Java Academy 2016

Webinar #4

Softblue

Exceções: Trate os Erros da Forma Correta

Detecção ~~Problemática~~ de Erros

Softblue

```
boolean sucesso = o.processar();

if(sucesso) {
    //faça algo em caso de sucesso
} else {
    //faça algo em caso de falha
}
```

Booleanos

```
int resultado = o.processar();

if(resultado == 100) {
    // sucesso
} else if (resultado == 110) {
    // falha na validação
} else if (resultado == 120) {
    // falha de gravação no arquivo
}
```

"Magic Numbers"

JAVA ACADEMY

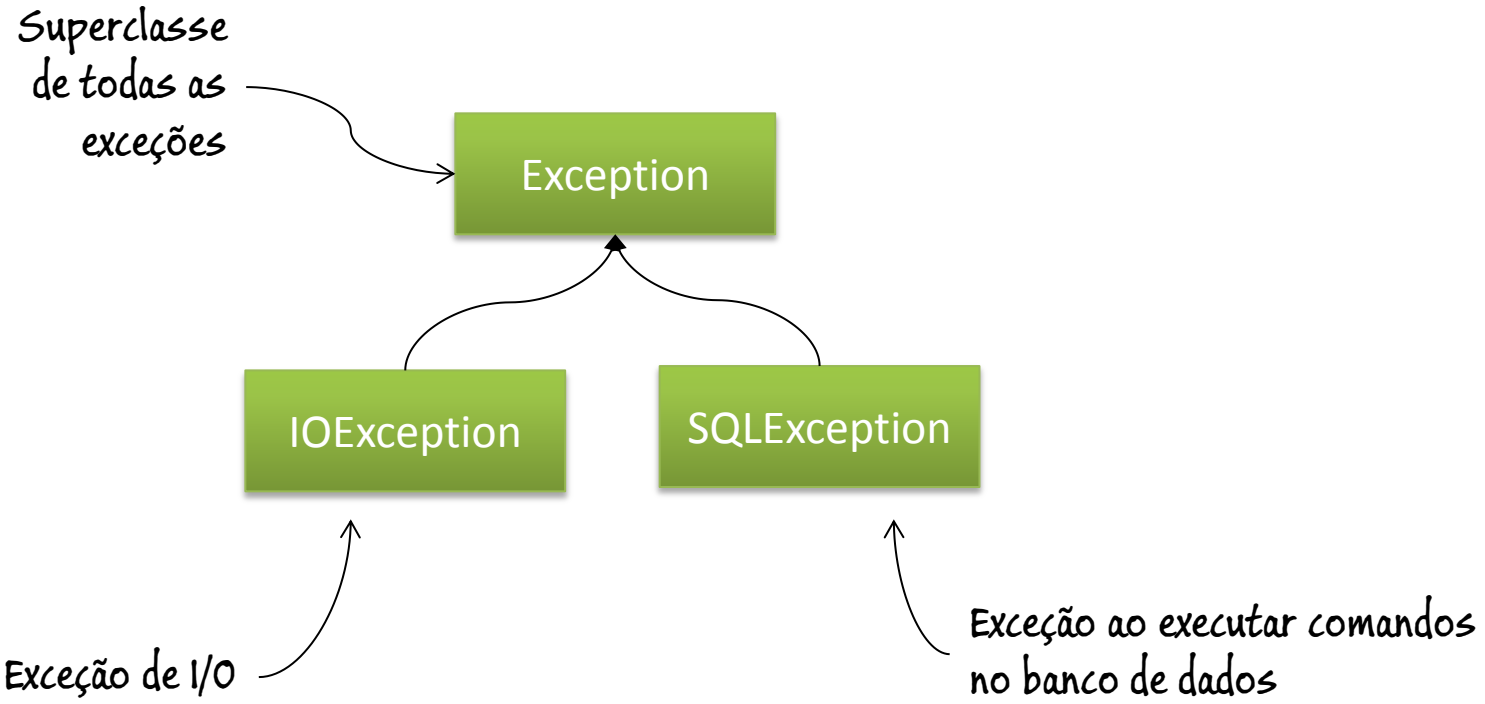
Exceções

- Exceções representam algo estranho ao sistema, que normalmente não ocorre
- Em Java, o tratamento de exceções é feito por um código diferente do código executado quando não ocorre a exceção

