px;
   resize: both;
   overflow: auto;
}
</style>
<!- Paragraph stylen -->
</head>
<body>
<h3>Bitte ausf&uuml;llen!</h3>
<form action="CLotto0.php" method="post">
<input id="vlotto" name="vlotto" required>Lotto(j/n)<br>
<input type="submit" name="sbmt" value="eingaben"><br>
</form>
<p>
<?php
class CLotto0{//Klasse
public $az1;//Attribute
public $az2;//Attribute
public $az3;//Attribute
public $az4;//Attribute
public $az5;//Attribute
public $az6;//Attribute
function mlotto(){//Methode
srand((double)microtime()*1000000);
$this->az1=rand(1,49);
$this->az2=rand(1,49);
$this->az3=rand(1,49);
$this->az4=rand(1,49);
$this->az5=rand(1,49);
$this->az6=rand(1,49);
$ret=$this->az1." ".$this->az2." ".$this->az3." ".$this->az4." ".$this->az5." ".$this->az6." ";
return $ret;
}
}
//main
if(isset($_POST['vlotto']) && $_POST['vlotto']!=""){
$OLotto0 = new CLotto0();
$ausgabe=$OLotto0->mlotto();//Übergabe von Methode an Variable
echo "Ziehung: ".$ausgabe;
}
?>
</p>
</body>
</html>
```

```
<html>
<head>
<!- Paragraph stylen -->
<style>
p {
   border: 2px solid;
   width: 400px;
   height: 200px;
   resize: both;
   overflow: auto;
}
</style>
<!- Paragraph stylen -->
</head>
<body>
<h3>Bitte ausf&uuml;llen!</h3>
<form action="CLotto1.php" method="post">
<input id="vlotto" name="vlotto" required>Lotto(j/n)<br>
<input type="submit" name="sbmt" value="eingaben"><br>
</form>
<p>
<?php
class CLotto1{//Klasse
public $az=array();//Attribute
function mlotto(){//Methode
srand((double)microtime()*1000000);
for($i=0;$i<=5;$i++)
$this->az[]=rand(1,49);
$ret=$this->az;
return $ret;
}
}
//main
if(isset($_POST['vlotto']) && $_POST['vlotto']!=""){
$OLotto1 = new CLotto1();
$ausgabe=$OLotto1->mlotto();
echo "Ziehung: ";
echo implode(" ",$ausgabe);
}
?>
</p>
</body>
</html>
```

```
<html>
<head>
<!- Paragraph stylen -->
<style>
p {
   border: 2px solid;
   width: 400px;
   height: 200px;
   resize: both;
   overflow: auto;
}
</style>
<!- Paragraph stylen -->
</head>
<body>
<h3>Bitte ausf&uuml;llen!</h3>
<form action="CLotto2.php" method="post">
<input id="vlotto" name="vlotto" required>Lotto(j/n)<br>
<input type="submit" name="sbmt" value="eingaben"><br>
</form>
<p>
<?php
class CLotto2{//Klasse
public $az=array();//Attribute
function mlotto(){//Methode
srand((double)microtime()*1000000);
for($i=0;$i<=5;$i++)
$this->az[]=rand(1,49);
asort($this->az);//Werte aufst., arsort Werte abst.
$ret=$this->az;
return $ret;
}
}
//main
if(isset($_POST['vlotto']) && $_POST['vlotto']!=""){
$OLotto2 = new CLotto2();
$ausgabe=$OLotto2->mlotto();
echo "Ziehung: ";
echo implode(" ",$ausgabe);
}
?>
</p>
</body>
</html>
```

```
<html>
<head>
<!- Paragraph stylen -->
<style>
p {
   border: 2px solid;
   width: 400px;
   height: 200px;
   resize: both;
   overflow: auto;
}
</style>
<!- Paragraph stylen -->
</head>
<body>
<h3>Bitte ausf&uuml;llen!</h3>
<form action="CLotto3.php" method="post">
<input id="vlotto" name="vlotto" required>Lotto(j/n)<br>
<input type="submit" name="sbmt" value="eingaben"><br>
</form>
<p>
<?php
class CLotto3{//class
public $az=array(0,0,0,0,0,0);//Attribut
function mlotto(){//Methode
srand((double)microtime()*1000000);
$d=1;
while($d>0){//while
$d=0;//keine Doppelten
for($i=0;$i<6;$i++)$this->az[$i]=rand(1,49);
$ret=$this->az;
sort($ret);
if($ret[0]==$ret[1])$d=1;//doch Doppelte
if($ret[1]==$ret[2])$d=1;
if($ret[2]==$ret[3])$d=1;
if($ret[3]==$ret[4])$d=1;
if($ret[4]==$ret[5])$d=1;
}//while
return $ret;
}//Methode
}//class
//main
if(isset($_POST['vlotto']) && $_POST['vlotto']!=""){
$OLotto3 = new CLotto3();
$ausgabe=$OLotto3->mlotto();
echo "Ziehung: ";
echo implode(" ",$ausgabe);
}
?>
</p>
</body>
</html>
```

