Tratamento de exceções

Emerson C. Lima

Programação Java

Objetivos dessa lição

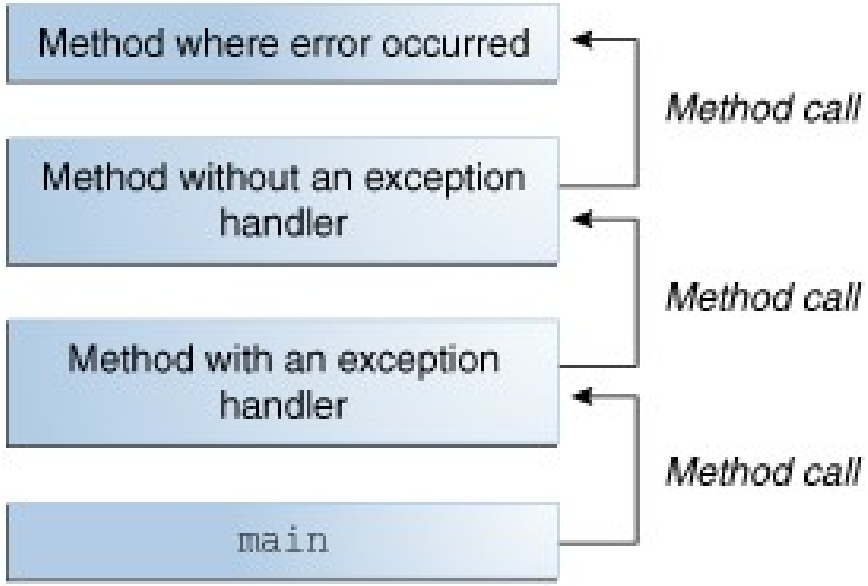
- Saber o que são exceções
- Capturar e tratar exceções
- Lançar exeções
- O comando try com recursos

Conteúdo

- Visão geral
- Exemplo I
- Exemplo II
- Hierarquia de exceção Java
- Bloco finally
- Desempilhamento de pilha

O que é uma Exceção

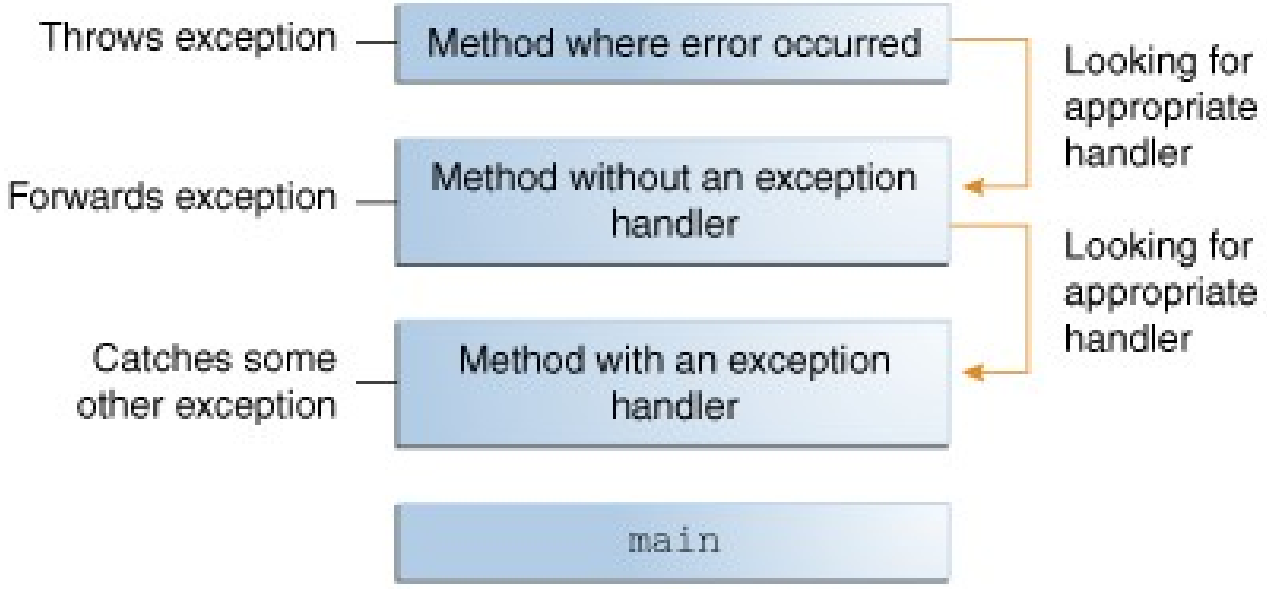
- Uma Exceção é um evento que ocorre durante a execução de um programa que interrompe o fluxo normal de instruções.



