

# Aprendendo com os Erros I

## 1. **javac: file not found: Teste.java**

Causa: tentativa de compilar um código-fonte (Teste.java) que não existe na pasta atual.

Solução: verificar a existência do arquivo, alterando sua localização e/ou nome.

## 2. **NoClassDefFoundError: principal**

Causa: tentativa de executar uma classe que não existe.

Solução: verificar se o nome da classe está correto.

## 3. **NoSuchMethodError: main**

Causa: tentativa de executar um classe sem o método main.

Solução: verificar se o método existe e se a sua assinatura está correta `public static void main(String a[])`.

## 4. **reached end of file while parsing**

Causa: o fim do arquivo foi atingido enquanto a classe estava sendo processada.

Solução: provavelmente a chave final da classe está faltando.

## 5. **class, interface or enum expected**

Causa: apareceu alguma instrução diferente de `import`, `package`, `class`, `interface` ou `enum`. Provavelmente a declaração de classe contém erro de sintaxe.

Solução: corrigir erro de digitação.

## 6. **cannot find symbol: <qual> <onde>**

Causa: alguma variável ou método `<qual>` usado não está definido em `<onde>`.

Solução: definir o que não está definido ou corrigir erro de digitação.

## 7. **variable <nome> might not have been initialized**

Causa: tentativa de obter conteúdo de uma variável para a qual não foi atribuído valor e não possui valor padrão

Solução: inicializar a variável

## 8. **<nome> is already defined in <localização>**

Causa: tentativa de declarar algo que já foi declarado no mesmo escopo (região de visibilidade)

Solução: escolher outro identificador

9. **incompatible types: found <tipo1> required <tipo2>**

Causa: tentativa de usar um tipo de dado <tipo1> onde era esperado <tipo2>

Solução: corrigir erro de lógica, definindo o tipo adequado e compatível
10. **possible loss of precision**

Causa: tentativa de atribuir um valor a uma variável que não comporta plenamente o tipo do valor atribuído. Indica que pode haver perda de dados

Solução: trocar o tipo da variável para um mais abrangente ou realizar uma coerção
11. **integer number too large: 2147483648**

Causa: tentativa de usar como inteiro um valor maior ou igual a 2147483648

Solução: verificar e ajustar intervalo de valores
12. **class <nome da classe> is public; should be declared in file named <nome da classe>.java**

Causa: existe uma classe pública cujo nome não faz parte do nome do arquivo.

Solução: modificar o nome do arquivo, de modo que contenha o nome da classe.
13. **modified private not allowed here**

Causa: o modificador private foi inserido em local não permitido, por exemplo em uma declaração de classe não interna.

Solução: retirar o modificador private ou mover a classe para dentro de outra.
14. **ArrayIndexOutOfBoundsException: 2**

Causa: tentativa de acessar o elemento de índice 2 de um array.

Solução: verificar erro de lógica e se todos os argumentos da linha de comando, esperados pelo programa, foram de fato passados.
15. **NumberFormatException: For input string "xpto"**

Causa: tentativa de gerar um tipo numérico a partir de um String, sendo que o String não pode ser convertido em número.

Solução: tratar o erro com try/catch ou assegurar de outra forma que o String que se deseja converter contenha apenas dígitos.
16. **',' expected**

Causa: estritamente, indica a falta do ponto-e-vírgula, mas vezes é gerado por outro erro de construção imediatamente depois do ponto onde o erro está indicado.

Solução: inserir ponto-e-vírgula

**17. ')' expected**

Causa: falta fechamento dos parênteses

Solução: fechar os parênteses

**18. package <nome\_pacote> does not exist**

Causa: tentativa de usar pacote inexistente

Solução: provavelmente o nome de uma classe, como System, foi digitado com letra minúscula

**19. unclosed string literal**

Causa: faltou fechar (ou abrir) as aspas de uma String

Solução: fechar (ou abrir) a String