

Criação e manipulação de objetos

Prof. Gustavo Wagner
(Alterações)

Slides originais: Prof. Tiago Massoni

Desenvolvimento de Sistemas

FATEC-PB
© Centro de Informática, UFPE

Criação de objetos

Um objeto é criado através do operador **new**

new Conta ()

Palavra reservada ordenando a criação de um objeto

Nome da classe do objeto a ser criado

Avaliação do operador `new`

Para avaliar uma expressão do tipo

`new NomeDaClasse()`

o computador...

- Cria um objeto da classe `NomeDaClasse` e armazena na sua memória
- Inicializa os atributos com valores default
- Devolve como resultado da avaliação uma referência para o objeto criado

Criando objetos

```
Conta c = null;  
  
c = new Conta();  
  
double saldo = c.getSaldo();  
  
c.setNumero(21);  
c.setSaldo(300);
```

Qual é o
saldo?

Inicialização
(tenho que
fazer isso
sempre?)

Além de métodos e atributos...

O corpo de uma classe pode conter
construtores
para inicializar os atributos dos objetos

```
public NomeDaClasse (Parâmetros) {  
    CorpoDoConstrutor  
}
```

Instruções para a
inicialização

Informações
necessárias para
inicialização

Primeiro construtor (s/ parâmetros)

```
class Conta {  
    private int numero;  
    private double saldo;  
    private Cliente titular;  
    ...  
  
    public Conta() {  
        saldo = 0;  
        numero = 0;  
        titular = null; Java já cria este  
    }                               construtor  
}                                     automaticamente
```

Outro construtor

```
class Conta {  
    ...  
    public Conta(int numero, double saldo,  
                  Cliente titular) {  
        this.numero = numero;  
        this.saldo = saldo;  
        this.titular = titular;  
    }  
}
```

Usando o construtor

```
Conta c = null;  
Cliente tit = new Cliente();  
...  
c = new Conta(20, 20.0, tit);  
double saldo = c.getSaldo();
```

E se tentarmos usar
new Conta(); ??

Sobrecarga

Como diferenciar??



```
class Conta {  
    ...  
    public Conta() {  
        this.numero = 0;  
        ...  
    }  
    public Conta(int numero, double saldo,  
                  Cliente titular) {  
        this.numero = numero;  
        this.saldo = saldo;  
        this.titular = titular;  
    }  
}
```

Manipulando objetos

- **Armazenar** a referência para o objeto, para depois ter acesso à mesma
- **Executar** (invocar ou chamar) os métodos do objeto através da sua referência
- **Passar** a referência para o objeto como argumento de um método ou construtor

Passando mensagens

Main é um possível usuário de objetos Conta e Cliente

```
class Programa {  
    public static void main(String [] args) {  
        Conta c = null;  
        c = new Conta(32,50.0,new Cliente());  
  
        c.creditar(50.0);  
  
        ...  
    }  
}
```

Passando referências

```
class Programa {  
    public static void main(String [] args) {  
        Conta c = null;  
        Cliente cli = new Cliente();  
        c = new Conta(32,50.0,cli);  
  
        c.creditar(50.0);  
        ...  
    }  
}
```

Aliasing

Mais de uma variável armazenando a referência para um dado objeto

```
Conta a = new Conta("12-7", 34.00);
```

```
Conta b;
```

```
b = a;
```

```
b.creditar(100);
```

```
console.println(a.getSaldo());
```

a e b passam a referenciar
o mesmo objeto

qualquer efeito via
b é refletido via **a**

Remoção de objetos

- Não existe mecanismo de remoção explícita de objetos da memória em Java (delete de C++)
- O coletor de lixo automático de Java elimina estes objetos da memória quando não são mais referenciados
- Processo de baixa prioridade da máquina virtual que permite melhor aproveitamento dos recursos (memória)

Remoção de objetos

- Apontar referências para null

```
Conta c;
```

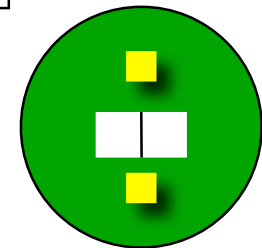
```
c = new Conta();
```

```
c.getSaldo();
```

```
c = null;
```

