

PRINCIPAIS PILARES DA LINGUAGEM DE PROGRAMAÇÃO JAVA



Sumário

1. Introdução ao Java.....	3
O que é Java?.....	3
Características Principais.....	3
2. Ambiente de Desenvolvimento.....	4
Configuração do Ambiente.....	4
3. Sintaxe Básica.....	5
Estrutura de um Programa Java.....	5
Tipos de Dados.....	5
Operadores.....	5
4. Estruturas de Controle.....	6
Condicionais.....	6
Laços.....	7
5. Classes e Objetos.....	9
Definição de Classes.....	9
Instanciação de Objetos.....	9
Construtores.....	9
6. Herança e Polimorfismo.....	10
Herança.....	10
Polimorfismo.....	10
7. Exceções e Tratamento de Erros.....	11
Bloco try-catch.....	11
Bloco finally.....	11
Criação de Exceções Personalizadas.....	12
8. Coleções e Generics.....	12
Listas.....	12
Mapas.....	12
Generics.....	13
9. Streams e Expressões Lambda.....	13
Streams.....	13
Expressões Lambda.....	13
10. Entrada e Saída (I/O).....	14
Leitura de Arquivos.....	14
Escrita de Arquivos.....	14
Threads e Concorrência.....	15
Comunicação em Rede.....	16
12. Conclusão.....	17

1. Introdução ao Java

O que é Java?

Java é uma linguagem de programação orientada a objetos, criada pela Sun Microsystems (agora parte da Oracle) em 1995. É amplamente utilizada para desenvolver aplicações web, móveis, de desktop e empresariais.

Características Principais

- Portabilidade: Código Java pode ser executado em qualquer dispositivo que tenha uma Máquina Virtual Java (JVM).
- Orientação a Objetos: Java é baseado em conceitos de objetos e classes.
- Segurança: Java tem várias características que garantem a execução segura do código.
- Robustez: Java minimiza erros de programação comuns graças à sua gestão de memória automática.


2. Ambiente de Desenvolvimento

Configuração do Ambiente

1. **Instalação do JDK (Java Development Kit):** Inclui a JVM, compilador e outras ferramentas.
 - **Download:** Baixe o JDK do site da Oracle.
 - **Instalação:** Siga as instruções de instalação para seu sistema operacional.
2. **IDE (Integrated Development Environment):** Ferramenta para facilitar o desenvolvimento.
 - **Eclipse, IntelliJ IDEA, NetBeans:** Escolha uma IDE popular e configure-a.

Primeiro Programa Java

java

 Copiar código

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

- **Compilação:** javac HelloWorld.java
- **Execução:** java HelloWorld

3. Sintaxe Básica

Estrutura de um Programa Java

- **Classes:** A base de qualquer programa Java.
- **Métodos:** Blocos de código executáveis.
- **Variáveis:** Armazenam dados.

Tipos de Dados

- **Primitivos:** int, double, char, boolean, etc.
- **Referência:** Objetos e arrays.

Operadores


- **Aritméticos:** +, -, *, /, %
- **Relacionais:** ==, !=, >, <, >=, <=
- **Lógicos:** &&, ||, !

4. Estruturas de Controle

Condicionais

- if, else if, else:


java

 Copiar código

```
if (condition) {  
    // código  
} else if (otherCondition) {  
    // código  
} else {  
    // código  
}
```

- switch:

java


 Copiar código

```
switch (variable) {  
    case value1:  
        // código  
        break;  
    case value2:  
        // código  
        break;  
    default:  
        // código  
}
```

Laços

- for:


java

 Copiar código

```
for (int i = 0; i < 10; i++) {  
    // código  
}
```

- **while:**


java

 Copiar código

```
while (condition) {  
    // código  
}
```

- **do-while:**

java


 Copiar código

```
do {  
    // código  
} while (condition);
```

5. Classes e Objetos

Definição de Classes


java

 Copiar código

```
public class MyClass {  
    // atributos  
    int myAttribute;  
  
    // métodos  
    void myMethod() {  
        // código  
    }  
}
```

