

## Variável do tipo INT (inteiro) pode Guardar Ponto Flutuante (valores quebrados)?

Na aula passada tínhamos uma variável do tipo int chamada idade.

Se tentarmos colocar no lugar da idade, um valor que não seja inteiro, como 37.5, a compilação não irá ocorrer.

Vamos por isso em pratica:

- vamos criar uma classe com nome TestePontoFlutuantes
- em seguida criar uma variável do tipo int com nome de salario e recebendo um valor não inteiro.

```
1
2 public class TestePontoFlutuante {
3     public static void main(String[] args) {
4
5         int salario = 3500.5;
6
7     }
8 }
9
```

Type mismatch: cannot convert from double to int  
2 quick fixes available:  
+ Add cast to 'int'  
→ Change type of 'salario' to 'double'  
Press 'F2' for focus

OBS: Lembrando que no Java usa o (ponto '.') para separar o decimal no código fonte.

O **erro** que se lê ao passar o mouse em cima: "**Type mismatch: cannot convert from double to int**", isto é, a conversão não é possível.

Não podemos colocar ponto flutuante (número quebrado) em variáveis do tipo inteira.

Temos dois tipos muito usados de ponto flutuantes que são: double e float, sendo que double é mais usado que o float.

## Como Guardar Valores Não Inteiros?

Deve ser guardado em variáveis de tipos de ponto flutuantes: como double ou float por exemplo.

```
TestePontoFlutuante.java ×
1
2 public class TestePontoFlutuante {
3     public static void main(String[] args) {
4
5         double salario; // declarando variavel
6         salario = 3500.5; // atribuindo valor quebrado
7
8         //imprimindo um texto mais o valor da variavel salario:
9         System.out.println("O salário é: R$ " + salario);
10
11     }
12 }
13
```

## Pode Guardar um int em um double diretamente?

**SIM.** O tipo `double` em Java é um tipo de ponto flutuante de precisão dupla que pode representar números inteiros e decimais com uma ampla faixa de valores. Como o `double` tem uma maior faixa de representação em comparação com o tipo `int` (inteiro), valores inteiros podem ser armazenados em variáveis do tipo `double`.

EXEMPLO:

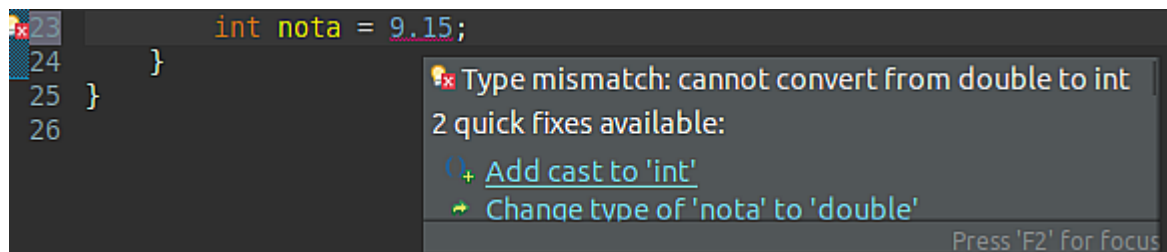
```
double idade = 30;
```

Quando atribui um valor **inteiro** a uma variável do tipo **double**, é feita uma **conversão automática** pelo Java. Ou seja no exemplo dado o valor 30 é automaticamente convertido para um tipo `double`. Após isso a variável `idade` contém o valor **30.0** do tipo `double`. Mesmo que tenha atribuído um valor inteiro, o Java trata automaticamente a conversão para o tipo de ponto flutuante.

## Pode Guardar um double em um int diretamente?

**NÃO.** Não é possível devido à possibilidade de perda de precisão. O tipo `double` é de ponto flutuante e pode representar números com decimais e uma maior faixa de valores do que o tipo `int`, que é um tipo inteiro.

Quando tenta atribuir um valor `double` a uma variável `int`, o compilador Java gera um erro porque pode haver perda de informação. **Por exemplo:**



Podemos ver que é dada soluções rápidas que podem ajudar, como adicionar conversão para `int` (CAST). Isso seria um CAST explícito, diferente do cast implícito que é feito de forma automática pelo Java. VEREMOS ISSO EM AULAS FUTURAS.

O importante é saber que de forma DIRETA NÃO pode guardar `double` em `int`.

## Brincando com o double e int:

Na seguinte divisão:

```
double divisao = 3.14 / 2;  
System.out.println(divisao);
```

teremos:

```
1.57
```

## Divisão entre Números INTEIROS:

Se for feito isso:

```
int novaDivisao = 5 / 2;
```

Quando você realiza a operação `int divisao = 5 / 2;` em Java, o resultado é 2, não 2.5. Isso ocorre porque a divisão de dois números inteiros em Java resulta em um valor inteiro, e a parte decimal é truncada (descartada, ignorada). Por isso nem apresenta erro de compilação, lembra que vimos que não podemos guardar tipo decimal(double ou float) em tipo inteiro, mas quando divide 2 números inteiros o resultado sempre será inteiro, mesmo o resultado correto sendo 2.5 será 2.

2

Quando você divide dois números inteiros em Java, o resultado é sempre um inteiro. Mesmo que o resultado da divisão seja um número fracionário, a parte decimal é descartada. Isso é conhecido como "divisão de inteiros".

### E se realizar a divisão de dois inteiros e guardar em um tipo double?

```
double outraDivisao = 5 / 2;  
System.out.println(outraDivisao);
```

Neste caso,  $5/2$  é uma operação de divisão **entre inteiros**, e o resultado é 2 (a parte decimal é truncada). Mesmo que esteja atribuindo esse resultado a uma variável **double**, a conversão para **double** ocorre depois que a operação já foi avaliada como uma divisão de inteiros (o Java primeiro olha pro lado direito, ele vê que tem 5 e vê que tem 2 aí pensa opa é uma divisão entre inteiros então vai da inteiro). Portanto, `outraDivisao` terá o valor 2.0, mas ainda será uma representação do número inteiro 2.

### Solução:

Para obter um resultado com parte decimal, pelo menos um dos operandos precisa ser do tipo de ponto flutuante como double. Você pode fazer isso explicitamente, por exemplo:

```
double divisaoFinal = 5.0 / 2;  
System.out.println(divisaoFinal);
```

Vai imprimir 2.5: 2.5

Digitando **5.0** ao invés de **5** está garantindo que pelo menos um dos operandos seja um tipo de ponto flutuante, e a divisão será tratada como uma operação de ponto flutuante, resultando em 2.5.