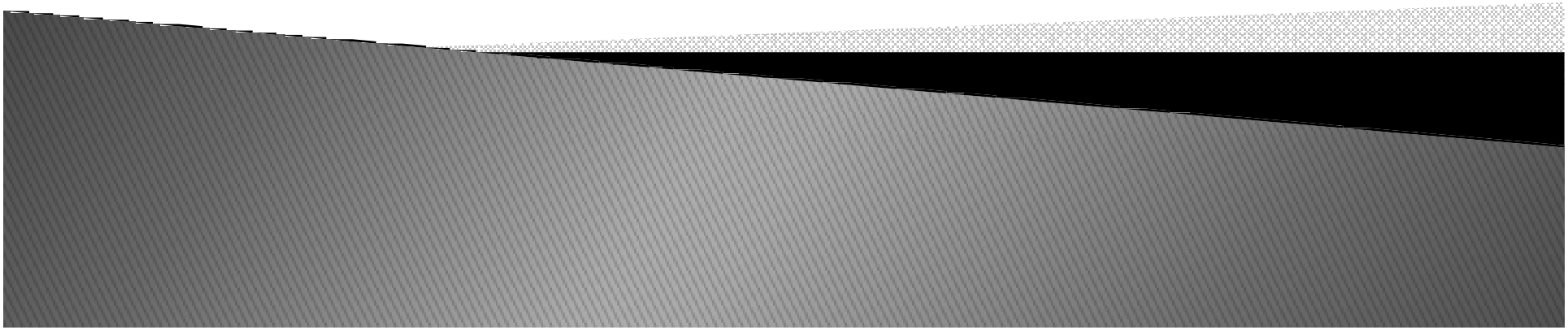
Sobrecarga de Métodos e Exceções

Aula09

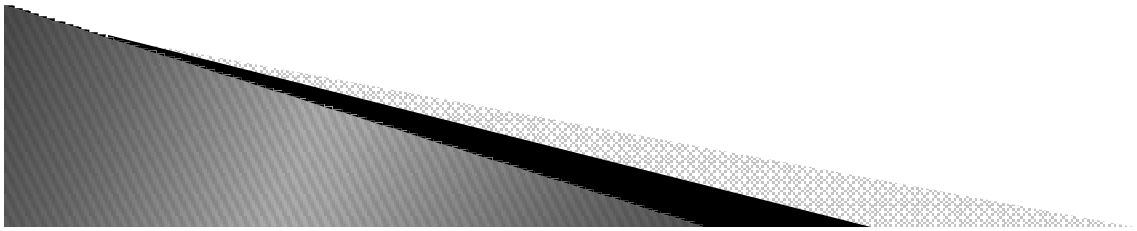
BSI – UFRPE
Prof. Gustavo Callou
gcallou@gmail.com



Sobrecarga de Métodos

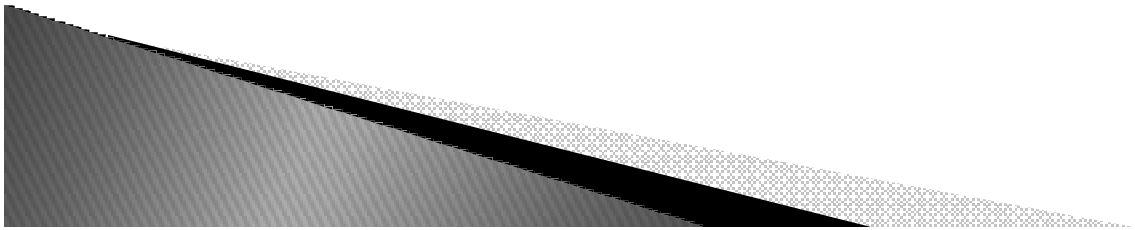
É permitido incluir numa classe métodos que possuem o *mesmo nome* e o *mesmo tipo de retorno*, mas que *diferem pelo número e/ou pelos tipos dos argumentos*.