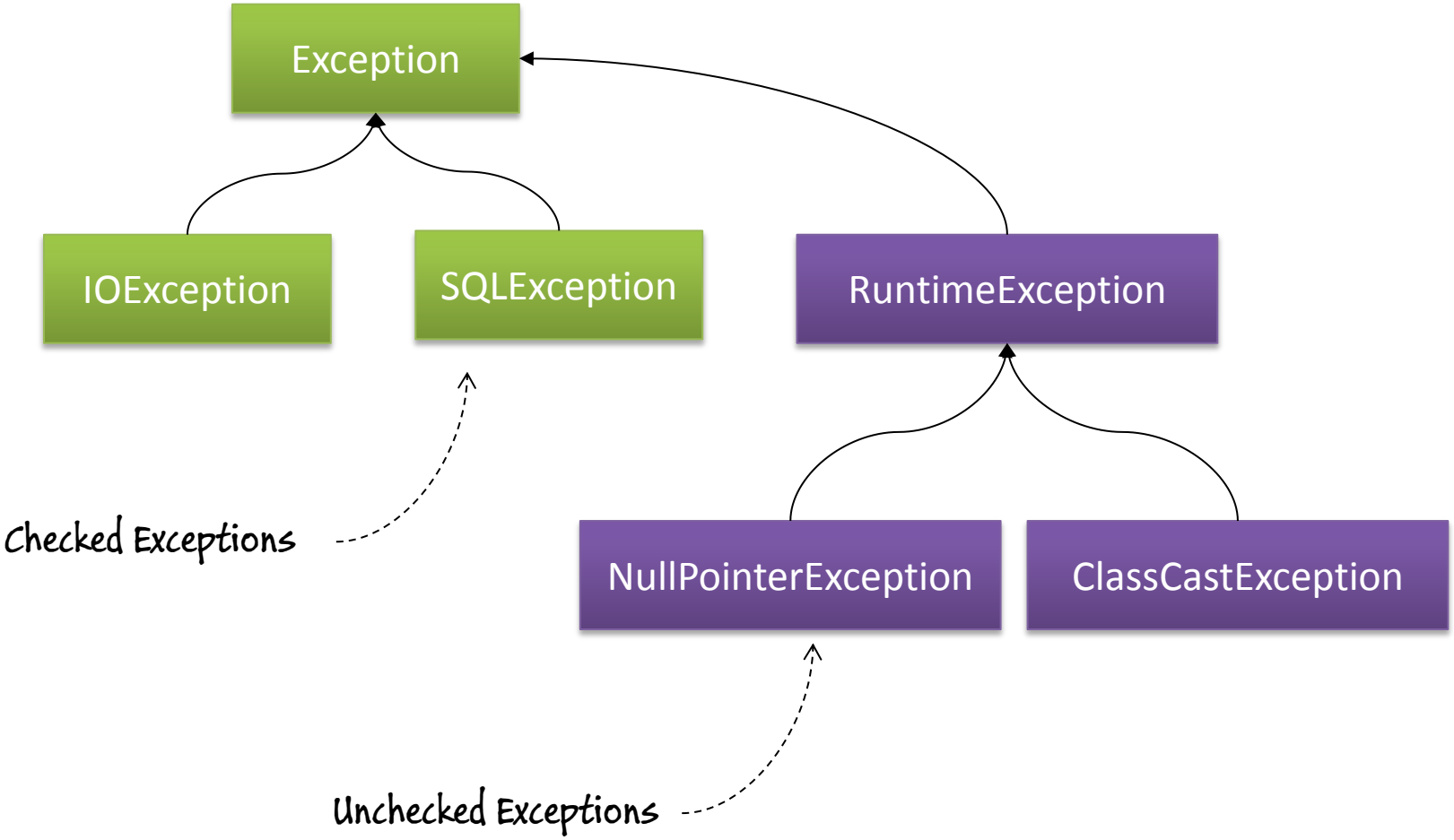
Classes Que Representam Exceções

- Exceções são representadas por classes
  - Devem herdar direta ou indiretamente de `Exception`
- O Java tem classes que representam diversos tipos de exceção, mas o programador pode criar exceções específicas de acordo com a necessidade

