Programação de Soluções Computacionais – PSC

Exercícios de Laboratório – Prof. Calvetti

Aula: 05
Associação de Objetos – Modularização

Exemplo Resolvido: Copie o código fornecido, cole-o em sua I.D.E., antes de executá-lo, analise o código e entenda-o, só depois, então execute-o para ver os resultados.

Ex1) Faça:
a. Crie a Classe Turma, com atributos privados codigo, do tipo String, e ano, do tipo int. Crie um construtor que receba parâmetros para inicializar os atributos e os métodos de acesso e métodos modificadores.
b. Altere a classe Aluno criada na semana passada para que tenha também um atributo privado turma do tipo Turma. Altere o construtor para receber um parâmetro que inicialize o novo atributo e crie o método de acesso e o modificador para este novo atributo.
c. Crie métodos getDados em ambas as classes que retornam strings com o valor dos atributos.
d. Altere a classe TesteAluno feita na semana passada para tratar este o novo atributo da classe Aluno.

Solução: Classe Turma

```
public class Turma {
    //atributos
    private String codigo;
    private int ano;
    //construtor
    public Turma(String codigo, int ano) {
        this.codigo = codigo;
        this.ano = ano;
    }
    //metodos de acesso
    public String getCodigo() {
        return codigo;
    }
    public int getAno() {
        return ano;
    }
    //metodos modificadores
    public void setCodigo(String codigo) {
        this.codigo = codigo;
    }
    public void setAno(int ano) {
        this.ano = ano;
    }
    //metodo getDados
    public String getDados() {
        return "Turma [codigo=" + codigo + ", ano=" + ano + "]";
    }
}
```

Solução: Classe Aluno

```
public class Aluno {
    // atributos
    private String nome;
    private int idade;
    private double peso;
    private boolean formando;
    private char sexo;
    private Turma turma;
    //construtor
    public Aluno(String nome, int idade, double peso, char sexo, Turma turma) {
        this.nome = nome;
        this.idade = idade;
        this.peso = peso;
        this.formando = false;
        this.sexo = sexo;
        this.turma = turma;
    }
    //metodos de acesso
    public String getNome() {
        return nome;
    }
    public int getIdade() {
        return idade;
    }
    public double getPeso() {
        return peso;
    }
    public boolean getFormando() {
        return formando;
    }
    public char getSexo() {
        return sexo;
    }
    public Turma getTurma() {
        return turma;
    }
    //metodos modificadores
    public void setNome(String nome) {
        this.nome = nome;
    }
    public void setIdade(int idade) {
        this.idade = idade;
    }
    public void setPeso(double peso) {
        this.peso = peso;
    }
    public void setFormando(boolean formando) {
        this.formando = formando;
    }
    public void setSexo(char sexo) {
        this.sexo = sexo;
    }
    public void setTurma(Turma turma) {
        this.turma = turma;
    }
    //metodo getDados
    public String getDados() {
        return "Aluno [nome=" + nome + ", idade=" + idade + ", peso=" + peso + ", formando=" + formando + ",
sexo=" + sexo + ", turma=" + turma.getDados() + "]";
    }
}
```

Solução: Classe TesteAluno

```
import javax.swing.JOptionPane;

public class TesteAluno {

    // cadastrar um novo aluno no metodo main
    public static void main(String[] args) {

        // coletando os dados do aluno a ser cadastrado

        String nome = JOptionPane.showInputDialog("Nome");
        int idade = Integer.parseInt(JOptionPane.showInputDialog("Idade"));
        double peso = Double.parseDouble(JOptionPane.showInputDialog("Peso"));
        // pega o primeiro caracter da String e retorna como char
        char sexo = JOptionPane.showInputDialog("Sexo M/F").charAt(0);
        String codigo = JOptionPane.showInputDialog("Codigo da Turma");
        int ano =
            Integer.parseInt(JOptionPane.showInputDialog("Ano da Turma"));

        //cria a turma
        Turma turma = new Turma(codigo, ano);

        // cria um objeto aluno
        Aluno aluno = new Aluno(nome, idade, peso, sexo, turma);

        // nao precisa mais montar a string de saida, e so chamar o metodo getDados

        // mostra o aluno
        JOptionPane.showMessageDialog(null, aluno.getDados());

        // altera informacoes; nao precisa criar todas as variaveis novamente
        idade = Integer.parseInt(JOptionPane.showInputDialog("Idade"));
        peso = Double.parseDouble(JOptionPane.showInputDialog("Peso"));
        // tem que digitar true ou false
        boolean formando = Boolean.parseBoolean(
            JOptionPane.showInputDialog("E' formando?true/false"));

        // muda usando os metodo modificadores
        aluno.setIdade(idade);
        aluno.setPeso(peso);
        aluno.setFormando(formando);

        // mostra novamente o cadastro do aluno

        // mostra o aluno
        JOptionPane.showMessageDialog(null, aluno.getDados());
    }
}
```

Note que há trechos com código repetido foram substituídos pelos métodos getDados. Este é um exemplo de modularização. Outro exemplo é o próprio fato de separar o código em 3 classes diferentes, cada uma com papeis distintos.

Problemas Propostos:

- Faça uma classe para cada solução;
- O nome da classe pode ser Solucao1a, Solucao1b, e assim por diante;
- Não use ***`Scanner`*** para ler dados;
- Os alunos podem consultar qualquer material.