Qual destes métodos é executado depende do número e/ou dos tipos dos argumentos passados pelo programa que chama o método. Esta técnica é chamada *sobrecarga*("*overloading*") de métodos. Ela é frequentemente utilizada, por exemplo para definir *vários construtores para uma determinada classe*.



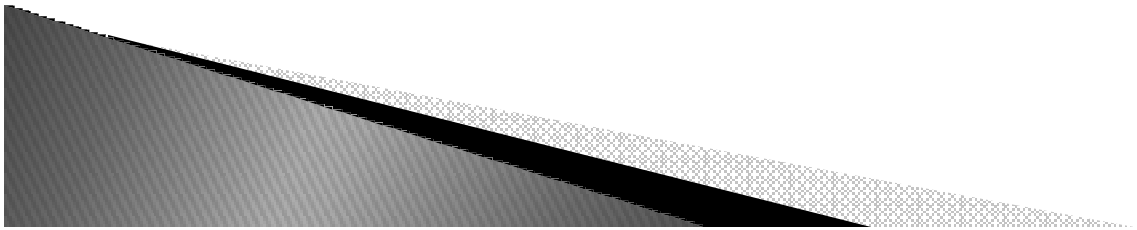
Exemplos

- //declaração de funções
- int f(int x) {return (1);}
- int f(float x) {return (2);}
- int f(float x, int y) {return (3);}
- int f(int x, float y) {return (4);}
- float f(int x, float y){return (5.3);}



Exemplos

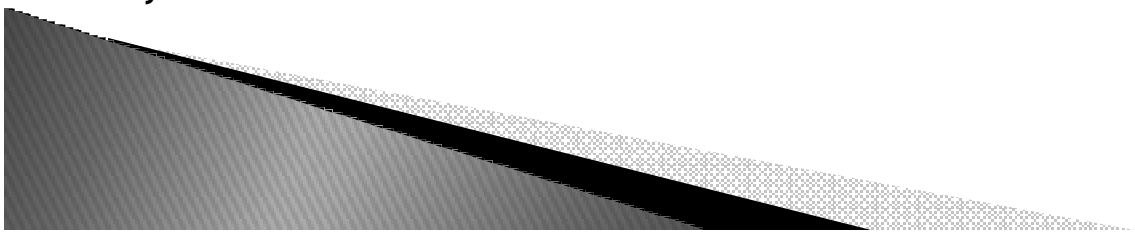
- //declaração de funções
- int f(int x) {return (1);}
- int f(float x) {return (2);}
- int f(float x, int y) {return (3);}
- int f(int x, float y) {return (4);}
- float f(int x, float y){return (5.3);} ⇐ //erro



Exemplo de sobrecarga em construtores

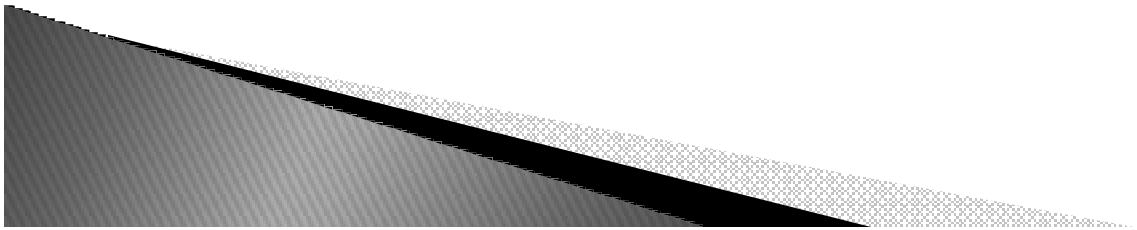
Exemplo:

```
public class Ponto{

 private double x, y;
 public Ponto(){
     setPonto( 0, 0 );
 }

 public Ponto( double a, double b ){
     setPonto( a, b );
   }
 public void setPonto( double a, double b ){
     x = a;
     y = b;
   }
 public String toString(){
     return "[" + x + ", " + y + "]";
   }
}
```



