

FE1

Ferramenta de Desenvolvimento de Aplicativos 1

Erros de Compilação e Execução - 02

Alencar de Melo Junior / Edgar Noda

Curso Técnico em Informática – IFSP Hortolândia 2011

Original de Helder da Rocha

Erros de Compilação e Execução

- Os erros podem ocorrer em dois domínios:
 - tempo de **compilação** e
 - tempo de **execução**
- Erros ocorridos durante a fase de **compilação** ocorrem quando se executa o **javac**.
 - **Erros de processamento do arquivo (parsing)**: ponto-e-vírgula faltando, parênteses, aspas, chaves ou colchetes descasados. Identifica apenas o arquivo e a linha onde o erro **pode** ter iniciado. Um erro causa vários outros e nem sempre a mensagem é precisa.
 - **Erros de compilação do código**, realizada depois do parsing: além da linha e do arquivo, identificam a classe e método. Geralmente as mensagens são bastante elucidativas.

2

Exemplos de erros de compilação comuns

▪ *Cannot resolve symbol:*

Compilador é incapaz de localizar uma definição do símbolo encontrado. Causas comuns:

- Erro de sintaxe no nome de variável ou método
- Variável/método não declarado
- Classe usada não possui variável, método ou construtor
- Número ou tipo de argumentos do método ou construtor incorretos
- Definição de classe não encontrada no CLASSPATH

3

Exemplos de erros de compilação comuns

▪ *Class Hello is public, should be declared in a file named Hello.java:*

Nome do arquivo tem que ser igual ao nome da classe pública*:

- Exemplo: Nome tem que ser Hello.java, literalmente. O nome hello.java causa este erro porque o "h" está minúsculo.

* Se classe não for pública, essa restrição não vale

4

Exemplos de erros de compilação

▪ Erro de parsing

- Na verdade, só há um erro no código, apesar do compilador acusar três

Apenas o primeiro erro é verdadeiro.

Ignore os outros. Eles foram causados pelo primeiro.

```
1: public class HelloWorldErro {
2:     public static void main(String args {
3:         System.out.println("Hello, 4: world!");
5:     }
6: }
```

Arquivo onde foi detectado o erro

Trecho do código e indicação da provável localização da causa do erro

```
C:\aulajava\files\cap01\erro>javac HelloWorldErro.java
HelloWorldErro.java:2: ')' expected
    public static void main(String args {
                                ^
HelloWorldErro.java:4: ';' expected
    HelloWorldErro.java:4: ';' expected
                                ^
HelloWorldErro.java:2: missing method body, or declare abstract
    public static void main(String args {
                                ^
3 errors
```

5

Exemplos de erros de compilação

```
C:\aulajava\files\cap01\erro>javac MensagemErro.java
MensagemErro.java:6: class mensagemerro is public, should be declared in a file
named mensagemerro.java
public class mensagemerro {
      ^
MensagemErro.java:8: cannot resolve symbol
symbol  : class string
location: class mensagemerro
    public string nome = "";
           ^
MensagemErro.java:11: cannot resolve symbol
symbol  : class string
location: class mensagemerro
    public string lerNome() {
           ^
MensagemErro.java:12: cannot resolve symbol
symbol  : class string
location: class mensagemerro
    string nomeEmMaiusculas = msg.toUpperCase();
                                ^
MensagemErro.java:12: cannot resolve symbol
symbol  : variable msg
location: class mensagemerro
    string nomeEmMaiusculas = msg.toUpperCase();
                                ^
5 errors
```

Nome do arquivo é **MensagemErro.java** mas classe foi criada com nome **mensagemerro.java**

Compilador não sabe quem é **string**: O tipo **String** sempre tem um **S** maiúsculo (como todas as classes da API)

Compilador não sabe quem é **msg**: não foi declarada nenhuma variável com esse nome.

```
5: public class mensagemerro {
6:     /** Atributo de dados msg é publicamente visível */
7:     public string nome = "";
8:
9:     /** Método lerNome() devolve objeto do tipo String */
10:    public string lerNome() {
11:        string nomeEmMaiusculas = msg.toUpperCase();
12:    }
13: }
14: }
```

6

Exemplos de erros de compilação

Erros em múltiplas classes

- Quando uma classe que possui **dependências** é compilada, suas dependências são compiladas primeiro e o compilador mostra mensagens de erros referentes a todas as classes envolvidas
- Identifique sempre o arquivo e o número da linha
- Compile as dependências primeiro

Erro na linha 11
do arquivo
HelloJava.java

Erro na linha 12
do arquivo
Mensagem.java
localizado no
mesmo diretório
que HelloJava.java

```

C:\aulajava\files\cap01\erro>javac HelloJava.java
HelloJava.java:11: ')' expected
    public static void main(String[] args {
                        ^
HelloJava.java:16: ';' expected
    }
    ^
HelloJava.java:11: missing method body, or declare abstract
    public static void main(String[] args {
                        ^
.\Mensagem.java:12: cannot resolve symbol
  symbol:   variable msg
  location: class Mensagem
    String nomeEmMaiusculas = msg.toUpperCase();
                                ^
4 errors
  
```

Erros durante a Execução (Runtime)

- Depois que o código compila com sucesso, os bytecodes (arquivos .class) são gerados e podem ser usados em um processo de execução
- Erros ocorridos durante a fase de execução (runtime)** ocorrem quando se executa o interpretador **java**, e são mais difíceis de localizar e consertar.

