Introdução à Programação Orientada a Objetos

Aula 4

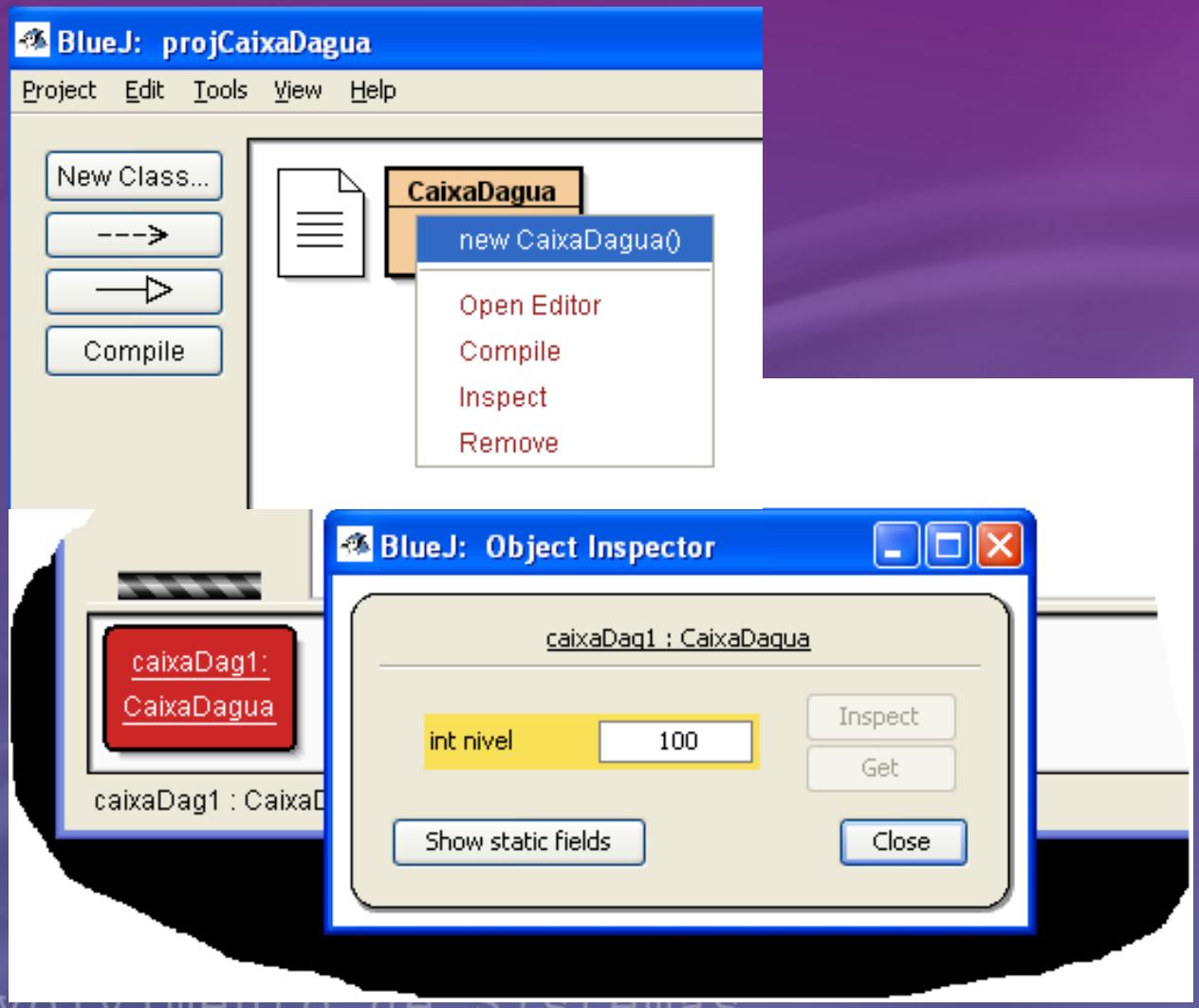
Análise e Desenvolvimento de Sistemas

Método Construtor

- É um método especial.
- É chamado automaticamente quando um objeto é criado.
- É usado para inicializar campos com valores diferentes dos default.
- Deve ter o mesmo nome da classe.
- Não pode retornar valor (nem void).

