



Web Adv

PHP: Exceptions & errors

DE HOGESCHOOL MET HET NETWERK

Hogeschool PXL – Dep. PXL-IT – Elfde-Liniestraat 26 – B-3500
Hasselt
www.pxl.be - www.pxl.be/facebook



Inleiding

exception , error = uitzonderlijke gebeurtenis die zich kan voordoen tijdens uitvoering programma.

- exception kan worden opgevangen in een try... catch
- error niet
- try ... catch block
 - Exception opgeworpen in try?
 - executie van try wordt gestaakt, overeenkomstige catch wordt uitgevoerd

vb1.php

Berekening van de faculteit van de variabele \$a
Exceptie wordt opgeworpen als er een misse waarde in de variabele \$a staat waarvan het faculteit berekend wordt.

De exceptie is een object van de klasse Exception en wordt geworpen en gevangen binnen het hoofdprogramma zelf.

```
<?php
try{
    echo 'Begin van de try<br/>';
    $a = 12 ; //test ook de waarden -1, 12.2, 'test'
    $fac =1;
    if (is_integer($a)){
        if ($a >0){
            for ($i =1; $i< $a; $i++ ){
                $fac *= $i;
            }
        }else{
            if ($a == 0){
                $fac =1;
            } else {
                throw new Exception("fout: $a < 0");
            }
        }
    } else {
        throw new Exception("fout: $a is geen getal");
    }
    echo "<br/>$a! = $fac<br/>";
    echo 'Einde van de try<br/>';
} catch (Exception $e){
    echo '<pre>'; var_dump ($e); echo '</pre>';
}
```

Opdracht 1 Voer *vb1.php* uit. Pas ook *\$a* aan. Geef *\$a* de waarden *-1*, *12.2*, *'test'*.

Opdracht 2 Het niet opvangen van een opgeworpen *exception* leidt altijd tot het staken van de uitvoer van het *PHP-script*. Verwijder *try* en *catch* uit *vb1.php* en geef voor *\$a* waarde *'test'* in.

`getFile()` geef het bestand waarin de *exception* optrad

`getLine()` geef het lijnnummer van de code waar
de *exception* optrad

`getTraceAsString()` geef de stack trace van de fout

Een *Exception* kan ook afgedrukt worden via *print*. De methode `__toString` van de klasse *Exception* wordt dan automatisch aangeroepen.

Opdracht 3 Gebruik *print(\$e)* in het *catch-gedeelte* in *vb1.php*.

vb2.php

fout wordt opgeworpen in de functie som

de fout wordt niet behandeld in deze functie maar wel waar de functie aangeroepen werd

vb2.php, deel 1 (declaratie van de functie)

```
<?php
function som () {
    $args = func_get_args();
    $s = 0;
    $cnt = 0;
    foreach ($args as $v) {
        if (is_integer($v) || is_double($v)) {
            $s += $v;
        } else {
            $stringArgs= '(' . implode(' ', $args) . ')';
            throw new Exception("fout bij waarde $v, ".
                                " dit is element $cnt ".
                                " uit de rij $stringArgs");
        }
        $cnt ++;
    }
    return $s;
}
```

vb2.php, deel 2 'hoofdprogramma' waar de functie aangeroepen wordt

```
try    {
    echo '<br/>begin try <br/>';
    $a = som(12.333,1,3, 'test',12);
    echo '$a = ' . $a ;
    echo '<br/>einde try <br/>';
} catch (Exception $e){
    print($e);
}
```

Opdracht 4 Voer *vb2.php* uit. Waarvoor dient de *implode*? Zoek op in de documentatie.

Exceptions definiëren

i.p.v. object van de klasse Exception

throw en catch een object van een klasse afgeleid van Exception

Bekijk 'SomException.php' en 'vb3.php'

Errors en error levels

error = fout die niet opgevangen kan worden door try catch

e.g. fopen

Errors/Exceptions

If the open fails, an error of level **E_WARNING** is generated. You may use [@](#) to suppress this warning.

vb4.php

```
<?php
```

```
$f=fopen('ikbestaniet.txt', 'r+');  
// fout wordt gegenereerd
```

Warning: fopen(ikbestaniet.txt): failed to open stream: No such file or directory in
/home/jan/Desktop/werk/webtech2tin/les2_exceptions/vbn/vb4/vb4.php on line 2

NB: foutboodschap wordt afgedrukt, uitvoering wordt niet gestaakt!!

