Programação orientada a objetos em Java
Tratamento de Exceções

Profª. Heloisa Moura

Programação orientada a objetos em Java
Tratamento de Exceções

Quando se cria programas de computador em Java, há possibilidade de ocorrer erros imprevistos durante sua execução, esses erros são conhecidos como exceções e podem ser provenientes de erros de lógica ou acesso a dispositivos ou arquivos externos.

O tratamento de exceção permite criar aplicativos que podem resolver (ou tratar) exceções. Em muitos casos, o tratamento de uma exceção permite que um programa continue executando como se nenhum problema tivesse sido encontrado.

Programação orientada a objetos em Java
Tratamento de Exceções

Alguns possíveis motivos externos para ocorrer uma exceção são:

Tentar abrir um arquivo que não existe.

Tentar fazer consulta a um banco de dados que não está disponível.

Tentar escrever algo em um arquivo sobre o qual não se tem permissão de escrita.

Tentar conectar em servidor inexistente.

Programação orientada a objetos em Java
Tratamento de Exceções

Alguns possíveis erros de lógica para ocorrer uma exceção são:

Tentar manipular um objeto que está com o valor nulo.

Dividir um número por zero.

Tentar manipular um tipo de dado como se fosse outro.

Tentar utilizar um método ou classe não existentes.

Programação orientada a objetos em Java
Tratamento de Exceções

Tratando Exceções

Uma maneira de tentar contornar esses imprevistos é realizar o tratamento dos locais no código que podem vir a lançar possíveis exceções, como por exemplo, campo de consulta a banco de dados, locais em que há divisões, consulta a arquivos de propriedades ou arquivos dentro do próprio computador.

Para tratar as exceções em Java são utilizados os comandos try e catch.

Programação orientada a objetos em Java
Tratamento de Exceções

Sintaxe:

```
try
{
        //trecho de código que pode vir a lançar uma
exceção
}
catch(tipo_exceçao_1 e)
{
        //ação a ser tomada
}
catch(tipo_exceçao_2 e)
{
        //ação a ser tomada
}
catch(tipo_exceçao_n e)
{
        //ação a ser tomada  }
```

Programação orientada a objetos em Java
Tratamento de Exceções

Onde:
try{ … } - Neste bloco são introduzidas todas as linhas de código que podem vir a lançar uma exceção.
catch(tipo_excessao e) { … } - Neste bloco é descrita a ação que ocorrerá quando a exceção for capturada.
Exemplificando uma exceção
Imagine uma classe que tem um método principal main que tem como seu único objetivo alterar todas as letras de uma frase para maiúsculas utilizando o método toUpperCase() da classe String, caso a frase esteja nula e se tente usar o método toUpperCase() na mesma será lançada uma exceção de NullPointerException.

Programação orientada a objetos em Java
Tratamento de Exceções

Primeiro vamos ver como ficaria a tal classe sem a utilização do try/catch.

```
public class TesteEnviaExcecao {

        public static void main(String args[]) {
                String frase = null;
                String novaFrase = null;
                novaFrase = frase.toUpperCase();
                System.out.println("Frase antiga: "+frase);
                System.out.println("Frase nova: "+novaFrase);
        }
}
```

Programação orientada a objetos em Java
Tratamento de Exceções

Quando este código for executado, o mesmo lançará uma NullPointerException, como pode ser visto na saída do console quando executamos tal programa.

```
Exception in thread "main" java.lang.NullPointerException
at aumenteaFrase.main(aumentaFrase.java:15)
```

Ou seja, o mesmo tentou acessar um atributo de um objeto que estava nulo. Para ajudar a melhorar a situação, deve-se usar o try/catch.

Programação orientada a objetos em Java
Tratamento de Exceções

Quando este código for executado, o mesmo lançará uma NullPointerException, como pode ser visto na saída do console quando executamos tal programa.

```
Exception in thread "main" java.lang.NullPointerException
at aumenteaFrase.main(aumentaFrase.java:15)
```

Ou seja, o mesmo tentou acessar um atributo de um objeto que estava nulo. Para ajudar a melhorar a situação, deve-se usar o try/catch.

Programação orientada a objetos em Java
Tratamento de Exceções

Exemplo pratico

public class TesteEnviaExcecao
public class TesteTrataExcecao

Programação orientada a objetos em Java
Tratamento de Exceções

Quando este código foi executado, o mesmo lançou uma NullPointerException, porém, desta vez esta exceção foi tratada, sendo a mesma capturada pelo catch{} e dentro deste bloco as devidas providências foram tomadas. Neste caso é atribuído um valor default à variável frase. A saída deste programa foi a Seguinte:

Frase antiga: Frase vazia

Frase nova: FRASE VAZIA

Programação orientada a objetos em Java
Tratamento de Exceções

Bloco finally

Temos a seguinte situação:

Foi aberta uma conexão com o banco de dados para realizar determinada ação, e no meio deste processo seja lançada alguma exceção, como por exemplo, NullPointerException ao tentar manipular um determinado atributo de um objeto. Neste caso seria necessário que mesmo sendo lançada uma exceção no meio do processo a conexão fosse fechada.

