

UNIVERSIDADE ESTÁCIO DE SÁ
ANÁLISE E DESENVOLVIMENTO DE SISTEMA
UNIDADE CAMPO GRANDE

PADRÕES DE PROJETOS DE SOFTWARE COM JAVA

Danielle Do Nascimento Coutinho - 202109094939
Caroline da Silva Longo – 202208616712
Maria Bruna Oliveira da Silva Gonçalves –202202883409

Rio de Janeiro - RJ

Exemplo 01:

```
public class VerificadorAprovacao {  
    public static void main(String[] args) {  
        double notaDaProva = 6.5; // Substitua isso pela nota da prova  
  
        if (notaDaProva >= 6.0) {  
            System.out.println("Aprovado!");  
        } else {  
            System.out.println("Reprovado!");  
        }  
    }  
}
```

Explicação:

Neste código, a variável nota representa a nota obtida na prova. Se nota for maior ou igual a 6.0, o programa imprimirá "Aprovado". Caso contrário, se a nota for menor que 6.0, o programa imprimirá "Reprovado".

Exemplo 02:

```
1  class Animal {
2      private int idade;
3      private String genero;
4
5      public Animal(int idade, String genero) {
6          this.idade = idade;
7          this.genero = genero;
8      }
9
10     public void setPelo(String pelo) {
11         System.out.println("O animal tem pelo " + pelo + ".");
12     }
13
14     public void fazerSom() {
15         System.out.println("O animal faz um som genérico.");
16     }
17 }
18
19 class Cachorro extends Animal {
20     public Cachorro(int idade, String genero) {
21         super(idade, genero);
22     }
23
24     public void fazerSom() {
25         System.out.println("O cachorro faz um som de latido.");
26     }
27 }
28
29 class Gato extends Animal {
30     public Gato(int idade, String genero) {
31         super(idade, genero);
32     }
33
34     public void fazerSom() {
35         System.out.println("O gato faz um som de miado.");
36     }
37 }
```

```

9 class Cavalo extends Animal {
10     public Cavalo(int idade, String genero) {
11         super(idade, genero);
12     }
13
14     public void fazerSom() {
15         System.out.println("O cavalo faz um som de relincho.");
16     }
17 }
18
19 public class Main {
20     public static void main(String[] args) {
21         Animal animal = new Animal(5, "Macho");
22         Cachorro cachorro = new Cachorro(3, "Fêmea");
23         Gato gato = new Gato(2, "Macho");
24         Cavalo cavalo = new Cavalo(7, "Macho");
25
26         animal.setPelo("curto");
27         animal.fazerSom();
28
29         cachorro.setPelo("longo");
30         cachorro.fazerSom();
31
32         gato.setPelo("macio");
33         gato.fazerSom();
34
35         cavalo.setPelo("curto");
36         cavalo.fazerSom();
37     }
38 }
39

```

Explicação:

Neste código, a classe Animal é a classe base com os atributos idade e gênero, bem como métodos para definir o pelo e fazer um som genérico. As subclasses Cachorro, Gato e Cavalo herdam da classe Animal e substituem o método fazerSom para fazer os sons específicos de cada animal. O método setPelo pode ser usado para definir o tipo de pelo do animal.

Exemplo 03:

```
1 import java.util.Scanner;
2
3 class Cliente {
4     private boolean contaAberta;
5     private boolean pedidoFeito;
6     private boolean pagamentoConfirmado;
7
8     public Cliente() {
9         contaAberta = false;
10        pedidoFeito = false;
11        pagamentoConfirmado = false;
12    }
13
14    public void abrirConta() {
15        contaAberta = true;
16        System.out.println("Conta aberta com sucesso.");
17    }
18
19    public void fazerPedido() {
20        if (contaAberta) {
21            pedidoFeito = true;
22            System.out.println("Pedido realizado com sucesso.");
23        } else {
24            System.out.println("Você deve abrir uma conta antes de fazer um pedido.");
25        }
26    }
27
28    public void confirmarPagamento() {
29        if (pedidoFeito) {
30            pagamentoConfirmado = true;
31            System.out.println("Pagamento confirmado.");
32        } else {
33            System.out.println("Você deve fazer um pedido antes de confirmar o pagamento.");
34        }
35    }
36
37    public boolean isPagamentoConfirmado() {
38        return pagamentoConfirmado;
39    }
40}
```

```
1 class Lanchonete {
2     public void receberPedido() {
3         System.out.println("A lanchonete recebeu o pedido.");
4     }
5
6     public void liberarPedidoParaEntregador() {
7         System.out.println("O pedido foi liberado para o entregador.");
8     }
9 }
10
11 class Entregador {
12     public void entregarPedido() {
13         System.out.println("O pedido foi entregue com sucesso.");
14     }
15 }
16
17 public class Main {
18     public static void main(String[] args) {
19         Cliente cliente = new Cliente();
20         Lanchonete lanchonete = new Lanchonete();
21         Entregador entregador = new Entregador();
22
23         Scanner scanner = new Scanner(System.in);
24
25         System.out.println("Bem-vindo ao sistema de delivery!");
26
27         cliente.abrirConta();
28
29         System.out.println("\nCliente deseja fazer um pedido? (Sim ou Não)");
30         String fazerPedido = scanner.next();
31
32         if (fazerPedido.equalsIgnoreCase("Sim")) {
33             cliente.fazerPedido();
34             System.out.println("\nCliente deseja confirmar o pagamento? (Sim ou Não)");
35             String confirmarPagamento = scanner.next();
36
37             if (confirmarPagamento.equalsIgnoreCase("Sim")) {
38                 cliente.confirmarPagamento();
39                 lanchonete.liberarPedidoParaEntregador();
40                 entregador.entregarPedido();
41             }
42         }
43     }
44 }
```

```
        if (confirmarPagamento.equalsIgnoreCase("Sim")) {  
            cliente.confirmarPagamento();  
            lanchonete.receberPedido();  
  
            if (cliente.isPagamentoConfirmado()) {  
                lanchonete.liberarPedidoParaEntregador();  
                entregador.entregarPedido();  
                System.out.println("\nPedido entregue com sucesso!");  
            }  
        }  
    }  
  
    scanner.close();  
}
```

Explicação:

Neste código, o cliente abre uma conta, faz um pedido e confirma o pagamento. A lanchonete recebe o pedido e o libera para o entregador, que finaliza o processo entregando o pedido.