Instanciação de Objetos


java

 Copiar código

```
MyClass obj = new MyClass();  
obj.myMethod();
```

Construtores

java


 Copiar código

```
public class MyClass {  
    int myAttribute;  
  
    // Construtor  
    public MyClass(int value) {  
        myAttribute = value;  
    }  
}
```


6. Herança e Polimorfismo

Herança


java

 Copiar código

```
public class Animal {  
    void makeSound() {  
        System.out.println("Animal sound");  
    }  
}  
  
public class Dog extends Animal {  
    void makeSound() {  
        System.out.println("Bark");  
    }  
}
```

Polimorfismo

java


 Copiar código

```
Animal myAnimal = new Dog();  
myAnimal.makeSound(); // Output: Bark
```

7. Exceções e Tratamento de Erros

Bloco try-catch


java

 Copiar código

```
try {  
    // código que pode lançar uma exceção  
} catch (ExceptionType e) {  
    // código de tratamento de exceção  
}
```

Bloco finally


java

 Copiar código

```
try {  
    // código  
} catch (ExceptionType e) {  
    // código de tratamento de exceção  
} finally {  
    // código que sempre é executado  
}
```

Criação de Exceções Personalizadas

java


 Copiar código

```
public class MyException extends Exception {  
    public MyException(String message) {  
        super(message);  
    }  
}
```

8. Coleções e Generics

Listas

java


 Copiar código

```
import java.util.ArrayList;
import java.util.List;

List<String> myList = new ArrayList<>();
myList.add("Item 1");
```

Mapas

java


 Copiar código

```
import java.util.HashMap;
import java.util.Map;

Map<String, Integer> myMap = new HashMap<>();
myMap.put("key1", 100);
```

Generics

java

 Copiar código


```
public class Box<T> {
    private T t;

    public void set(T t) { this.t = t; }
    public T get() { return t; }
}
```

9. Streams e Expressões Lambda

Streams

java


 Copiar código

```
import java.util.Arrays;
import java.util.List;

List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 5);
numbers.stream().filter(n -> n % 2 == 0).forEach(System.out::println);
```

Expressões Lambda

java


 Copiar código

```
List<String> names = Arrays.asList("Anna", "Bob", "Charlie");
names.forEach(name -> System.out.println(name));
```

10. Entrada e Saída (I/O)

Leitura de Arquivos

java


 Copiar código

```
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.List;

List<String> lines = Files.readAllLines(Paths.get("file.txt"));
```

Escrita de Arquivos

java

 Copiar código


```
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.List;
import java.util.Arrays;

List<String> lines = Arrays.asList("Line 1", "Line 2");
Files.write(Paths.get("file.txt"), lines);
```

11. Tópicos Avançados

Threads e Concorrência


java

 Copiar código

```
public class MyThread extends Thread {  
    public void run() {  
        System.out.println("Thread running");  
    }  
}  
  
MyThread t = new MyThread();  
t.start();
```

Comunicação em Rede

java

 Copiar código

```
import java.io.*;  
import java.net.*;  
  
public class MyClient {  
    public static void main(String[] args) throws IOException {  
        Socket socket = new Socket("localhost", 8080);  
        PrintWriter out = new PrintWriter(socket.getOutputStream(), true);  
        out.println("Hello, server");  
    }  
}
```

12. Conclusão

Neste ebook, exploramos os principais pilares da linguagem de programação Java, desde a configuração do ambiente de desenvolvimento até tópicos avançados como threads e comunicação em rede. Java é uma linguagem poderosa e versátil, ideal para uma variedade de aplicações. Com prática e estudo contínuo, você estará bem equipado para criar aplicações robustas e eficientes.

Totalmente criado por IA, nos mostra quão benéfica pode ser nossa relação com a tecnologia.