The call stack.

O que é uma Exceção

- Uma Exceção é um evento que ocorre durante a execução de um programa que interrompe o fluxo normal de instruções.



Searching the call stack for the exception handler.

O Requisito “Capture ou Especifique”

- Um código que pode lançar exceções deve ser:

O Requisito “Capture ou Especifique”

- Um código que pode lançar exceções deve ser:

O Requisito “Capture ou Especifique”

- Um código que pode lançar exceções deve ser:
  - Protegido em um comando try que captura a exceção

O Requisito “Capture ou Especifique”

- Um código que pode lançar exceções deve ser:
  - Protegido em um comando try que captura a exceção
  - Protegido em um método que especifica que lança a exceção

O Requisito “Capture ou Especifique”

- Um código que pode lançar exceções deve ser:
  - Protegido em um comando try que captura a exceção
  - Protegido em um método que especifica que lança a exceção

Pessoa.java

```
1  import java.io.File;
2  import java.util.Scanner;
3
4  public class Pessoa {
5
6      private String nome;
7      private String cpf;
8      private int idade;
9
10     // métodos gets e sets
11
12     public void carregar() {
13         File arquivo = new File("pessoa.txt");
14         Scanner sc = new Scanner(arquivo);
15         this.cpf = sc.next();
16         this.idade = sc.nextInt();
17         this.nome = sc.nextLine();
18         sc.close();
19     }
```

O Requisito “Capture ou Especifique”

- Um código que pode lançar exceções deve ser:
  - Protegido em um comando try que captura a exceção
  - Protegido em um método que especifica que lança a exceção

Pessoa.java

```
1  import java.io.File;
2  import java.io.FileNotFoundException;
3  import java.util.Scanner;
4
5  public class Pessoa {
6
7      private String nome;
8      private String cpf;
9      private int idade;
10
11     // métodos gets e sets
12
13     public void carregar() {
14         File arquivo = new File("pessoa.txt");
15         Scanner sc = null;
16         try {
17             sc = new Scanner(arquivo);
18         } catch (FileNotFoundException e) {
19             System.out.println("Arquivo nao existe!");
20         }
21         this.cpf = sc.next();
22         this.idade = sc.nextInt();
23         this.nome = sc.nextLine();
24         sc.close();
25     }
```

O Requisito “Capture ou Especifique”

- Um código que pode lançar exceções deve ser:
  - Protegido em um comando try que captura a exceção
  - Protegido em um método que especifica que lança a exceção

Pessoa.java

```
1  import java.io.File;
2  import java.io.FileNotFoundException;
3  import java.util.Scanner;
4
5  public class Pessoa {
6
7      private String nome;
8      private String cpf;
9      private int idade;
10
11     // métodos gets e sets
12
13     public void carregar() {
14         File arquivo = new File("pessoa.txt");
15         try {
16             Scanner sc = new Scanner(arquivo);
17             this.cpf = sc.next();
18             this.idade = sc.nextInt();
19             this.nome = sc.nextLine();
20             sc.close();
21         } catch (FileNotFoundException e) {
22             System.out.println("Arquivo nao existe!");
23         }
24     }
```

O Requisito “Capture ou Especifique”