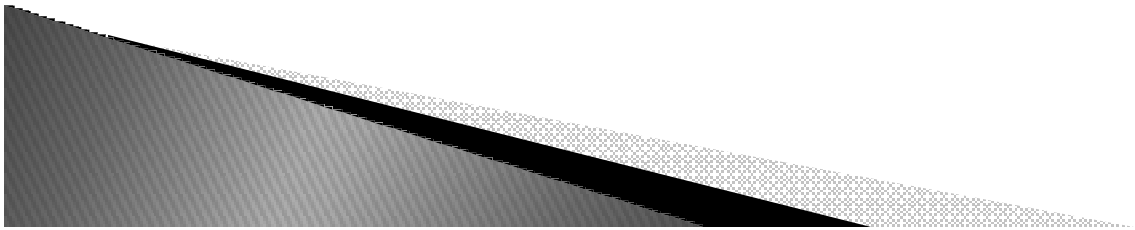
Exceções

- Exceções representam erros ocorridos no programa que interrompem o fluxo de execução normal.
- O processo de geração de exceções é chamado de disparo ou lançamento
- O tratamento de uma exceção disparada é chamado de captura
- Exceções não capturadas devem ser repassadas



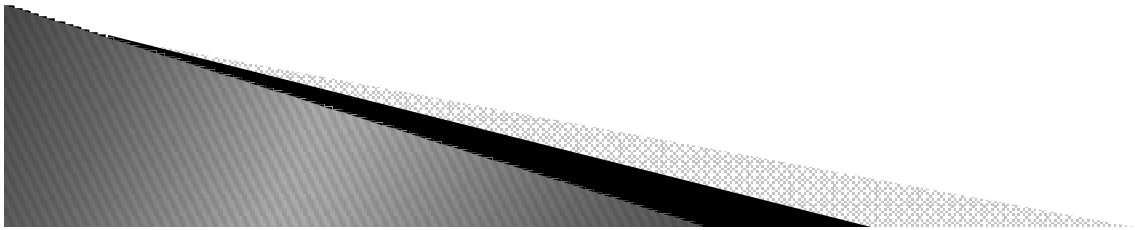
Exceções

- Exemplos típicos de exceções são:
  - Índice de uma lista (Array) fora do intervalo permitido.
  - Problemas em operações aritméticas, tais como "overflows" e divisões por zero.
  - Argumentos inválidos numa chamada a um método.
  - Uso de uma referência que não aponta para nenhum objeto.
  - Falta de memória (relativamente improvável em Java graças ao coletor de lixo).

