

Programování v PHP

Katedra softwarového inženýrství
Fakulta informačních technologií
České vysoké učení technické v Praze

© Pavel Štěpán, Helena Wallenfelsová, 2014

Chyby a výjimky
BI-PHP



```

<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Osetrení chyb a výjimek</title>
    </head>
    <body>
        <?php

            // Chyby možno zapisovat do logu - nastavení v php.ini

            // Vetsina chyb se při standardním běhu PHP NEZOBRAZUJE!!
            // (co s nimi uživatel?). Pro ladení vhodné změnit:
            ini_set('display_errors', 1); // změni nastavení určene v php.ini

            ini_set('error_reporting', E_ALL);
            // oznamování vybraných chyb:
            // ini_set('error_reporting', E_ALL & ~E_NOTICE);

            $nejni++;
            @$taky_nejni++; // potlačení chybových hlásek

            // vlastní error handler
            function my_error_handler($severity, $msg, $filename, $linenum) {
                // zde možno také provádět vlastní zápisy chyb do souboru, db, ...
                switch($severity) {
                    case E_NOTICE:
                        echo "<br>Ejhle chyba typu notice: $msg v ".
                            "$filename:$linenum<br>";
                        break;
                    case E_WARNING:
                        echo "<br>Ejhle chyba typu warning: $msg v ".
                            "$filename:$linenum<br>";
                        break;
                    case E_ERROR:
                        echo "<br>Ejhle chyba typu error: $msg v ".
                            "$filename:$linenum<br>";
                        break;
                    default:
                        echo "<br>Neznámá chyba $filename:$linenum<br>";
                        break;
                }
            }

            set_error_handler("my_error_handler"); // přiřazení error handleru

            // a zase delám chybu - zpracuje nový error handler:
            $xxx++;

```

```

// pouziti vyjimiek (modernejsi)
class ZdrojVyjimiek{
    public function setJmeno($jmeno){
        if (empty($jmeno)) throw new Exception("Prazdne jmeno!",
                                                    1000);
    }
}

$objZdrojVyjimiek = new ZdrojVyjimiek();

try {
    echo "Pred chybou<br>";
    $objZdrojVyjimiek->setJmeno("");
    echo "Za chybou<br>";
}
catch (Exception $ex){
    echo "Vyjimka: ".$ex->getMessage()."<br>";
}

// definice vlastnich vyjimiek
class emptyException extends Exception{
    // ...
}

class noNumberException extends Exception{
    // ...
}

class RuzneVyjimky{
    public function setJmeno($jmeno){
        if (empty($jmeno))
            throw new emptyException("Prazdne jmeno!", 1000);
    }

    public function setID($ID){
        if (!is_numeric($ID))
            throw new notNumberException("Neciselne ID!", 1200);
    }
}

$objRuzneVyjimky = new RuzneVyjimky();

try {
    echo "Pred RuzneVyjimky<br>";
    $objRuzneVyjimky->setID("ABC");
    echo "Za RuzneVyjimky<br>";
}
catch (emptyException $ex){
    echo "Zachycena vyjimka typu emptyException: ".
        $ex->getMessage()."<br>";
}
catch (notNumberException $ex){
    echo "Zachycena vyjimka typu notNumberException: ".
        $ex->getMessage()."<br>";
}
catch (Exception $ex){
    echo "Zachycena neznama vyjimka: ".
        $ex->getMessage()."<br>";
}

```

```
// vlastní handler pro zachytávání neosetřených výjimek
function my_exception_handler(Exception $vyjimka){
    echo "Nezachycená výjimka typu ".get_class($vyjimka)."<br>";
    exit();
}

// přiřazení exception handleru
set_exception_handler("my_exception_handler");

echo "Pred neosetřenou výjimkou<br>";
$objRuzneVyjimky->setJmeno("");
echo "Za neosetřenou výjimkou<br>";

?>
</body>
</html>
```