Exemplos de Exceções



Checked/Unchecked Exceptions



Lançando Exceções

Softblue

`throw` → Lança uma exceção para quem chamou o método

`throws` → Indica que um método pode lançar uma exceção

JAVA ACADEMY

Tratando Exceções

Softblue

`try` → Tenta executar um código que pode gerar exceção

`catch` → Trata a exceção que aconteceu

```
try {
    // Fluxo normal

} catch (Exception e) {
    // Fluxo de exceção
}
```

JAVA ACADEMY

Lançando Unchecked Exceptions

Softblue

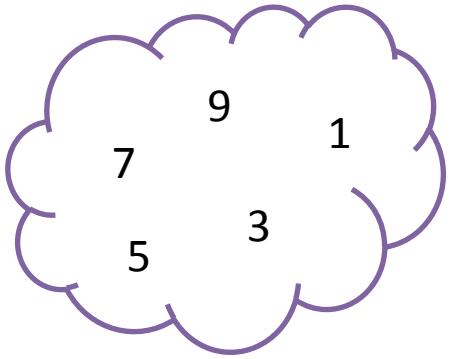
- Estas exceções normalmente são provocadas por problemas de programação, não devendo ser tratadas
- Por este motivo, o Java não obriga o programador a tratar as unchecked exceptions

JAVA ACADEMY

Softblue

Collections API: Conjuntos e Mapas

Conjuntos



- Representam conjuntos como na matemática
- Não permitem elementos duplicados
- A ordem dos elementos no conjunto pode não ser a mesma da ordem de inserção
- Representados pela interface `java.util.Set`

Tipos de Conjuntos

Softblue

`HashSet` → Não existe garantia de ordem

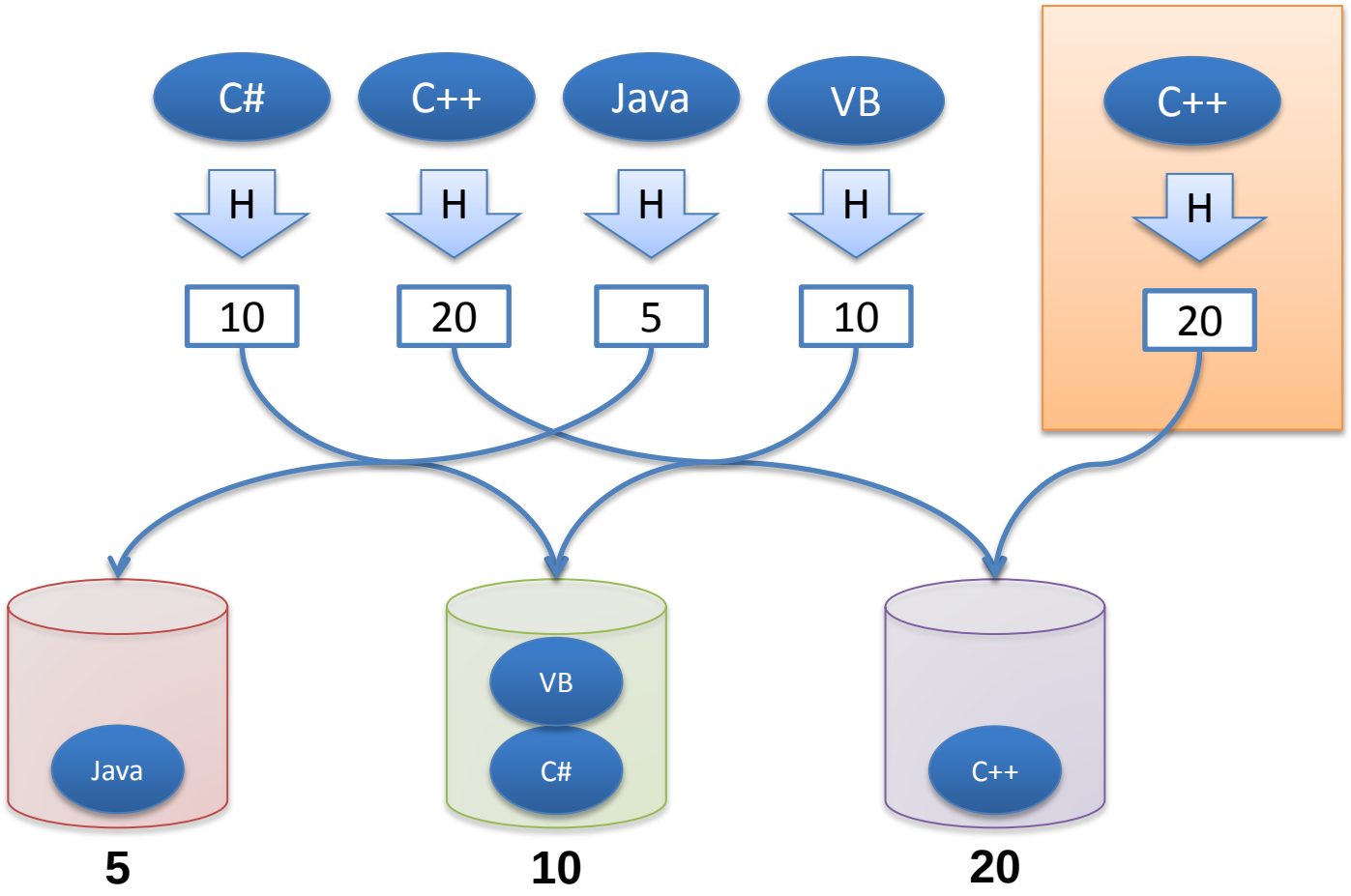
`LinkedHashSet` → A ordem de inserção é mantida