Exemplo da caixa dágua

```
class CaixaDagua
{
//    int nivel = 0;
    int nivel;
    CaixaDagua()
    {
        nivel = 100;
    }
    int mostrarNivel()
    {
        return nivel;
    }
    void acrescentarQua
    {
```



Operadores relacionais

- == igual
- != diferente
- > maior que
- < menor que
- >= maior ou igual a
- <= menor ou igual a
- Exemplo:

  2 > 1

  2 == 1

Operadores lógicos

- || OU
- && E
- ! NÃO
- Exemplo:

```
true || false
1 == 1 && 2 > 1
```

Expressões lógicas

- São as que resultam em um valor lógico: true ou false.
- Exemplo:

```
2 > 1  ||  3 > 1        // é uma expressão lógica
2 + 1  /  3 + 1         // não é uma expressão lógica
```

A instrução if

- Avalia uma expressão lógica e executa um bloco de comandos caso o resultado seja true.
- Sintaxe:

```
if(expressão lógica)
    Comando;
```

- Ou

```
if(expressão lógica)
{
    Bloco de comandos;
}
```

Exemplo da caixa dágua

```
class CaixaDagua
{
    int nivel = 0;
    int mostrarNivel()
    {
        return nivel;
    }
    void acrescentarQuantidade(int quanto)
    {
        nivel = nivel + quanto;
    }
    void esvaziarUmMetro()
    {
        if(nivel > 0)
            nivel = nivel - 1;
        else
            System.out.println("Não há mais água.");
    }
}
```

A instrução if-else

- Avalia uma expressão lógica e executa um bloco de comandos caso o resultado seja true ou outro bloco de comandos caso o resultado seja false.
- Sintaxe:

```
if(expressão lógica)
    Comando;
else
    Outro comando;
```

A instrução if-else

- Sintaxe:
- Ou

```
if(expressão lógica)
{
    Bloco de comandos;
}
else
{
    Bloco de comandos;
}
```