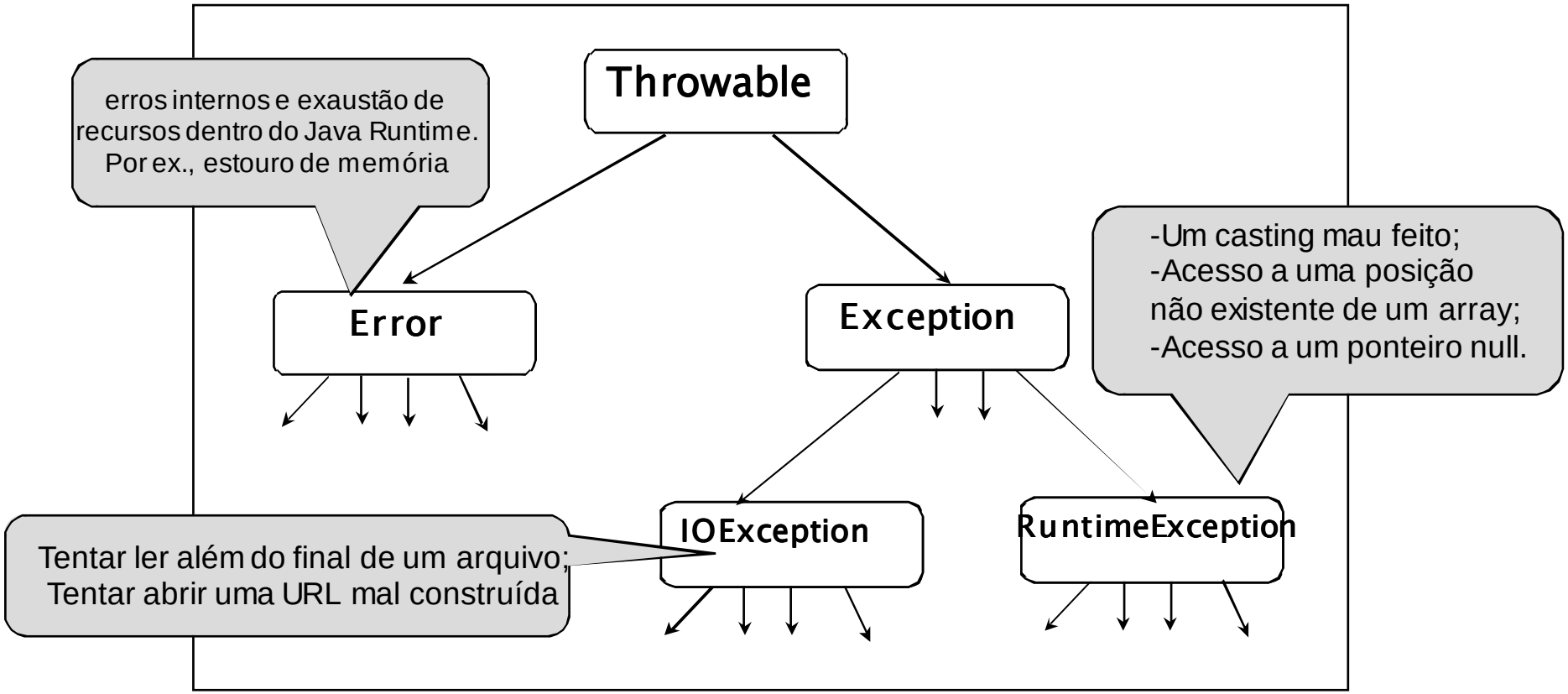


Tratamento de Exceções

- O tratamento de exceções em Java permite o gerenciamento de erros em tempo de execução.
- Uma exceção em Java é um objeto que descreve uma condição de exceção que ocorreu em algum fragmento de código.
- Quando surge uma condição excepcional, um objeto *Exception* é criado e lançado no método que causa a exceção.

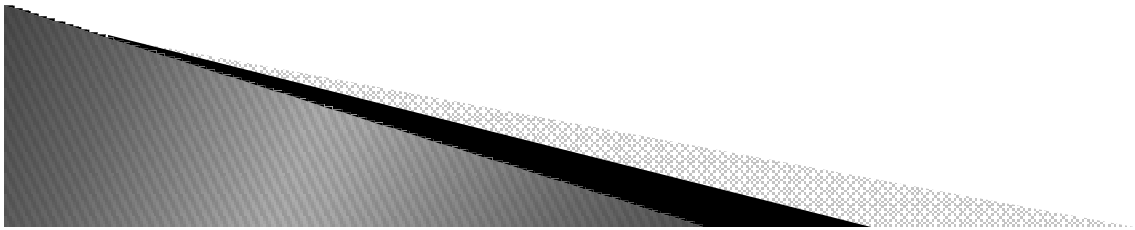


Classificação de tipos de Exceções



Exceções Não Capturadas

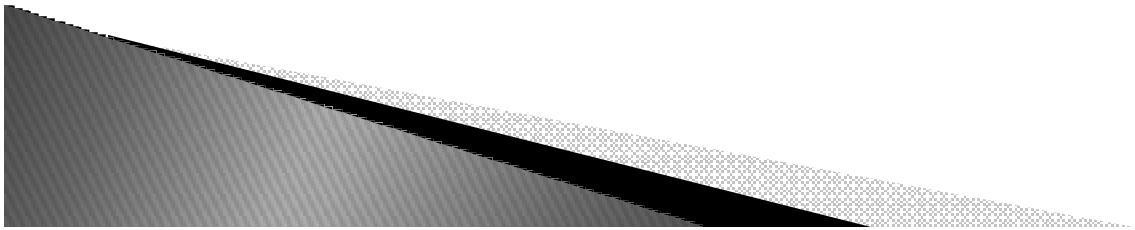
- Se dentro de um método uma exceção é lançada mas não é capturada ela deve ser "anunciada" ou documentada.
- Este anúncio é feito no cabeçalho do método
- Sintaxe:
- tipo método([params]) **throws** Exceção
- Exceções do tipo RuntimeException não devem ser capturadas



