



02. Guardando qualquer referência

📅 Date	@04/10/2022
▼ Categoria	Java
▼ Curso	Java e java.util: Coleções Wrappers e Lambda expressions

Tópicos

- Array do tipo Object
- Array de clientes
- Cast explícito e implícito
- Cast possível e impossível
- Qual declaração?
- Quais casts
- Sobre ClassCastException
- Entendendo o array String args
- Mão na massa: Arrays
- O que aprendemos?

Cast explícito e implícito

Já falamos bastante sobre o *Type Cast* que é nada mais do que a conversão de um tipo para outro.

Cast implícito e explícito de primitivos

Para ser correto, já vimos o cast acontecendo antes mesmo de defini-lo. Temos dois exemplos, o primeiro do mundo de primitivos:

```
int numero = 3;
double valor = numero; //cast implícito
```

Repare que colocamos um valor da variável `numero` (tipo `int`) na variável `valor` (tipo `double`), sem usar um cast explícito. Isso funciona? A resposta é sim, pois qualquer inteiro cabe dentro de um double. Por isso o compilador fica quieto e não exige um *cast explícito*, mas nada impede de escrever:

```
int numero = 3;
double valor = (double) numero; //cast explícito
```

Agora, o contrário não funciona sem cast, uma vez que um `double` não cabe em um `int`:

```
double valor = 3.56;
int numero = (int) valor; //cast explícito é exigido pelo compilador
```

Nesse caso o compilador joga todo valor fracional fora e guarda apenas o valor inteiro.

Cast implícito e explícito de referências

Nas referências, o mesmo princípio se aplica. Se o cast sempre funciona não é necessário deixá-lo explícito, por exemplo:

```
ContaCorrente cc1 = new ContaCorrente(22, 33);
Conta conta = cc1; //cast implicito
```

Aqui também poderia ser explícito, mas novamente, o compilador não exige pois qualquer `ContaCorrente` é uma `Conta`:

```
ContaCorrente cc1 = new ContaCorrente(22, 33);
Conta conta = (Conta) cc1; //cast explícito mas desnecessário
```

Cast possível e impossível

Type cast explícito sempre funciona?

A resposta é não. O cast explícito só funciona se ele for *possível*, mas há casos em que o compilador sabe que um cast é impossível e aí nem compila, nem com *type cast*. Por exemplo:

```
Cliente cliente = new Cliente();  
Conta conta = (Conta) cliente; //impossível, não compila
```

Como o cliente não estende a classe `Conta` ou implementa uma interface do tipo `Conta`, é impossível esse cast funcionar, pois uma referência do tipo `Conta` jamais pode apontar para um objeto do tipo `Cliente`.

A certificação Java tem muitas dessas perguntas sobre cast possível, impossível, explícito e implícito. Se você pretende tirar essa certificação, vale a pena estudar esse assunto com muita calma.

O que aprendemos?

Nessa aula aprendemos:

- uma array do tipo `Object` pode guardar qualquer tipo de referência
- quando convertemos uma referência genérica para uma referência mais específica é preciso usar um *type cast*
- o cast só compila quando é possível, mesmo assim pode falhar na hora de rodar
- quando o *type cast* falha podemos receber uma `ClassCastException`

- para receber valores ao chamar o programa Java na linha de comando podemos usar o array `String[]` no método main