Erros durante a Execução (Runtime)

- A mensagem impressa geralmente é um "stack trace" e mostra todo o "caminho" percorrido pelo erro
 - *Relaciona métodos e classes* da sua aplicação e classes da API Java que sua aplicação usa (direta ou indiretamente)
 - *Nem sempre mostra a linha de código onde o erro começou*
 - *O início do trace geralmente contém informações mais úteis*
- Erros de runtime *nem sempre* indicam falhas no software
 - Frequentemente se devem a *causas externas*: não existência de arquivos externos, falta de memória, falha em comunicação de rede

9

Exemplos de erros de execução comuns

- *Exception in thread "main": ClassNotFoundException: Classe:* a classe "Classe" não foi encontrada no CLASSPATH.
 - O CLASSPATH não inclui todos os diretórios requeridos
 - O nome da classe foi digitado incorretamente ou requer pacote
- *Exception in thread "main": NoSuchMethodError: main:* o sistema tentou chamar main() mas não o encontrou.
 - A classe não tem método main() (talvez não seja executável)
 - Confira *assinatura* do main: `public static void main(String[] args)`

10

Exemplos de erros de execução comuns

- **ArrayIndexOutOfBoundsException:**
programa tentou acessar vetor além dos limites definidos.
 - Erro de lógica com vetores
 - Aplicação pode requerer argumentos de linha de comando
- **NullPointerException:** referência para objeto é nula
 - Variável de tipo objeto foi declarada mas não inicializada
 - Vetor foi declarado mas não inicializado

11

Exemplos de erros de tempo de execução

```

C:\aulajava\files\cap01\erro>java HelloJava
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException
    at HelloJava.main(HelloJava.java:13)
  
```

Na linha **13**, está havendo uma referência a um índice de um vetor. Esse índice tenta acessar uma posição inexistente no vetor. Neste caso específico, o erro pode ser evitado passando-se um parâmetro após o nome da classe na linha de comando, porém a aplicação é pouco robusta pois não prevê que o erro possa acontecer.

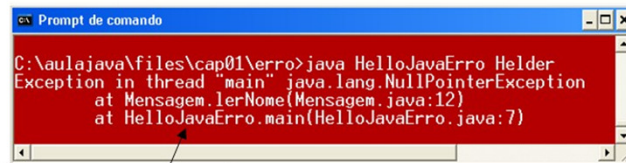
```

C:\aulajava\files\cap01\errobin>java hello
Exception in thread "main" java.lang.NoClassDefFoundError: hello (wrong name: Hello)
    at java.lang.ClassLoader.defineClass0(Native Method)
    at java.lang.ClassLoader.defineClass(ClassLoader.java:509)
    at java.security.SecureClassLoader.defineClass(SecureClassLoader.java:124)
    at java.net.URLClassLoader.defineClass(URLClassLoader.java:246)
    at java.net.URLClassLoader.access$100(URLClassLoader.java:54)
    at java.net.URLClassLoader$1.run(URLClassLoader.java:193)
  
```

A **classe** não foi encontrada. Pode ser que ela não esteja no CLASSPATH. Neste caso específico o interpretador nos sugere que o nome pode estar errado. Para consertar, basta chamar a classe pelo nome correto: **Hello**. O Stack Trace mostra que esse erro teve origem em outras classes, mas foi nossa classe que, na verdade, o provocou. 12

Exemplos de erros de tempo de execução

- Erros de tempo de execução freqüentemente ocorrem em dependências
 - Causa *pode* estar na dependência
 - Causa *pode* ter tido origem na dependência mas ter sido iniciada por erro na classe principal
- Stack Trace *pode* ajudar a localizar a origem do erro
 - As informações também podem desviar a atenção



```

C:\aulajava\files\cap01\erro>java HelloJavaErro Helder
Exception in thread "main" java.lang.NullPointerException
    at Mensagem.lerNome(Mensagem.java:12)
    at HelloJavaErro.main(HelloJavaErro.java:7)
  
```

Erro começou na linha 7 (método main) de HelloJavaErro mas teve origem na linha 12 do método lerNome de Mensagem

13

Erros em tempo de execução

- Há dois tipos de erros que surgem em tempo de execução:
 - Causados por situações externas, que fogem do controle do programador (ex: rede fora do ar)
 - Causados por erros de lógica de programação
- Deve-se criar aplicações robustas que prevejam a possibilidade de erros de tempo de execução devido a fatores externos e ajam da melhor forma possível

14

Erros em tempo de execução

- *Deve-se achar erros de lógica e evitar que sobrevivam além da fase de desenvolvimento.*
- *Para evitá-los:*
 - *Escreva código claro, fácil de entender, organizado, pequeno*
 - *Use endentação, siga convenções, nomes significativos, documente*
 - *Escreva testes para todo código e rode-os com frequência*
- *Para achar os erros difíceis*
 - *Rode código de testes (se os tiver); Ative nível de mensagens de log*
 - *Aprenda a usar um depurador para navegar no fluxo de execução*

15

Referências Bibliográficas

Rocha, H. da. Curso J100: Java 2 Standart Edition.
Revisão 17.0. www.argonavis.com.br, 2003.

DEITEL, H. M.; DEITEL, P. J. **Java: Como Programar**. 6ª
edição, Pearson Prentice Hall, 2005. São Paulo.

16