1) Crie as classes conforme abaixo:

a. Crie a classe Professor com seu construtor, métodos de acesso e modificadores e os atributos privados nome, do tipo String, idade, do tipo int. Crie o método getDados que retorna o valor dos atributos.

b. Crie a classe Disciplina com seu construtor, métodos de acesso e modificadores e os atributos privados nome, do tipo String, pratica, do tipo boolean. Crie o método getDados que retorna o valor dos atributos.

c. Crie a classe Atribuicao com seu construtor, métodos de acesso e modificadores e os atributos privados professor, do tipo Professor, e disciplina, do tipo Disciplina. Crie o método getDados que retorna o valor dos atributos.

d. Crie a classe TesteAtribuicao com o método main que instancia um Professor, uma Disciplina e uma Atribuicao. Imprima dos dados da Atribuicao.

2) Crie as classes Cliente, ContaCorrente e Agencia conforme abaixo:

a) A classe Cliente possui os atributos nome e cpf, ambos do tipo String, e um atributo conta do tipo ContaCorrente. Crie um construtor que recebe os atributos como parâmetros e os métodos de acesso e os modificadores.

b) A classe ContaCorrente tem os atributos numero e digito, ambos inteiros, o atributo agencia do tipo Agencia e o atributo saldo do tipo double. Crie um construtor que recebe os atributos como parâmetros e os métodos de acesso e os modificadores. Crie também um método depositar que receba um parâmetro double com o valor do depósito e aumente o saldo da conta. Crie também um método sacar que receba um parâmetro double com o valor do saque e diminua o saldo da conta. A conta não pode ficar negativa. Neste caso, deve ser dada uma mensagem que o saque não foi efetuado e o retorno deve ser zero. Caso contrário o retorno deve ser o valor sacado. Crie também um método consultarSaldo que não recebe parâmetros e retorne o saldo. Crie, finalmente, um método imprimirSaldo que imprima o numero da conta corrente com dígito, o número da agência com dígito e o saldo da conta corrente.

c) Ainda na classe ContaCorrente, o número da conta deve ter no máximo 4 dígitos e ser positivo. O digito da conta deve ser validado a partir do seguinte algoritmo de módulo 11: multiplique o primeiro dígito da conta por 4, o segundo por 6, o terceiro por 8 e o quarto por 2; some tudo e calcule o resto da divisão (módulo) da soma por 11. Este é o valor do dígito. Obs: se o resultado for 10 o dígito é 0.

d) A classe Agencia tem os atributos nome do tipo String, numero e digito do tipo int. Crie um construtor que recebe os atributos como parâmetros e os métodos de acesso e os modificadores. O número e o digito da Agencia devem seguir os mesmos padrões do número e do dígito da conta corrente.

e) Para testar faça uma classe CaixaEletronico, que irá conter o método main. No main instancie um cliente com os seguintes dados:

Nome: Ademar Apior
CPF: 123231518-12
Conta Corrente: 1234 Dígito: 4
Agencia: 7890 Dígito: 5
Saldo Inicial: 150.00

Operações:
- Sacar 140.0 (sucesso)
- Consultar saldo (resultado é 10.0)
- Sacar 200.0 (falha)
- Consultar saldo (resultado é 10.0)
- Depositar 25.45 (sucesso)
- Imprimir saldo (além dos dados de cliente, conta e agencia, o saldo deve ser 35.45)

3) Criar as classes CondicionadorDeAr, Termostato.

a) A classe CondicionadorDeAr tem um atributo termostato do tipo Termostato e um atributo boolean chamado ligado. O construtor de CondicionadorDeAr não recebe nenhum parâmetro, mas instancia um termostato e coloca ligado em false. Crie um método de acesso para ligado e outro para termostato. Não precisa fazer os métodos modificadores. Crie um método ligar, que muda ligado para true, e um desligar, que muda ligado para false. Crie um método aumentarTemperatura, que aumenta a temperatura do termostato em um grau cada vez que é chamado até o limite de 28 graus. Crie um método reduzirTemperatura que reduz a temperatura em um grau cada vez que é chamado até o limite de 15 graus. Crie um método imprimirTemperatura que imprime a temperatura atual. Não se esqueça de verificar se o condicionador está ligado antes de aumentar ou diminuir a temperatura ou imprimi-la.

b) A classe Termostato tem um atributo temperatura. Seu construtor não recebe parâmetros, mas instancia a temperatura em 20. Crie um método de acesso o outro modificador. Estes métodos devem respeitar os limites estabelecidos no item anterior.

c) Crie a classe Usuario, com o método main. Neste método você deve instanciar um CondicionadorDeAr, aumentar a temperatura para 30 graus (receber mensagem de erro), reduzir a temperatura para 10 gruas (receber mensagem de erro). Aumentar a temperatura para 25 graus e imprimir a temperatura.

Bibliografia

LOPES, ANITA. GARCIA, GUTO. Introdução à Programação: 500 algoritmos resolvidos. Rio de Janeiro: Elsevier, 2002.

DEITEL, P. DEITEL, H. Java: como programar. 8 Ed. São Paulo: Prentice – Hall (Pearson), 2010.