`TreeSet` → Os elementos são ordenados de acordo com um critério

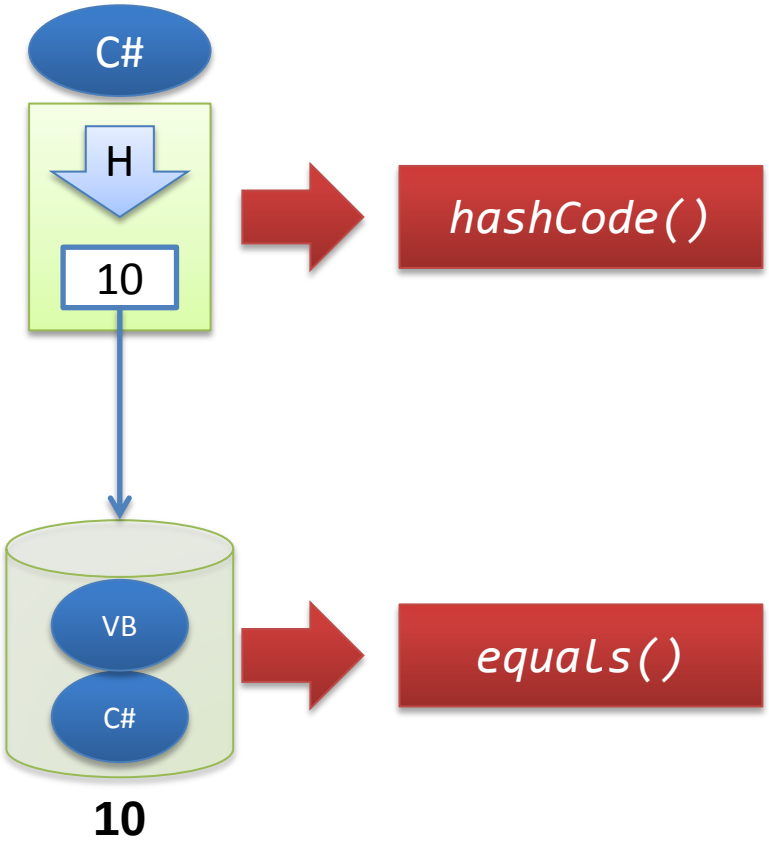
- Coleções com "Hash" no nome: usam equals() e hashCode()
- Coleções com "Tree" no nome: usam as interfaces Comparable ou Comparator