Captura de Exceções

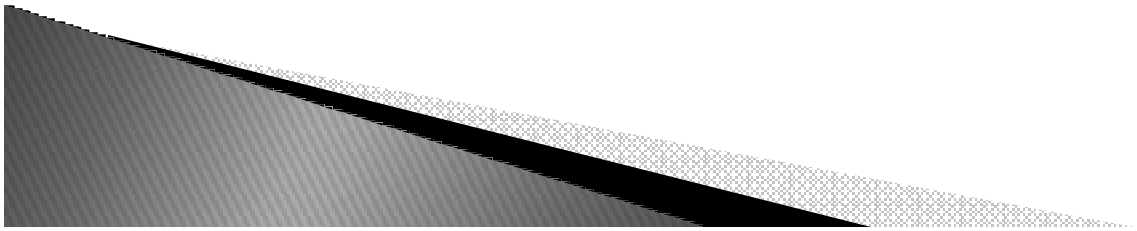
- Para tratar uma exceção ela deve ser capturada em um bloco **try** e manipulada em um **catch**
- Sintaxe:

```
try{
}
catch (Exception e){
}
```



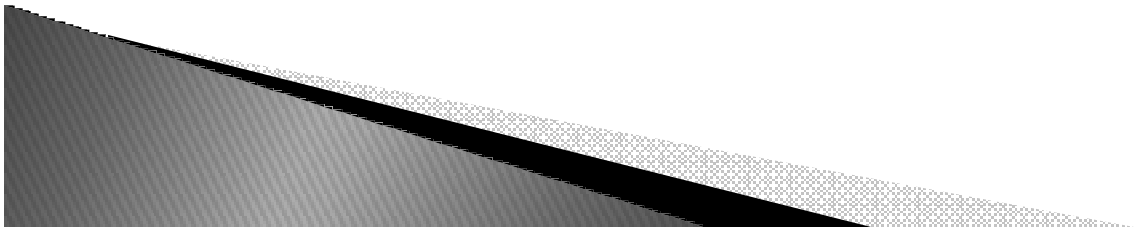
Captura de Exceções

- Dentro do bloco **try** deve ser inserido o código que poderá gerar uma exceção
- Se tudo correr bem, o bloco **catch** será pulado
- Caso contrário, o **catch** correspondente será ativado, independente da posição onde o programa estiver
- Se houver código que deva ser executado tanto em caso de erro quanto de acerto ele deve ser inserido no bloco **finally**



Exemplo

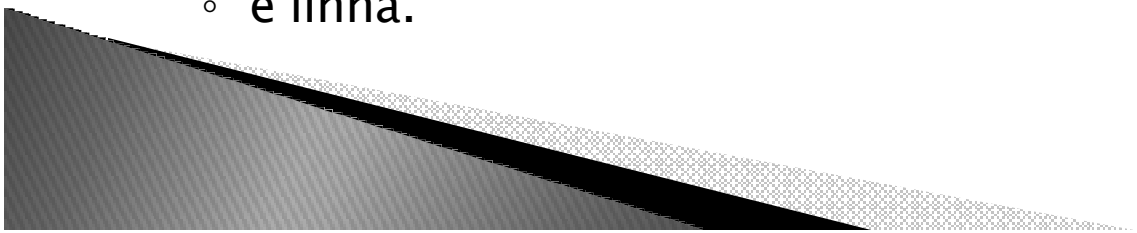
```
class ExcecaoDivisaoPorZero {
      public static void main (String args []){

        int d=0;
       int a=42/d;
       System.out.println ("Execução continua.");

      }
}
```



