

Exceções & Ficheiros

Prof. Dr. Tiago Araújo

tiagodavi70@ua.pt

$$0/0 = ?$$

Tipos de Erros

As linguagens modernas de programação possuem um poderoso esquema para tratamento de erros baseado em tratamento de exceções

Existem erros que são detectáveis no momento da compilação, por falha do programador

- *Compile-time errors*

Exemplos

- Associar valores a variáveis de tipos diferentes
- Invocar um método inexistente
- Tentar acessar um campo com acesso restrito
- Tentar acessar uma variável de instância em um método estático

Tipos de Erros

Há erros, porém, que podem surgir durante a execução do programa

- *Runtime errors*

Exemplos

- Extrapolar limites de um vetor
- Divisão por zero
- Casting explícito inapropriado
- Tentar acessar um arquivo que não existe
- Tentar gravar em um disco cheio
- Tentar acessar um diretório que não há permissão

Exceções

O sistema de tratamento de exceções se refere aos erros em tempo de execução

Toda vez que um erro ocorre, uma exceção (objeto) é criada e lançada

A exceção (objeto) encapsula as informações do erro

Essa exceção deve ser capturada em algum momento

- Cascata de chamadas de métodos
- Bloco try-catch, try-catch-finally

Exceções

Quando surge um erro na execução de um método, o método cria um objeto de exceção e passa para o sistema de runtime do java (lança a exceção)

A execução do programa para naquele ponto e o runtime procura o trecho de código mais próximo (stack) capaz de capturar uma exceção daquele tipo

Se a busca chegar ao main e não encontrar quem capture a exceção, o programa termina

Exceções

Quando uma exceção é lançada, o fluxo do programa é interrompido naquele instante

Se for dentro de um try, a execução é deslocada para o bloco catch que captura aquele tipo de exceção

Se não houver bloco try-catch é porque o método lança aquele tipo de exceção

Neste caso, o método retorna levando a exceção para quem o chamou

Se não houver exceção, apenas o bloco try é executado

Pula os blocos catch e prossegue o programa

Ficheiros

Ficheiro

Um ficheiro é um conjunto de dados relacionados que os computadores tratam como uma única unidade.

Os computadores armazenam ficheiros no armazenamento secundário para que o conteúdo dos ficheiros permaneça intacto quando um computador é desligado.

Quando um computador lê um ficheiro, copia-o do dispositivo de armazenamento para a memória.

Quando escreve num ficheiro, transfere os dados da memória para o dispositivo de armazenamento.

Operações

As linguagens de programação de alto nível suportam operações de ficheiros

Atribuição de nomes

Abertura

Leitura

Escrita

Encerramento

Operações

Criar o *stream* (fluxo) através de uma variável

Abrir o ficheiro, associando o nome do stream ao nome do ficheiro.

Ler ou escrever os dados.

Fechar o ficheiro.