

Tipos primitivos de dados

Tipos primitivos de dados em Java

Em programação, um tipo de dado é um atributo que determina o tipo de valor que uma variável pode armazenar. Em Java, existem oito tipos primitivos de dados, que são:

1. **int**: para armazenar números inteiros, como -10, 0, 100, etc.
2. **double**: para armazenar números decimais, como 3.14, -0.5, 2.0, etc.
3. **float**: para armazenar números decimais menores que o tipo double, como 3.14f, -0.5f, 2.0f, etc.
4. **boolean**: para armazenar valores booleanos, que são verdadeiros ou falsos.
5. **char**: para armazenar um único caractere, como 'a', 'B', '&', etc.
6. **byte**: para armazenar números inteiros pequenos, como -128 até 127.
7. **short**: para armazenar números inteiros maiores que byte, como -32768 até 32767.
8. **long**: para armazenar números inteiros grandes, como -9223372036854775808 até 9223372036854775807.

Esses tipos de dados são considerados primitivos porque não são objetos. Eles são armazenados diretamente na memória e têm uma representação binária simples.

Além desses tipos de dados, Java também tem tipos de dados de referência, que são objetos criados a partir de classes. Esses tipos de dados incluem String, Array e outros tipos definidos pelo usuário.

Em resumo, entender os tipos primitivos de dados em Java é fundamental para a programação em Java. Eles são a base de dados para muitas aplicações em Java.

Conversão de tipo de dados

A conversão dos tipos primitivos de dados em Java pode ser feita por meio de castings ou métodos de conversão. Os castings são usados para converter um tipo

de dado em outro tipo compatível. Por exemplo, para converter um `int` em um `double`, pode-se fazer:

```
int numInt = 10;
double numDouble = (double) numInt;
```

Já os métodos de conversão são usados para converter um tipo de dado em outro tipo incompatível. Por exemplo, para converter uma `String` em um `int`, pode-se fazer:

```
String numString = "10";
int numInt = Integer.parseInt(numString);
```

Existem métodos de conversão para todos os tipos de dados primitivos em Java. É importante lembrar que, ao fazer uma conversão de tipo de dado, pode ocorrer perda de precisão ou estouro de limite. Portanto, é necessário tomar cuidado ao fazer essas conversões.

Visibilidades e Modificadores de Acesso em Java

Em Java, a visibilidade é usada para determinar quais partes do código podem acessar determinada variável, método ou classe. Existem quatro tipos de visibilidade em Java: `public`, `protected`, `default` e `private`.

1. **public:** um membro público pode ser acessado por qualquer classe em qualquer pacote.
2. **protected:** um membro protegido pode ser acessado apenas por classes do mesmo pacote ou por classes que estendem a classe que contém o membro protegido.
3. **default:** a visibilidade padrão (sem especificação de visibilidade) é usada quando nenhum modificador de acesso é especificado. Um membro com visibilidade padrão pode ser acessado apenas por classes do mesmo pacote.
4. **private:** um membro privado pode ser acessado apenas pela classe que o contém.

Além disso, Java também tem modificadores de acesso, que são usados para modificar o comportamento padrão da visibilidade. Existem três tipos de modificadores de acesso em Java: final, static e abstract.

1. **final:** quando um membro é declarado como final, ele não pode ser alterado depois que é inicializado.
2. **static:** um membro estático pertence à classe em vez de pertencer a cada objeto individual da classe.
3. **abstract:** uma classe ou método abstrato é declarado sem implementação e precisa ser implementado por uma classe filha.

Em resumo, entender as visibilidades e modificadores de acesso em Java é fundamental para o desenvolvimento de aplicativos Java. Eles permitem que os desenvolvedores controlem o acesso e o comportamento de membros e classes em um aplicativo Java.