JAVA ACADEMY

Algoritmo de Hashing



Métodos equals() e hashCode()




```
public int hashCode()
{
    //...
}
```

O algoritmo de hashing deve retornar valores bem distribuídos

`hashCode()` → Encontra o índice

`equals()` → Encontra o elemento dentro do índice

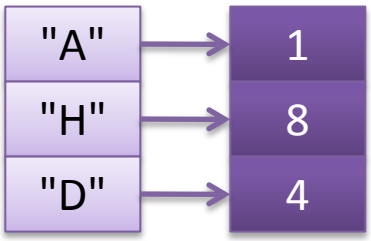
Interfaces Comparable & Comparator

Softblue

Usadas para determinar o critério de ordenação em elementos de coleções

JAVA ACADEMY

Mapas



- Utilizados quando é necessário mapear uma chave a um valor
- Chaves e valores podem ser qualquer tipo de objeto
- Representados pela interface *java.util.Map*

Tipos de Mapas

Softblue

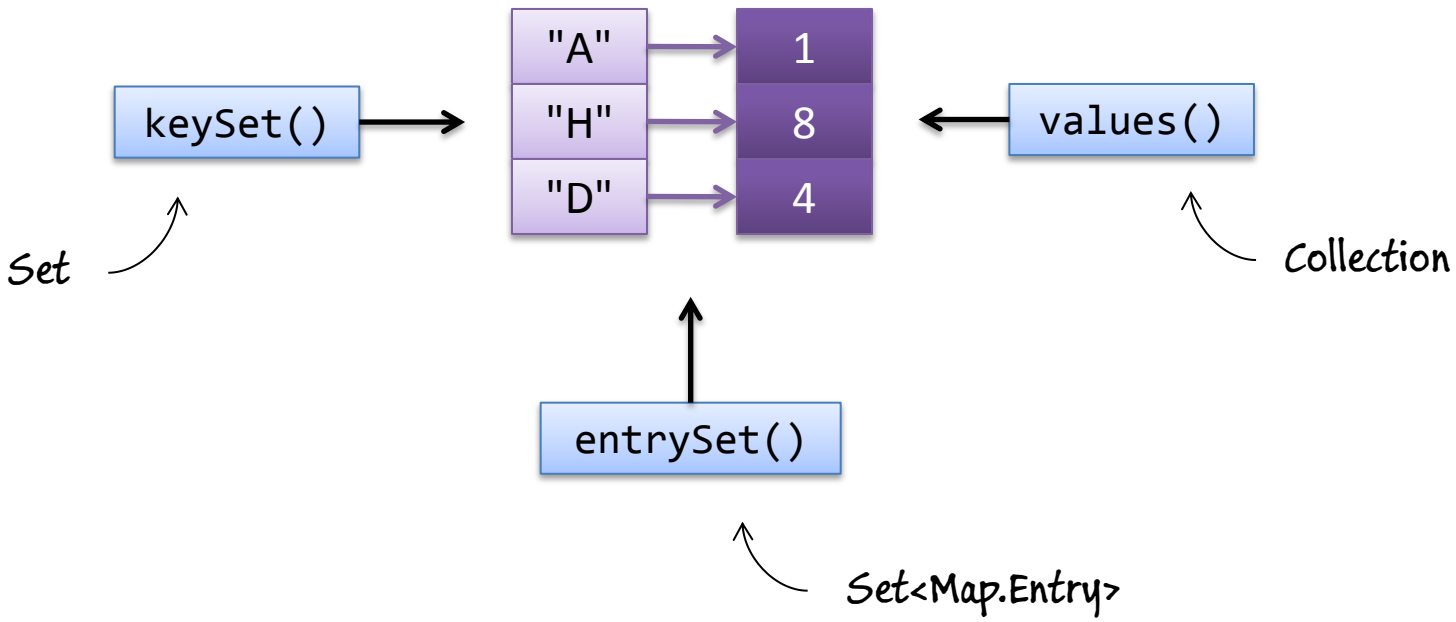
`HashMap` → Não existe garantia de ordem das chaves

`TreeMap` → As chaves são ordenadas de acordo com um critério

- Coleções com "Hash" no nome: usam equals() e hashCode()
- Coleções com "Tree" no nome: usam as interfaces Comparable ou Comparator

JAVA ACADEMY

Chaves & Valores




Softblue