Programação orientada a objetos em Java
Tratamento de Exceções

Bloco finally

Outra situação:

A abertura de um determinado arquivo para escrita, e no meio deste processo é lançada uma exceção por algum motivo, o arquivo não seria fechado, o que resultaria em deixar o arquivo aberto.

Quando uma exceção é lançada e é necessário que determinada ação seja tomada, mesmo após a sua captura, utilizamos a palavra reservada finally.

Programação orientada a objetos em Java
Tratamento de Exceções

Sintaxe:

```
try
{
  //trecho de código que pode vir a lançar uma exceção
}
catch(tipo_exceçao_1 e)
{
  //ação a ser tomada
}
catch(tipo_exceçao_2 e)
{
  //ação a ser tomada
}
catch(tipo_exceçao _n e)
{
  //ação a ser tomada
}
finally
{
  //ação a ser tomada
}
```

Programação orientada a objetos em Java
Tratamento de Exceções

Bloco finally

Neste exemplo, mesmo o código lançando uma exceção durante a sua execução e a mesma sendo capturada pelo catch, uma determinada ação será tomada no bloco finally, neste caso tanto com a exceção ou não será executada a linha “ novaFrase = frase.toUpperCase(), tornando todas letras da frase maiúsculas. O bloco finally sempre será executado. A saída deste programa seria a seguinte:

A frase inicial está nula, para solucionar tal problema, foi lhe atribuito um valor default.
Frase antiga: Frase vazia
Frase nova: FRASE VAZIA

Programação orientada a objetos em Java
Tratamento de Exceções

O bloco finally é opcional e é colocado após o último bloco catch para tratar vazamento de recurso;

O bloco finally será executado nas seguintes situações:

- Se uma exceção for ou não lançada no bloco try correspondente;
- Se um bloco try fechar utilizando return, break ou continue.

O bloco finally não será executado se o aplicativo fechar antes de um bloco try chamando o método System.exit;

Programação orientada a objetos em Java
Tratamento de Exceções

Dica de prevenção de erro

O bloco finally é um lugar ideal para liberar os recursos adquiridos em um bloco try (como arquivos abertos), o que ajuda a eliminar vazamentos de recurso.

- Por exemplo, arquivos, conexões de banco de dados e conexões de rede que não são fechadas adequadamente depois que não são mais necessárias

Programação orientada a objetos em Java
Tratamento de Exceções

Instrução throws

Pense na situação em que não é desejado que uma exceção seja tratada na própria classe ou método, mas sim em outro que venha lhe chamar. Para solucionar tal situação utilizamos a instrução throws na assinatura do método com a possível exceção que o mesmo poderá a vir lançar.

Sintaxe:

```
tipo_retorno nome_metodo() throws tipo_exceção_1,
tipo_exceção_2, tipo_exceção_n {

...}
```

Programação orientada a objetos em Java
Tratamento de Exceções

Instrução throws

Onde:

tipo_retorno – Tipo de retorno do método.

nome_metodo() - Nome do método que será utilizado.

tipo_exceção_1 a tipo_exceção_n – Tipo de exceções separadas por virgula que o seu método pode vir a lançar.

Programação orientada a objetos em Java
Tratamento de Exceções

Instrução throws

Exemplo Pratica

public class TesteInstrucaoThrows

Neste exemplo será lançada uma exceção no método aumetarLetras():
E o mesmo será tratado no método main().

Programação orientada a objetos em Java
Tratamento de Exceções

Instrução throw

Nesse exemplo temos o caso em que seja necessário lançar uma exceção padrão ao invés de uma especifica. Para resolver este problema, utilizamos a instrução throw dentro do bloco catch que desejamos converter a exceção.

Sintaxe:

```
try
{
 //...
}
catch(tipoExcessão_1 e)
{
 throw new novoTipoExceçao(e);
}
```

Programação orientada a objetos em Java
Tratamento de Exceções

Instrução throw

Onde:

tipoExcessão_1 e – Tipo de exceção que pode ser capturada pelo bloco catch.

NovoTipoExceçao – Tipo de exceção que será lançada.

Programação orientada a objetos em Java
Tratamento de Exceções

Instrução throw

Exemplo Pratica

public class TesteInstrucaoThrow

Neste exemplo será lançada uma NullPointerException e a mesma será convertida para Exception e relançada como Exception no método aumentarLetras() e, por fim, a mesma é tratada no método main().

Programação orientada a objetos em Java
Tratamento de Exceções

Instrução throw

Outro Exemplo Pratica

public class Exemplo_Throw

Nesse exemplo é criada uma exceção especifica dentro do bloco if do método e em seguida lançada na instrução throw.

Programação orientada a objetos em Java
Tratamento de Exceções

Criando exceções

Assim como qualquer objeto, em Java também é possível criar suas próprias exceções. Vamos ao cenário em que nenhuma exceção existente faça sentido para ser lançada por você.

Por exemplo, imagine que por algum motivo você precisa que uma exceção seja lançada quando a letra B ou b não existe em determinada frase, como não existe nenhuma exceção específica para este caso será necessário criar uma exceção.

