Lösungen zu Übungsaufgaben

Quelle: Maurice (2015); Theis (2016)

Kapitel: "8. Objektorientierung"

Aufgabe 8.1:
Definieren Sie eine Klasse mit dem Namen Fahrzeug, die die folgenden Eigenschaften hat:

- Farbe
- Hersteller

Und außerdem hat sie zwei Methoden, die jeweils eine Meldung ausgeben:

- Starten
- Stoppen

Erstellen Sie dann ein Objekt zu der Klasse!

Lösung:

```
class Fahrzeug
{
    public $hersteller;
    public $farbe;
    public function __construct($farbe, $hersteller)
    {
        $this->farbe=$farbe;
        $this->hersteller = $hersteller;
    }
    public function starten()
    {
        echo "gestartet<br />\n";
    }
    public function stoppen()
    {
        echo "gestoppt<br />\n";
    }
}
$f = new Fahrzeug("rot", "Mobile");
$f->starten();
$f->stoppen();
```

Aufgabe 8.2:
Definieren Sie eine Klasse *Math* mit mindestens zwei statischen Methoden:

- Die Methode wurzel() soll die Quadratwurzel liefern. Hierfür können Sie *sqrt()* benutzen (http://php.net/manual/de/function.sqrt.php).
- Die Methode absolut() soll den absoluten Wert einer Zahl berechnen, was Sie über die PHP-Funktion *abs()* bewerkstelligen können.
- Sie können gerne noch weitere Methoden integrieren.

Rufen Sie die Methoden dann auf.

Lösung:

```
class Math
{
    public static function wurzel($zahl)
    {
        return sqrt($zahl);
    }

    public static function absolut($zahl)
    {
        return abs($zahl);
    }
}
echo Math::wurzel(9);
echo "<br />\n";
echo Math::absolut(-54);
```

Aufgabe 8.3:
In einer Übung (Aufgabe 8.1) haben Sie eine Klasse Fahrzeug definiert. Erstellen Sie jetzt dazu eine abgeleitete Klasse namens Auto. Diese Klasse hat folgende Unterschiede im Vergleich zur Basisklasse:

- Es gibt eine Eigenschaft zur Speicherung des Kilometerstands.
- Es gibt eine Methode zum Fahren, bei der der Kilometerstand um die gefahrenen Kilometer erhöht wird.

Lösung:

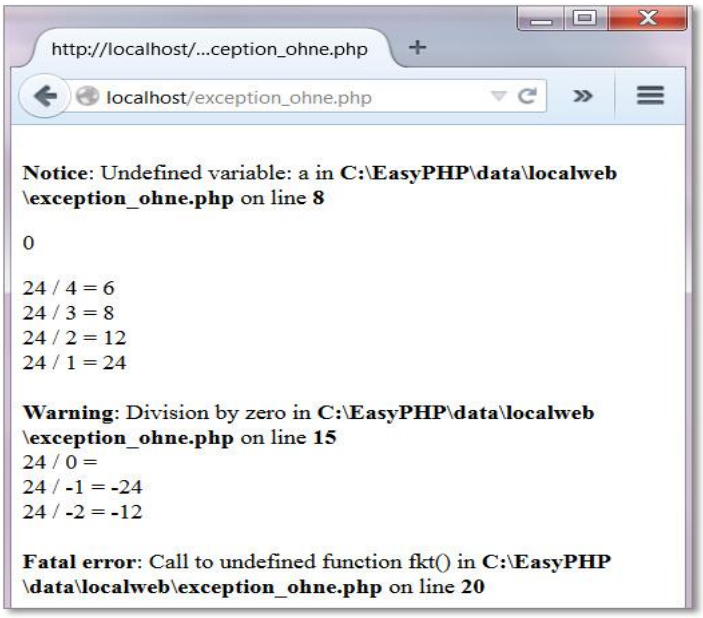
```
class Auto extends Fahrzeug
{
    public $kilometerstand;
    public function __construct($farbe, $hersteller,
                                        $kilometerstand)
    {
        $this->farbe=$farbe;
        $this->hersteller = $hersteller;
        $this->kilometerstand = $kilometerstand;
    }

    public function fahren($kilometer)
    {
        $this->kilometerstand = $this->kilometerstand +
                                              $kilometer;
        echo "Aktueller Kilometerstand: " . $this->
                                kilometerstand . "<br />\n";
    }
}
```

```
$f = new Auto("grau", "XY", 22);
$f->starten();
$f->fahren(30);
$f->stoppen();
```

Aufgabe 8.4:
Behandeln Sie die Fehler im folgenden Programm. Ergänzen Sie es mit entsprechenden Codes für die Ausnahmebehandlung. Sie können hierzu die Funktionen *isset* und *function_exists* verwenden.

```
<html>
<body>
<?php
/* Für dieses Programm alle Fehler anzeigen */
ini_set("error_reporting", 32767);
/* Variable unbekannt */
$c = 2 * $a;
echo "<p>$c</p>";
/* Division durch 0 */
$x = 24;
for($y=4; $y>-3; $y--) {
  $z = $x / $y;
  echo "$x / $y = $z<br />";
}
/* Zugriff auf Funktion */
fkt();
echo "Ende";
?>
</body>
</html>
```



Lösung:

Mit Ausnahmebehandlung

```
// Datei exception_mit.php

<html>
<body>
<?php
/* Für dieses Programm alle Fehler anzeigen */
ini_set("error_reporting", 32767);

/* Variable unbekannt */
try {
    if(!isset($a))
        throw new Exception("Variable unbekannt");
    $c = 2 * $a;
    echo "<p>$c</p>";
}
catch(Exception $e) {
      echo $e->getMessage() . "<br />";
}
finally {
    echo "<p>Ende, Variable unbekannt</p>";
}

/* Division durch 0 */
$x = 24;
for($y=4; $y>-3; $y--) {
   try {
       if($y == 0)
            throw new Exception("Division durch 0");
```

```
        $z = $x / $y;
       echo "$x / $y = $z<br />";
    }
    catch(Exception $e) {
          echo $e->getMessage() . "<br />";
     }
}

/* Zugriff auf Funktion */
try {
    if(!function_exists("fkt"))
          throw new Exception("Funktion unbekannt");
          // der Rest des try-Blocks wird nicht mehr bearbeitet.
    fkt();
}
catch(Exception $e) {
    echo $e->getMessage() . "<br />";
}
finally {       // Die Anweisungen in einem finally- Block werden
                // in jedem Fall durchgeführt
     echo "<p>Ende, Funktion unbekannt</p>";
}
echo "Ende";
?>
</body>
</html>
```