

MÓDULO OP2 – TÉCNICAS DE DETECÇÃO E TRATAMENTO DE ERROS

Programação e Sistemas de Informação

Módulo OP2

Técnicas de Detecção e Tratamento de Erros

Erros

Como já têm tido oportunidade de constatar, com mais frequência do que concerteza desejavam, é vulgar acontecerem erros a nível da programação.

- O nome vulgar dado a um erro de programação é:

BUG

- A detecção / eliminação de erros, tem também uma designação própria:

DEBUGGING

- É um dos motivos de frustração para quem se inicia na programação: a não detecção de bugs. Algo que se vai resolvendo com experiência, calma e perspicácia...



MÓDULO OP2 – TÉCNICAS DE DETECÇÃO E TRATAMENTO DE ERROS

Como Detectar?

Não existe nenhuma fórmula mágica para a detecção de erros...

Mas é óbvio que se podem utilizar algumas estratégias, como por exemplo:

- Tentar isolar o erro

Pode-se utilizar “couts” para verificar o que o programa está a fazer em determinada parte do código, ou mesmo passo a passo.

- Utilização de debuggers

Programas que permitem executar o código passo a passo e ver o conteúdo das variáveis utilizadas a cada momento.



MÓDULO OP2 – TÉCNICAS DE DETEÇÃO E TRATAMENTO DE ERROS

Tipos de Erros

Existem três tipos de erros:

- Erros de Sintaxe;
- Erros Lógicos;
- Erros em tempo de execução(runtime).

MÓDULO OP2 – TÉCNICAS DE DETECÇÃO E TRATAMENTO DE ERROS

Tipos de Erros – Erros de Sintaxe

- Programa que escrevemos não obedece às regras gramaticais da linguagem;
- Normalmente são fáceis de detectar;
- Quando o compilador encontra erros sintácticos mostra mensagem de erro;
- Como consequência do ponto anterior, o programa não compila;
- Por exemplo:
 - Esquecer um ponto e vírgula ou colocar antes uma vírgula;
 - Esquecer de fechar uma chaveta, aspas ou comentário;
 - Escrever mal uma das palavras reservadas pela linguagem;
 - ...

Tipos de Erros – Erros de Lógicos

- Tipos de erros mais difícil detecção que os anteriores;
- Não são detectados na compilação, ou seja, programa compila sem erros;
- Não terminam a execução do programa;
- Têm a particularidade de o programa não fazer aquilo que era esperado, produzindo um resultado que não era o resultado desejado;
- Por exemplo:
 - Erro num parâmetro do cálculo de uma expressão;
 - Má avaliação de uma expressão lógica;
 - Ciclos infinitos;
 - ...

MÓDULO OP2 – TÉCNICAS DE DETEÇÃO E TRATAMENTO DE ERROS

Tipos de Erros – Erros de Runtime

- Este tipo de erros, tal como os anteriores, não são detectados na compilação;
- O erro só acontece aquando da execução do programa;
- Mal o erro é detectado, o programa termina e é mostrada uma mensagem de erro;
- Por exemplo:
 - Divisão por zero;
 - ...

Alguns erros típicos...

- Não inicializar variáveis;
- Não colocação de ponto e vírgula ou então a sua colocação em local indevido;
- Confusão entre os sinais de atribuição (=) e de igualdade (==);
- Ciclos infinitos;
- Agrupamento de instruções (em instruções como os if e os for)



MÓDULO OP2 – TÉCNICAS DE DETEÇÃO E TRATAMENTO DE ERROS

Instalação XDebug

<https://www.youtube.com/watch?v=xME6uHYTcLU>

<https://youtu.be/r4crYOCdm20>

Lê os passos seguintes, não instales ainda, depois de leres visualiza os 2 vídeos anteriores, tirando notas da instalação e depois sim, procede à instalação e configuração do XDebug

Uma das ferramentas mais úteis no desenvolvimento de software é um depurador apropriado. Ele permite que tracemos a execução do código e monitorize os itens na pilha de execução. XDebug, um depurador de PHP, pode ser utilizado por várias IDEs para colocar *breakpoints* e inspecionar o programa. Ele também permite que ferramentas como PHPUnit e KCacheGrind realizem análise de cobertura e perfis de código.