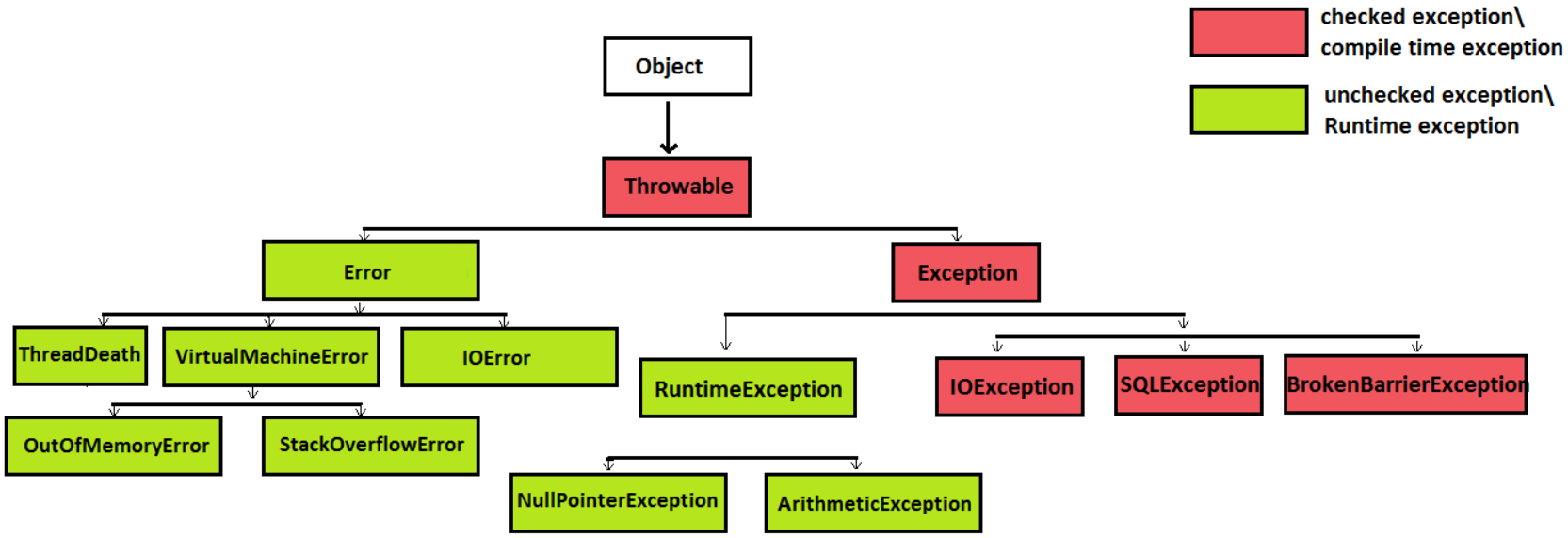
- Um código que pode lançar exceções deve ser:
  - Protegido em um comando try que captura a exceção
  - Protegido em um método que especifica que lança a exceção

Pessoa.java

```
1  import java.io.File;
2  import java.io.FileNotFoundException;
3  import java.util.Scanner;
4
5  public class Pessoa {
6
7      private String nome;
8      private String cpf;
9      private int idade;
10
11     // métodos gets e sets
12
13     public void carregar() throws FileNotFoundException {
14         File arquivo = new File("pessoa.txt");
15         Scanner sc = new Scanner(arquivo);
16         this.cpf = sc.next();
17         this.idade = sc.nextInt();
18         this.nome = sc.nextLine();
19         sc.close();
20     }
```

Os três tipos de Exceções

- Erro
- Exceções verificadas
- Exceções de tempo de execução



Os três tipos de Exceções

Pessoa.java

```java
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class Pessoa {

    private String nome;
    private String cpf;
    private int idade;

    @Override
    public String toString() {
        return "Pessoa [nome=" + nome + ", cpf=" + cpf + ", idade=" + idade + "]";
    }

    public void carregar() {
        File arquivo = new File("pessoa.txt");
        try {
            Scanner sc = new Scanner(arquivo);
            this.cpf = sc.next();
            this.idade = sc.nextInt();
            this.nome = sc.nextLine();
            sc.close();
        } catch (FileNotFoundException e) {
            System.out.println("Ops! Arquivo não encontrado!");
        }
    }

    public static void main(String[] args) throws FileNotFoundException {
        Pessoa p = new Pessoa();
        p.carregar();
        System.out.println(p);
    }
}
```

Exemplo

Pessoa.java

```java
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class Pessoa {

    private String nome;
    private String cpf;
    private int idade;

    @Override
    public String toString() {
        return "Pessoa [nome=" + nome + ", cpf=" + cpf + ", idade=" + idade + "]";
    }

    public void carregar() {
        File arquivo = new File("pessoa.txt");
        try {
            Scanner sc = new Scanner(arquivo);
            this.cpf = sc.next();
            this.idade = sc.nextInt();
            this.nome = sc.nextLine();
            sc.close();
        } catch (FileNotFoundException e) {
            System.out.println("Ops! Arquivo não encontrado!");
        }
    }

    public static void main(String[] args) throws FileNotFoundException {
        Pessoa p = new Pessoa();
        p.carregar();
        System.out.println(p);
    }
}
```

Exemplo

Pessoa.java

```java
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class Pessoa {

    private String nome;
    private String cpf;
    private int idade;

    @Override
    public String toString() {
        return "Pessoa [nome=" + nome + ", cpf=" + cpf + ", idade=" + idade + "]";
    }

    public void carregar() {
        File arquivo = new File("pessoa.txt");
        try {
            Scanner sc = new Scanner(arquivo);
            this.cpf = sc.next();
            this.idade = sc.nextInt();
            this.nome = sc.nextLine();
            sc.close();
        } catch (FileNotFoundException e) {
            System.out.println("Ops! Arquivo não encontrado!");
        }
    }

    public static void main(String[] args) throws FileNotFoundException {
        Pessoa p = new Pessoa();
        p.carregar();
        System.out.println(p);
    }
}
```