```
<html>
<head>
<!- Paragraph stylen -->
<style>
p {
   border: 2px solid;
   width: 400px;
   height: 200px;
   resize: both;
   overflow: auto;
}
</style>
<!- Paragraph stylen -->
</head>
<body>
<h3>Bitte ausf&uuml;llen!</h3>
<form action="CLotto4.php" method="post">
<input id="vlotto" name="vlotto" required>Lotto(j/n)<br>
<input type="submit" name="sbmt" value="eingaben"><br>
</form>
<p>
<?php
class CLotto4{//class
public $az=array(0,0,0,0,0,0);//Attribut
function mlotto(){//Methode
srand((double)microtime()*1000000);
$d=1;
while($d>0){//while
$d=0;//keine Doppelten
for($i=0;$i<6;$i++)$this->az[$i]=rand(1,49);
$ret=$this->az;
sort($ret);
for($i=0;$i<5;$i++){//for
if($ret[$i]==$ret[$i+1])
$d=1;//doch Doppelte
if($d==1)break;
}//for
}//while
return $ret;
}//Methode
}//class
//main
if(isset($_POST['vlotto']) && $_POST['vlotto']!=""){
$OLotto4 = new CLotto4();
$ausgabe=$OLotto4->mlotto();
echo "Ziehung: ";
echo implode(" ",$ausgabe);
}
?>
</p>
</body>
</html>
```

Etwas einfacher als bei java gestaltet sich die Dateibehandlung.

```
<html>
<head>
<!- Paragraph stylen -->
<style>
p {
   border: 2px solid;
   width: 400px;
   height: 200px;
   resize: both;
   overflow: auto;
}
</style>
<!- Paragraph stylen -->
</head>
<body>
<h3>Bitte ausf&uuml;llen!</h3>
<form action="CMemo0.php" method="post">
<input id="vwann" name="vwann" required>wann(jjmmtt)<br>
<input id="vwas" name="vwas" required>was<br>
<input type="submit" name="sbmt" value="eingaben"><br>
</form>
<p>
<?php
class CMemo0{//Klasse
public $awann;//Attribute
public $awas;
function mspeichern(){//Methode
$fp=fopen("memo0.csv","a");
fputs($fp,$this->awann.">");
fputs($fp,$this->awas."\n");
fclose($fp);
readfile("memo0.csv");//von excel/localc lesbar!
}
}
//main
if(isset($_POST['vwann']) && isset($_POST['vwann'])){
$vwann=$_POST['vwann'];
$vwas=$_POST['vwas'];
$OMemo0 = new CMemo0();
$OMemo0->awann=$vwann;
$OMemo0->awas=$vwas;
$OMemo0->mspeichern();//Übergabe von Methode an Variable
}
?>
</p>
</body>
</html>
```

```
<html>
<head>
<!- Paragraph stylen -->
<style>
p {
   border: 2px solid;
   width: 400px;
   height: 200px;
   resize: both;
   overflow: auto;
}
</style>
<!- Paragraph stylen -->
</head>
<body>
<h3>Bitte ausf&uuml;llen!</h3>
<form action="CMemo1.php" method="post">
<input id="vwann" name="vwann" required>wann(jjmmtt)<br>
<input id="vwas" name="vwas" required>was<br>
<input type="submit" name="sbmt" value="eingaben"><br>
</form>
<p>
<?php
class CMemo1{//Klasse
public $awann;//Attribute
public $awas;
function mspeichern(){//Methode
$fp=fopen("memo1.txt","a");
fputs($fp,$this->awann.">");
fputs($fp,$this->awas.chr(10));
fclose($fp);
$daten=file("memo1.txt");
for($i=0;$i<sizeof($daten);$i++)
echo $daten[$i]."<br>";
}
}
//main
if(isset($_POST['vwann']) && isset($_POST['vwann'])){
$vwann=$_POST['vwann'];
$vwas=$_POST['vwas'];
$OMemo1 = new CMemo1();
$OMemo1->awann=$vwann;
$OMemo1->awas=$vwas;
$OMemo1->mspeichern();//Übergabe von Methode an Variable
}
?>
</p>
</body>
</html>
```

```
<html>
<head>
<!- Paragraph stylen -->
<style>
p {
   border: 2px solid;
   width: 400px;
   height: 200px;
   resize: both;
   overflow: auto;
}
</style>
<!- Paragraph stylen -->
</head>
<body>
<h3>Bitte ausf&uuml;llen!</h3>
(in ein Feld # in anderes Suchteil moeglich)<br>
<form action="CMemo2.php" method="post">
<input id="vwann" name="vwann" required>wann(jjmmtt)<br>
<input id="vwas" name="vwas" required>was<br>
<input type="submit" name="sbmt" value="eingaben"><br>
</form>
<p>
<?php
class CMemo2{//Klasse
public $awann;//Attribute
public $awas;
function mspeichern(){//Methode
$fp=fopen("memo2.txt","a");
if($this->awann!="#" && $this->awas!="#"){
fputs($fp,$this->awann.">".$this->awas.chr(10));
}
fclose($fp);
$daten=file("memo2.txt");
if($this->awann=="#"){
$h=$this->awas;
for($i=0;$i<sizeof($daten);$i++){
if(strpos($daten[$i],$h))
echo $daten[$i]."<br>";
}
}
if($this->awas=="#"){
$h=$this->awann;
for($i=0;$i<sizeof($daten);$i++){
if(strpos($daten[$i],$h)!==false)
echo $daten[$i]."<br>";
}
}