c == null

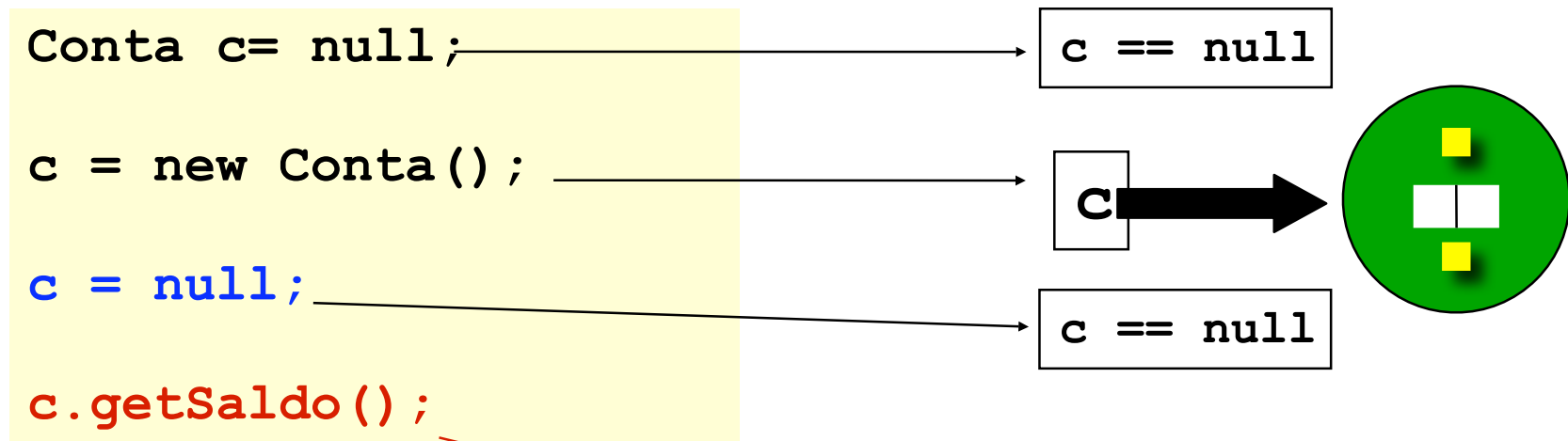
c



Coleta
automática
de lixo!

Chama método do
objeto "apontado" por c

Tempo de execução e referência null

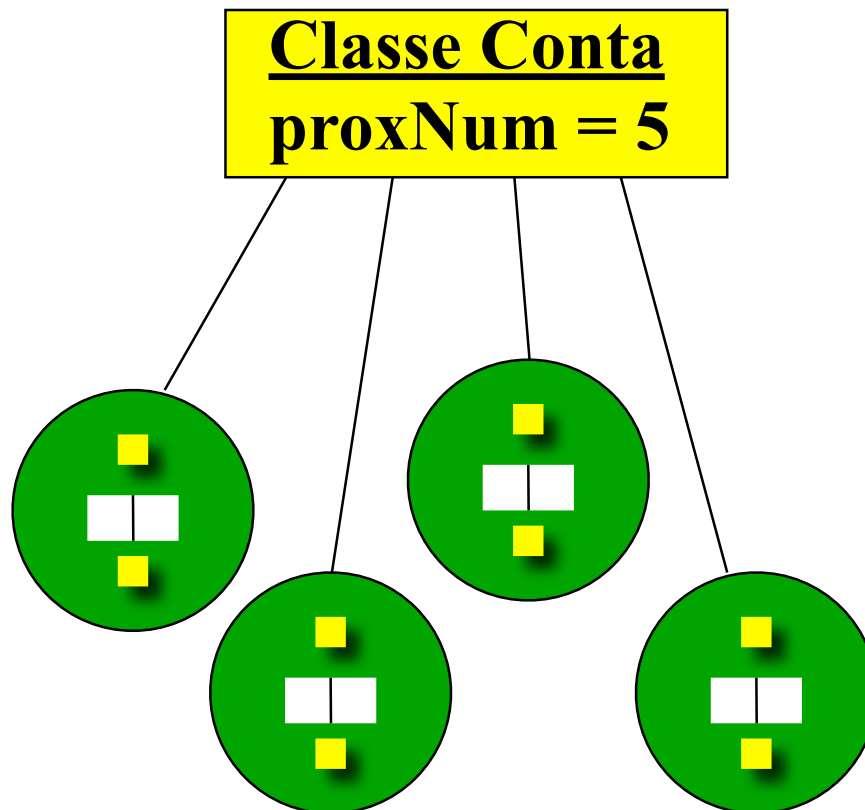


Erro de execução, mas não de compilação!

NullPointerException

Variáveis estáticas

```
public class Conta {  
    ...  
    private static int proximoNum = 1;  
    ...  
}
```



Podem ser vistas como variáveis globais, fugindo do paradigma orientado a objetos...

Aula Prática

Atividades

- Importem as classes da atividade anterior
- Criar uma classe Programa, que vai conter um método main
- Criar construtores para todas as classes Conta
-

Atividades

- O main deve
 - Pegar entrada do usuário (cpf, nome, saldo inicial)
 - Criar 1 cliente e 1 conta com aquele titular (número deverá ser gerado por uma variável estática)
 - Fazer um crédito com o valor novamente pedido ao usuário
 - Mostrar informações da conta na tela
 - Programa deve aguardar entrada de novos dados continuamente, até o usuário pedir para parar