Exemplo

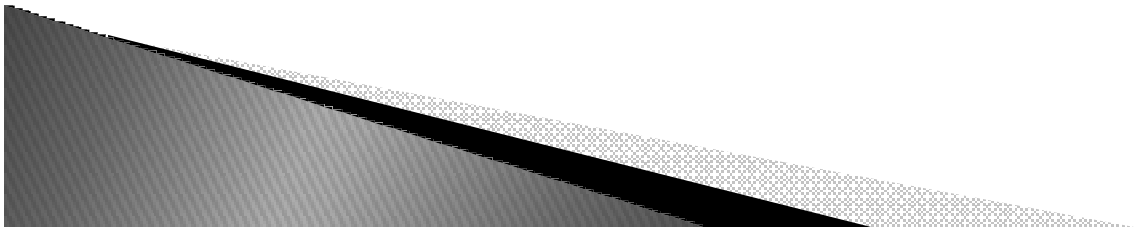
```
class ExcecaoDivisaoPorZero {
      public static void main (String args []){

        int d=0;
       int a=42/d;
       System.out.println ("Execução continua.");
      }
}
```

- O programa pára a execução e é chamado o manipulador de exceção padrão que mostra:
  - o tipo de exceção,
  - método em que ela aconteceu,
  - nome
  - o arquivo
  - e linha.



Solução

```
class ExcecaoDivisaoPorZero   {
  public static void main (String args []) {
       try{
          int d=0;
          int a=42/d;
           System.out.println ("Dentro do bloco da exceção.");
       }
       catch (ArithmeticException e) {
         System.out.println ("Aconteceu divisão por zero.");
       }
       System.out.println ("Execução continua.");
  }
}
```

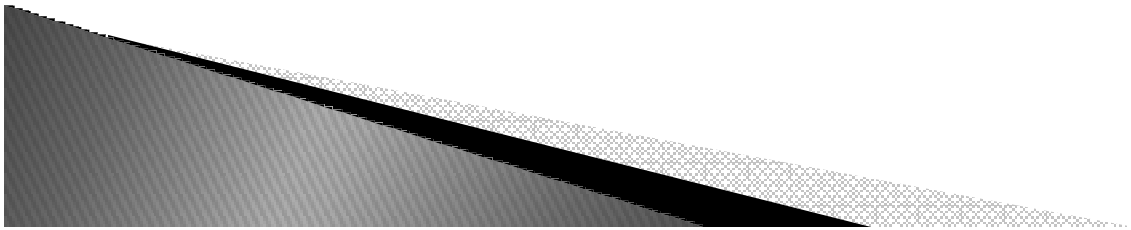


Exemplo2 – Pegando diferentes exeções

```
class MultiCatch {
  public static void main (String args [ ]) {
      try{
             int a = 10;
             int b = 42/a;
             int c [ ] = {1};
             c[42] = 99;
      }
      catch (ArithmeticException e) {
             System.out.println ("Div por 0: "+e);
      }
      catch (ArrayIndexOutOfBoundsException e) {
             System.out.println ("Estouro indice array: "+e);

       }
       System.out.println ("Execução continua.");
  }
}
```

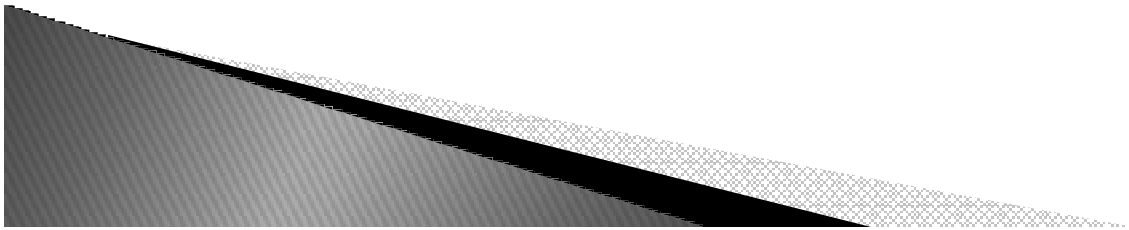
Exemplo em Python

```
try:
    a=10
    b=0
    c=a/b
    print (c)
except(ZeroDivisionError)as err:
    print ("erro na divisao: ", err)
```