Exemplo

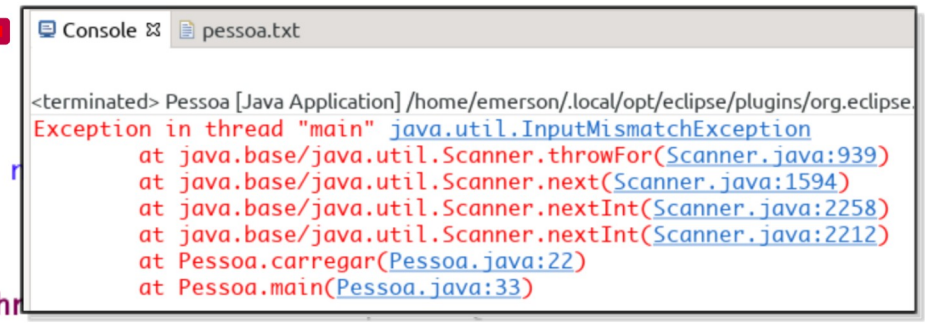
Pessoa.java

```java
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class Pessoa {

    private String nome;
    private String cpf;
    private int idade;

    @Override
    public String toString() {
        return "Pessoa [nome=" + nome + ", cpf=" + cpf + ", idade=" + idade + "]";
    }

    public void carregar() {
        File arquivo = new File("pessoa.txt");
        try {
            Scanner sc = new Scanner(arquivo);
            this.cpf = sc.next();
            this.idade = sc.nextInt();
            this.nome = sc.nextLine();
            sc.close();
        } catch (FileNotFoundException e) {
            System.out.println("Ops! Arquivo não encontrado!");
        }
    }

    public static void main(String[] args) throws FileNotFoundException {
        Pessoa p = new Pessoa();
        p.carregar();
        System.out.println(p);
    }
}
```

Exemplo

Pessoa.java

```java
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class Pessoa {

    private String nome;
    private String cpf;
    private int idade;

    @Override
    public String toString() {
        return "Pessoa [nome=" + nome + ", cpf=" + cpf + ", idade=" + idade + "]";
    }

    public void carregar() {
        File arquivo = new File("pessoa.txt");
        try {
            Scanner sc = new Scanner(arquivo);
            this.cpf = sc.next();
            this.idade = sc.nextInt();
            this.nome = sc.nextLine();
            sc.close();
        } catch (FileNotFoundException e) {
            System.out.println("Ops! Arquivo n
        }
    }

    public static void main(String[] args) thr
        Pessoa p = new Pessoa();
        p.carregar();
        System.out.println(p);
    }
}
```



Exemplo

Pessoa.java

```java
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class Pessoa {

    private String nome;
    private String cpf;
    private int idade;

    @Override
    public String toString() {
        return "Pessoa [nome=" + nome + ", cpf=" + cpf + ", idade=" + idade + "]";
    }

    public void carregar() {
        File arquivo = new File("pessoa.txt");
        try {
            Scanner sc = new Scanner(arquivo);
            this.cpf = sc.next();
            this.idade = sc.nextInt();
            this.nome = sc.nextLine();
            sc.close();
        } catch (FileNotFoundException e) {
            System.out.println("Ops! Arquivo não encontrado!");
        }
    }

    public static void main(String[] args) throws FileNotFoundException {
        Pessoa p = new Pessoa();
        p.carregar();
        System.out.println(p);
    }
}
```

Fechando recursos

```
16      public void carregar() {
17          File arquivo = new File("pessoa.txt");
18          Scanner sc = null;
19          try {
20              sc = new Scanner(arquivo);
21              this.cpf = sc.next();
22              this.idade = sc.nextInt();
23              this.nome = sc.nextLine();
24          } catch (FileNotFoundException e) {
25              System.out.println("Ops! Arquivo não encontrado!");
26          } finally {
27              sc.close();
28          }
29      }
```

Fechando recursos

```
16    public void carregar() {
17        File arquivo = new File("pessoa.txt");
18        try (Scanner sc = new Scanner(arquivo)) {
19            this.cpf = sc.next();
20            this.idade = sc.nextInt();
21            this.nome = sc.nextLine();
22        } catch (FileNotFoundException e) {
23            System.out.println("Ops! Arquivo não encontrado!");
24        }
25    }
```