Quando não se consegue sair do lugar, querendo recorrer a *var_dump/print_r*, e ainda assim não conseguir resolver o problema – é altura de usar um debugger.

Depurar, em inglês significa *debug*, é o processo de encontrar e reduzir defeitos numa aplicação de software ou até mesmo de hardware. Quando citamos linguagens de programação para desenvolvimento Desktop, a depuração é comum e uma rotina para o programador, visto que a grande maioria das IDE (*Integrated Development Environment*) já suportam a depuração do código de forma completa.

MÓDULO OP2 – TÉCNICAS DE DETECÇÃO E TRATAMENTO DE ERROS

Ferramenta case XDebug

O primeiro passo é realizar a instalação do XDebug.

Na instalação deve seguir os passos abaixo:

No Vs Code cria um ficheiro em php com o seguinte conteúdo `phpinfo()`;

Visualiza o ficheiro no browser

Agora acede a <https://xdebug.org/wizard>

Copia e cola toda a informação resultante do `phpinfo()` para dentro da box

```
PHP Quality Assurance Team
Ilia Alshanetsky, Joerg Behrens, Antony Dovgal, Stefan Esser, Moriyoshi Koizumi,
Magnus Maatta, Sebastian Nohn, Derick Rethans, Melvyn Sopacua, Pierre-Alain
Joye, Dmitry Stogov, Felipe Pena, David Soria Parra, Stanislav Malyshev, Julien
Pauli, Stephen Zarkos, Anatol Belski, Remi Collet, Ferenc Kovacs
Websites and Infrastructure team
PHP Websites Team      Rasmus Lerdorf, Hannes Magnusson, Philip Olson, Lukas
Kahwe Smith, Pierre-Alain Joye, Kalle Sommer Nielsen, Peter Cowburn, Adam
Harvey, Ferenc Kovacs, Levi Morrison
Event Maintainers      Damien Seguy, Daniel P. Brown
Network Infrastructure  Daniel P. Brown
Windows Infrastructure  Alex Schoenmaker
PHP License
This program is free software; you can redistribute it and/or modify it under
the terms of the PHP License as published by the PHP Group and included in the
distribution in the file: LICENSE

This program is distributed in the hope that it will be useful, but WITHOUT ANY
WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.

If you did not receive a copy of the PHP license, or have any questions about
PHP licensing, please contact license@php.net.

1die
```

Agora analise a informação do teu PHP e configurações

Analyse my `phpinfo()` output

Aparece agora toda a informação



MÓDULO OP2 – TÉCNICAS DE DETEÇÃO E TRATAMENTO DE ERROS

If you find Xdebug useful, please consider [supporting the project](#).

Installation Wizard

Summary

- **Xdebug installed:** 2.8.0
- **Server API:** Built-in HTTP server
- **Windows:** yes
- **Compiler:** MS VC15
- **Architecture:** x64
- **Zend Server:** no
- **PHP Version:** 7.3.11
- **Zend API nr:** 320180731
- **PHP API nr:** 20180731
- **Debug Build:** no
- **Thread Safe Build:** yes
- **OPcache Loaded:** no
- **Configuration File Path:** C:\WINDOWS
- **Configuration File:** C:\Ampps\php\php.ini
- **Extensions directory:** C:\Ampps\php-7.3\ext

Atenção à referência do Compiler



MÓDULO OP2 – TÉCNICAS DE DETEÇÃO E TRATAMENTO DE ERROS

Instructions

1. Download `php_xdebug-3.1.3-7.3-vc15-x86_64.dll`
2. Move the downloaded file to `C:\Ampps\php-7.3\ext`, and rename it to `php_xdebug.dll`
3. Update `C:\Ampps\php\php.ini` to have the line:
`zend_extension = xdebug`
4. Restart PHP's built-in HTTP server (`php -S`)

Enabling Features

Now Xdebug is installed, you can enable its features. Please refer to the dedicated sections in the documentation about information on how to enable and configure these Xdebug features. Where these sections refer to `php.ini` or similar, please remember to use `C:\Ampps\php\php.ini`:

- **Development Helpers** — help you get better error messages and obtain better information from PHP's built-in functions.
- **Step Debugging** — allows you to interactively walk through your code to debug control flow and examine data structures.
- **Profiling** — allows you to find bottlenecks in your script and visualize those with an external tool.

Faça o download e siga as instruções anteriores.