Encapsulamento

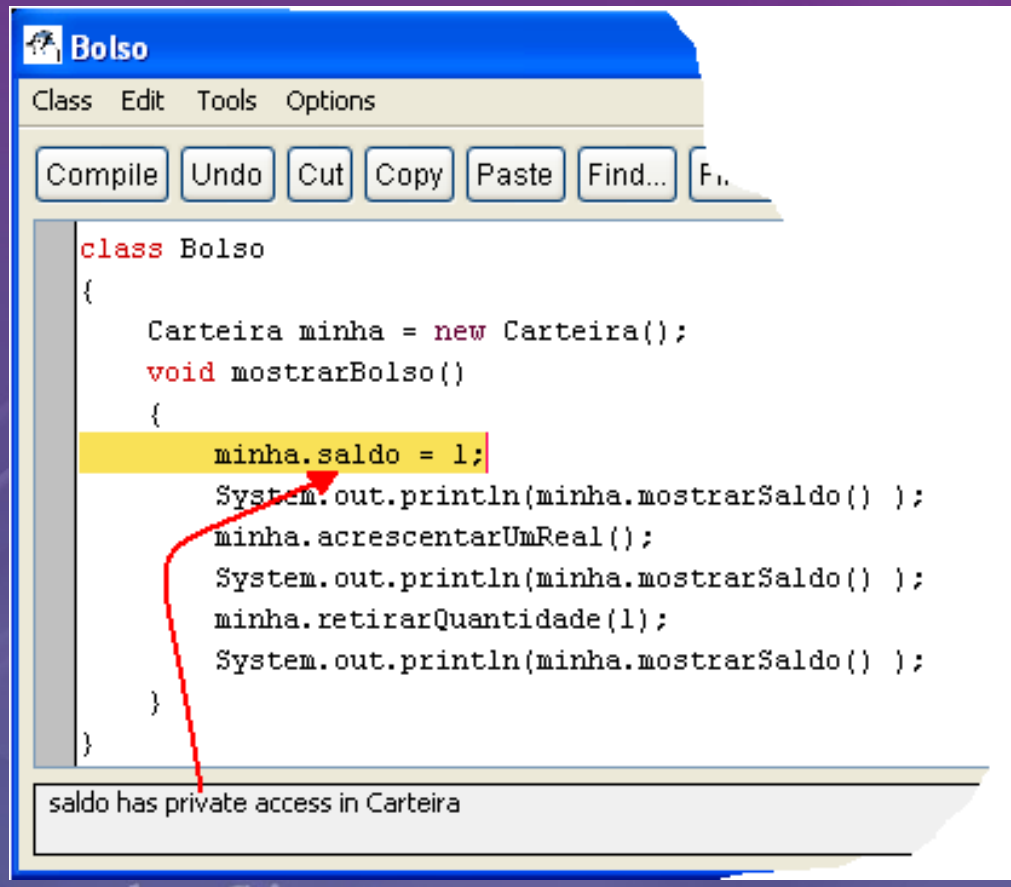
- É a capacidade de se esconder os detalhes mais internos do objeto e dar um meio para acessá-los.
- Para isso usa-se:
  - Modificadores de acesso.
  - Métodos de acesso.
  - Métodos modificadores.

Modificadores de acesso

- São mecanismos de permitir a visibilidade de classes, métodos e campos.
- São informados na extrema esquerda dos seus nomes.
  - public: campos, métodos e classes podem ser visualizados por quaisquer outros métodos e classes.
  - protected: será estudado juntamente com conceito de herança.
  - package: campos, métodos e classes podem ser visualizados por outros métodos e classes do mesmo pacote/pasta. (Default)
  - private: campos e métodos só podem ser vistos dentro da classe em que foram declarados.

Exemplos

```
public class Carteira
{
    private int saldo = 0;
    public int mostrarSaldo()
    {
        return saldo;
    }
    public void acrescentarUmReal()
    {
        saldo = saldo + 1;
    }
    public void retirarQuantidade(int quanto)
    {
        if(quanto >= saldo)
            saldo = saldo - quanto;
    }
}
```



Métodos de acesso

- São métodos que permitem obter o valor encapsulado de cada campo.

Métodos de acesso

```
public class Carteira
{
    private int saldo = 0;
    public int getSaldo()
    {
        return saldo;
    }
    public int mostrarSaldo()
    {
        return saldo;
    }
    public void acrescentar
```

```
class Bolso
{
    Carteira minha = new Carteira();
    void mostrarBolso()
    {
        //minha.saldo = 1;
        System.out.println(minha.getSaldo() );
        minha.acrescentarUmReal();
        System.out.println(minha.mostrarSaldo() )
        minha.retirarQuantidade(1);
```

Métodos modificadores

- São métodos que permitem alterar adequadamente os valores dos campos.

Exemplo

```
public class Carteira
{
    private int saldo = 0;
    public int getSaldo()
    {
        return saldo;
    }
    public void setSaldo(int umSaldo)
    {
        if(umSaldo >=0)
            saldo = umSaldo;
    }
    public int mostrarSaldo()
    {
```

```
class Bolso
{
    Carteira minha = new Carteira();
    void mostrarBolso()
    {
        minha.setSaldo(1);    //minha.saldo = 1;
        System.out.println(minha.getSaldo() );
        minha.acrescentarUmReal();
        System.out.println(minha.mostrarSaldo() );
        minha.retirarQuantidade(1);
        System.out.println(minha.mostrarSaldo() );
    }
}
```