In het bestand php.ini (xampp/php/php.ini) wordt dit gedrag vastgelegd:

php.ini

; - Show all errors except for notices and coding standards warnings

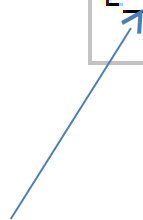
error_reporting = E_ALL & ~E_NOTICE

; Print out errors (as a part of the output). ...

display_errors = On

Verscheidene niveaus van errors in te stellen via functie error_reporting

E_WARNING	Non-fatal run-time errors. Execution of the script is not halted
E_NOTICE	Run-time notices. The script found something that might be an error, but could also happen when running a script normally
E_USER_ERROR	Fatal user-generated error. This is like an E_ERROR set by the programmer using the PHP function trigger_error()
E_USER_WARNING	Non-fatal user-generated warning. This is like an E_WARNING set by the programmer using the PHP function trigger_error()
E_USER_NOTICE	User-generated notice. This is like an E_NOTICE set by the programmer using the PHP function trigger_error()
E_RECOVERABLE_ERROR	Catchable fatal error. This is like an E_ERROR but can be caught by a user defined handle (see also set_error_handler())
E_ALL	All errors and warnings, except level E_STRICT (E_STRICT will be part of E_ALL as of PHP 6.0)



Strengste niveau

vb5.php

```
<?php  
error_reporting(E_ALL);  
print($var);  
?>
```

Opdracht: Probeer vb5.php met en zonder de functie error_reporting.

Behandelen en negeren van errors

Foutboodschappen negeren (niet afdrukken) via @

```
@ $f=fopen('ikbestaniet.txt','r+');
```

Uitvoer staken via 'die'

```
$f=fopen('ikbestaniet.txt','r+') or die("stop");
```

Combinatie van @ en die

```
@ $f=fopen('ikbestaniet.txt','r+') or die("stop");
```

vb6-vb8

set_error_handler

Via de functie `set_error_handler` kan ervoor gezorgd worden dat een functie gebruikt wordt om alle fouten te behandelen.

In vb9.php wordt getoond dat (via een omweg) een error ook als een exception behandeld kan worden


```
<?php
error_reporting(E_ALL);
set_error_handler('myErrorHandler');

function myErrorHandler($errno, $errstr, $file, $line) {
    throw new Exception("$errstr");
}

try{
    print ("begin");
    print($a);
    print("einde");
} catch (Exception $e) {
    print($e);
}
```

Errors genereren

```
<?php
```

```
function bekijkInhoudBestand($bestand) {  
    if (file_exists($bestand) && is_readable($bestand)) {  
        $file = fopen($bestand, 'r');  
        $inhoud = fread($file, filesize($bestand));  
        fclose($file);  
        print ("<pre>$inhoud</pre>");  
    } else {  
        trigger_error("...", E_USER_NOTICE);  
    }  
}
```

```
bekijkInhoudBestand('./a.txt');  
bekijkInhoudBestand('./a2.txt');
```

Oefeningen

Vertrek van de klasse Datum (zie vorige week). Plaats deze in de map 'tijd' en gebruik de namespace 'tijd' in de klasse.

Voorzie de foutbehandeling bij de klasse Datum. Zorg ervoor dat enkel positieve gehele getallen ingegeven mogen worden voor \$dag, \$maand, \$jaar.

Maak gebruik van de klasse DatumException. Deze klasse plaats je ook in de map en de namespace 'tijd'.

Test de code uit in het bestand Test.php. Dit bestand komt in de default namespace.

Bronnen



<http://php.net/manual/en/language.exceptions.php>
<http://php.net/manual/en/language.errors.php>



<http://phpro.org/tutorials/PHP-Exceptions.html>
<http://phpro.org/examples/PHP-Errors-As-Exceptions.html>