Exemplo2 em Python

```
try:
    file = None
    file = open ("teste.txt","r")
    lista = file.readlines()
    print (lista)
except(IOError)as err:
    print ("arquivo nao existe: ", err)
finally:
    if file is not None:
        file.close()
```

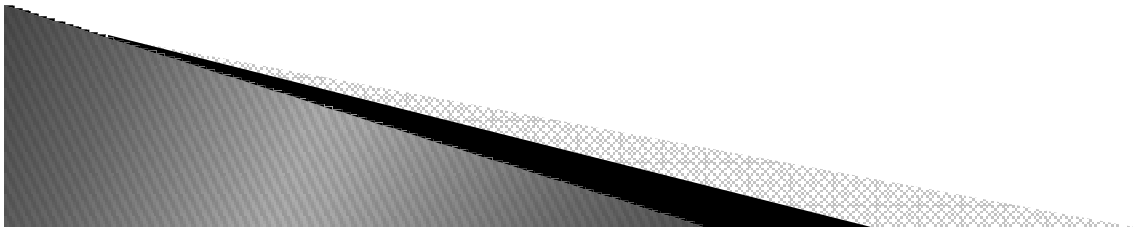


Exercício

- Passe para a máquina virtual java um argumento que seja lido pelo método main:
  - java ExercicioExcecao 5
  - No netbean, Menu run→ Set project Configuration →Customize→ Em Arguments colocar o valor → OK.
- Converta o argumento para inteiro usando:
  - int i = Integer.parseInt(args[0])
- Trate as exceções:
  - NumberFormatException
  - ArrayIndexOutOfBoundsException
- Teste para as três situações seguintes:
  - java ExercicioExcecao 5
  - java ExercicioExcecao
  - java ExercicioExcecao ola

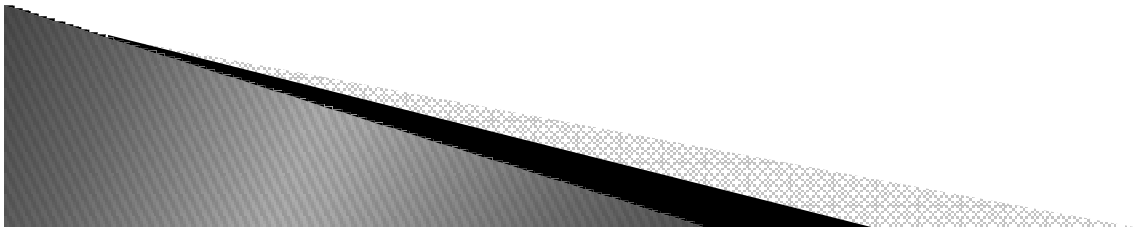
Disparo de Exceções

- Para disparar uma exceção basta criar um objeto de uma exceção qualquer e utilizar o **throw** para o lançamento
- Sintaxe:
- if (erro)
- throw new Exception(“Erro”);
- Esta exceção deve ser documentado no cabeçalho do método



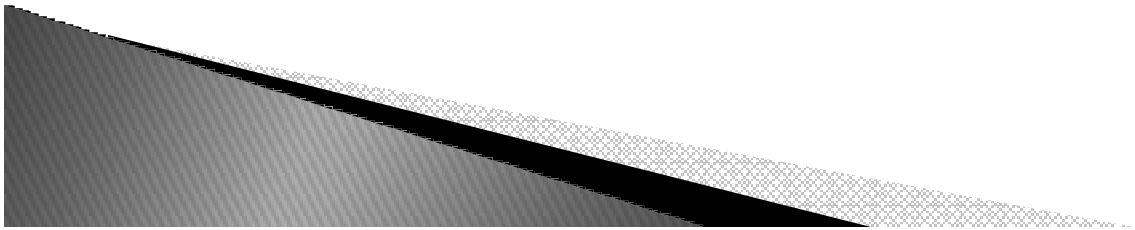
Métodos Úteis

- Os seguintes métodos podem ser usados:
- getMessage()
  - Retorna o texto passado para o construtor
- toString
  - Retorna uma string contendo o nome da exceção seguido do texto do contrutor
- printStackTrace
  - Imprime toda a pilha de execução
- finllInStackTrace
  - Reinsere a pilha de execucão na exceção