Programação orientada a objetos em Java
Tratamento de Exceções

Criando exceções

Criando uma exceção para ser lançada toda vez que uma letra B ou b não é encontrada em uma determinada frase.

```
 @Override
public String getMessage(){
        return "Não existe letra B em sua frase";
}
```

Toda exceção criada deve estender Exception, neste exemplo foi sobrescrito o método getMessage(), que é exibida no prompt toda vez que a exceção é lançada.

Programação orientada a objetos em Java
Tratamento de Exceções

Criando exceções

Exemplo pratica

public class TesteCriandoExcecao

Quando o programa acima for executado, uma exceção do tipo SemLetraBException() seria lançada.

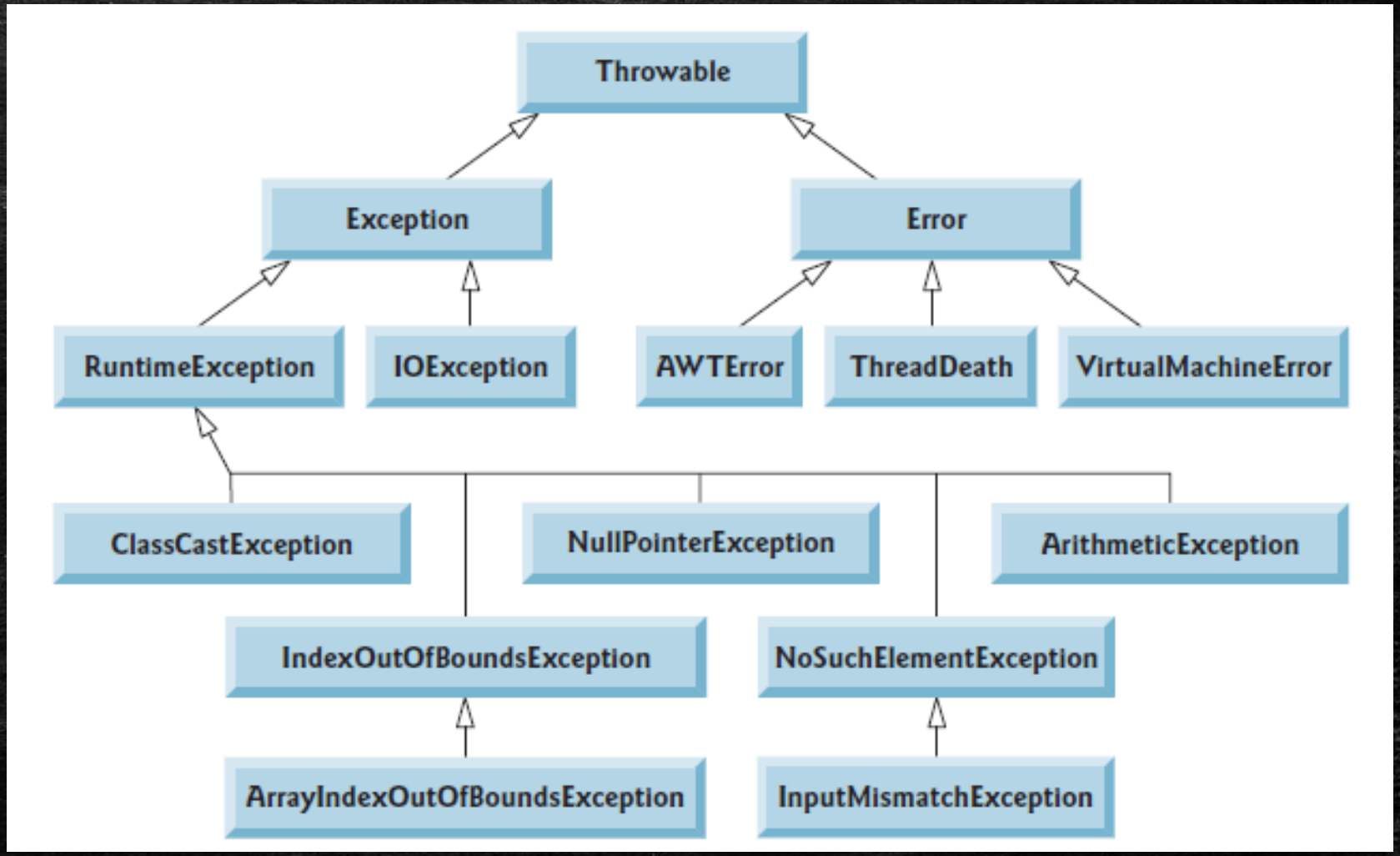
Programação orientada a objetos em Java
Tratamento de Exceções

Conclusão

O tratamento de exceções em Java é bem simples e sua manipulação pode ser feita de acordo com o que o programador deseja, desde tratá-la no próprio método ou lança-la para ser tratada em um método que venha chamar o método que lança a mesma. Além de poder criar suas próprias exceções.

Programação orientada a objetos em Java
Tratamento de Exceções

Hierarquia de exceção Java



Programação orientada a objetos em Java
Tratamento de Exceções

Exercício