}
}
```

```
//main
if(isset($_POST['vwann']) && isset($_POST['vwann'])){
$vwann=$_POST['vwann'];
$vwas=$_POST['vwas'];
$OMemo2 = new CMemo2();
$OMemo2->awann=$vwann;
$OMemo2->awas=$vwas;
$OMemo2->mspeichern();//Übergabe von Methode an Variable
}
?>
</p>
</body>
</html>
```

Kapitel 3 JAVASCRIPT

Der Javascript-Teil befindet sich in einer *.html Datei und zwar ab onclick. Damit hat man die größte Ähnlichkeit mit php. Im Javascript-Teil darf man nur Hochkommata (‘) verwenden.

```
<!DOCTYPE html>
<html>
<head>
<title>Test mit Parametern und Formular</title>
<style>
p {
   border: 2px solid;
   width: 400px;
   height: 200px;
   resize: both;
   overflow: auto;
}
</style>
</head>
<body>
<h3> Bitte ausf&uuml;llen </h3>
<table>
<form method="" action="">
<input id="laenge" name="laenge" required>Laenge<br>
<input id="breite" name="breite" required>Breite<br>
<input type="button" value="click!" onclick=
"
function CRechteck(){//Klasse
this.alaenge;//Attribute
this.abreite;
this.mflaeche=function(){//Methoden
return this.alaenge*this.abreite;
}
this.mumfang=function(){
return 2.0*this.alaenge+2.0*this.abreite;
}
}
//main
ORechteck=new CRechteck();//Objekt
ORechteck.alaenge=laenge.value;//Zuweisungen
ORechteck.abreite=breite.value;
ausgabe1=ORechteck.mflaeche();//die Variable erhält den Rückgabewert
ausgabe2=ORechteck.mumfang();//die Variable erhält den Rückgabewert
pausgabe.innerHTML='Flaeche='+ausgabe1+' Umfang='+ausgabe2;
"
>
</form>
<p id="pausgabe">
</p>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
<title>Test mit Parametern und Formular</title>
<style>
p {
   border: 2px solid;
   width: 400px;
   height: 200px;
   resize: both;
   overflow: auto;
}
</style>
</head>
<body>
<h3> Bitte ausf&uuml;llen </h3>
<table>
<form method="" action="">
<input id="laenge" name="laenge" required>Laenge<br>
<input id="breite" name="breite" required>Breite<br>
<input id="hoehe" name="hoehe" required>Hoehe<br>
<input type="button" value="click!" onclick=
"
function CRechteck(){
this.alaenge;
this.abreite;
this.mflaeche=function(){
return this.alaenge*this.abreite;
}
}
function CQuader(){
this.ahoehe;
this.mvolumen=function(){
return this.alaenge*this.abreite*this.ahoehe;
}
}
//main
CQuader.prototype=new CRechteck;//CQuader extends CRechteck
var OQuader=new CQuader;
OQuader.alaenge=laenge.value;
OQuader.abreite=breite.value;
OQuader.ahoehe=hoehe.value;
ausgabe1=OQuader.mvolumen();//die Variable erhält den Rückgabewert
ausgabe2=OQuader.mflaeche();
pausgabe.innerHTML='Volumen='+ausgabe1+' Grundflaeche='+ausgabe2;
"
>
</form>
<p id="pausgabe">
</p>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
<title>Test mit Parametern und Formular</title>
<style>
p {
    border: 2px solid;
    width: 400px;
    height: 200px;
    resize: both;
    overflow: auto;
}
</style>
</head>
<body>
<h3> Bitte ausf&uuml;llen </h3>
<table>
<form method="" action="">
<input type="button" value="Lotto!" onclick=
"
function CLotto(){//Klasse
this.alotto=[];//Attribute
this.mlotto=function(){//Methoden
my=1;
while(my==1){
my=0;
 for (i = 0; i < 6; i++)
   this.alotto[i]=Math.floor((Math.random() * 49) + 1);
this.alotto.sort(function(a, b){return a-b});//Zahlen sortieren(strings nur sort())
for (i=0;i<5;i++){
if(this.alotto[i]==this.alotto[i+1])
my=1;
}
}
}
}
//main
OLotto=new CLotto();//Objekt
OLotto.mlotto();
ausgabe='Lotto=';
for (i = 0; i < 6; i++) {
   ausgabe=ausgabe+OLotto.alotto[i]+'-';
}
pausgabe.innerHTML=ausgabe;
"
>
</form>
<p id="pausgabe">
</p>
</body>
</html>
```