Exceções Próprias

- Para criar uma classe de exceção deve-se:
  - Derivá-la de Exception ou de uma de suas subclasses
  - Criar um construtor vazio e um com um parâmetro do tipo String
  - No construtor com o parâmetro chamar o construtor da classe mãe coma palavra super
  - Caso seja necessário podem ser inseridos outros métodos e atributos como uma classe comum

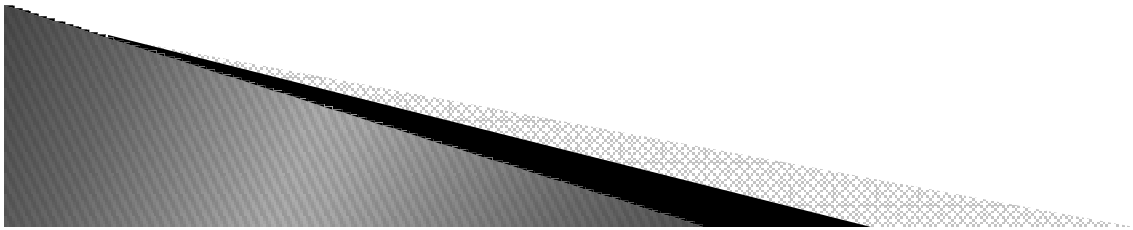


Exemplo

```
class MinhaExcecao extends Exception {
  private String detalhe;

  public MinhaExcecao() {
  }

   public MinhaExcecao(String a) {
      detalhe = a;
  }

  public String toString() {
      return "MinhaExcecao [" + detalhe + "]";
  }
}
```

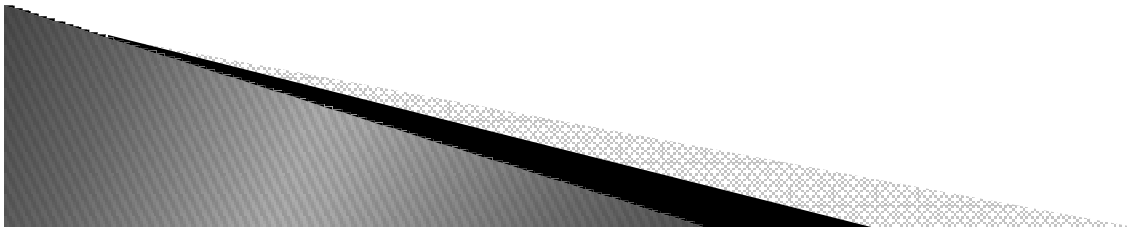
Exemplo(Cont.)

```
class DemoExcecao {
  public static void main(String args[]) {
      try {
         int a = 11;
         if (a > 10) {
             MinhaExcecao minhaExc = new
  MinhaExcecao("valor " + a + " maior que dez");
             throw minhaExc;
         }
      } catch (MinhaExcecao e) {
         System.out.println("Excecao capturada: " + e);
      }
}
```



Exemplo(Cont2.)

```
class DemoThrows   {
  public static void proced () throws MinhaExcecao{
      System.out.println ("No Procedimento.");
      throw new MinhaExcecao (1);
  }
  public static void main (String args []) {
      try {
             proced ();
      }
      catch (MinhaExcecao e) {
         System.out.println("Aconteceu exceção do"+
      "tipo MinhaExcecao.");
      }
  }
}
```



Referências

- http://www.javaworld.com/javaworld/jw-07-1998/jw-07-exceptions.html?page=1
- http://java.sun.com/docs/books/tutorial/essential/exceptions/definition.html
- http://www.if.ufrgs.br/~betz/jaulas/aula8o.htm#Exception