

Programação Orientada a Objetos

Referências entre Objetos

Exercícios



Joinville Batista Junior

Exemplo

```
package util;

import java.util.Calendar;
import java.util.GregorianCalendar;

public class Data {

    private int dia, mês, ano;

    public Data(int dia, int mês, int ano) {
        this.dia = dia;
        this.mês = mês;
        this.ano = ano;
    }

    public int getDia() {
        return dia;
    }
}
```

Exemplo

```
public int getMês() {
    return mês;
}

public int getAno() {
    return ano;
}

public int calcularIdade (){
    GregorianCalendar data_atualGregorianCalendar = new GregorianCalendar();
    int dia_atual = data_atualGregorianCalendar.get(Calendar.DAY_OF_MONTH);
    int mês_atual = data_atualGregorianCalendar.get(Calendar.MONTH) + 1;
    // GregorianCalendar: mês varia de 0 a 11
    int ano_atual = data_atualGregorianCalendar.get(Calendar.YEAR);
    int idade = ano_atual - ano;
    if ((mês_atual < mês) || ((mês_atual == mês) && (dia_atual < dia))) idade--;
    return idade;
}
```

Exemplo

```
public int compareTo(Data outraData) {
    if (ano > outraData.getAno()) return 1;
    if (ano < outraData.getAno()) return -1;
    if (mês > outraData.getMês()) return 1;
    if (mês < outraData.getMês()) return -1;
    if (dia > outraData.getDia()) return 1;
    if (dia < outraData.getDia()) return -1;
    return 0;
}

public String toString(){
    String data;
    if (dia < 10) data = "0" + dia; else data = "" + dia;
    if (mês < 10) data += "/0" + mês + "/"; else data += "/" + mês + "/";
    return data += ano;
}

}
```

Exemplo

```
package entidade;

import util.Data;

public class Aluno {

    private String nome, rg, cpf;
    private Data nascimentoData;
    private char sexo;
    private Endereço alunoEndereço;
```

Exemplo

```
public Aluno(String nome, String rg, String cpf,
    Data nascimentoData, char sexo, Endereço alunoEndereço) {
    this.nome = nome;
    this.rg = rg;
    this.cpf = cpf;
    this.nascimentoData = nascimentoData;
    this.sexo = sexo;
    this.alunoEndereço = alunoEndereço;
}

public String getNome() {
    return nome;
}

public String getRg() {
    return rg;
}
```

Exemplo

```
public String getCpf() {
    return cpf;
}

public Data getDataNascimento() {
    return nascimentoData;
}

public char getSexo() {
    return sexo;
}

public Endereço getEndereço() {
    return alunoEndereço;
}

public void setEndereço(Endereço alunoEndereço) {
    this.alunoEndereço = alunoEndereço;
}
```

Exemplo

```
public String toStringSexo() {
    if ((sexo == 'M') || (sexo == 'm')) return "masculino";
    if ((sexo == 'F') || (sexo == 'f')) return "feminino";
    return "indefinido";
}

public String toString() {
    return nome + " - RG:" + rg + " - CPF:" + cpf
        + " - nascimento:" + nascimentoData + " - sexo:" + toStringSexo()
        + "\n    " + alunoEndereço.toString();
}

}
```

Exemplo

```
package entidade;

public class Endereço {

    private String logradouro, complemento, bairro, cidade, cep;
    private int número;

    public Endereço(String logradouro, int número, String complemento,
        String bairro, String cidade, String cep) {
        this.logradouro = logradouro;
        this.número = número;
        this.complemento = complemento;
        this.bairro = bairro;
        this.cidade = cidade;
        this.cep = cep;
    }
}
```

Exemplo

```
public String getLogradouro() {
    return logradouro;
}

public void setLogradouro(String logradouro) {
    this.logradouro = logradouro;
}

public int getNúmero() {
    return número;
}

public void setNúmero(int número) {
    this.número = número;
}

public String getComplemento() {
    return complemento;
}
}
```

Exemplo

```
public void setComplemento(String complemento) {
    this.complemento = complemento;
}

public String getBairro() {
    return bairro;
}

public void setBairro(String bairro) {
    this.bairro = bairro;
}

public String getCidade() {
    return cidade;
}

public void setCidade(String cidade) {
    this.cidade = cidade;
}
```

Exemplo

```
public String getCep() {
    return cep;
}

public void setCep(String cep) {
    this.cep = cep;
}

public String toString() {
    String info = logradouro + " " + número;
    if (complemento != null) info += " - " + complemento;
    info += " - Bairro: " + bairro + " - " + cidade + " - CEP:" + cep;
    return info;
}

}
```

Exemplo

```
package entidade;

import java.util.ArrayList;

public class Curso {

    private String título;
    private int carga_horária_semanal, duração_semanas;
    private ArrayList<Curso> pré_requisitos_listaCurso = new ArrayList();

    public Curso(String título, int carga_horária_semanal, int duração_semanas) {
        this.título = título;
        this.carga_horária_semanal = carga_horária_semanal;
        this.duração_semanas = duração_semanas;
    }
}
```

Exemplo

```
public String getTítulo() {
    return título;
}

public int getCargaHoráriaSemanal() {
    return carga_horária_semanal;
}

public int getDuraçãoSemanas() {
    return duração_semanas;
}

public ArrayList<Curso> getPréRequisitos() {
    return pré_requisitos_listaCurso;
}
```

Exemplo

```
public boolean inserirPréRequisito(Curso pré_requisitoCurso) {
    if (!pré_requisitos_listaCurso.contains(pré_requisitoCurso)) {
        pré_requisitos_listaCurso.add(pré_requisitoCurso);
        return true;
    } else return false;
}

public boolean removePréRequisito(Curso pré_requisitoCurso) {
    return pré_requisitos_listaCurso.remove(pré_requisitoCurso);
}
```

Exemplo

```
public String toString() {
    String dados = título + " - "
        + carga_horária_semanal + " horas semanais - "
        + duração_semanas + " semanas";
    int n_pré_requisitos = pré_requisitos_listaCurso.size();
    if (n_pré_requisitos > 0) {
        dados += "\n    pré-requisitos: " + pré_requisitos_listaCurso.get(0).getTítulo();
        for (int n = 1; n < n_pré_requisitos; n++) {
            dados += " - " + pré_requisitos_listaCurso.get(n).getTítulo();
        }
    }
    return dados;
}

}
```


Exemplo

```
package entidade;

import util.Data;

public class Matrícula {

    public enum Bolsa {completa, parcial, nenhuma};

    private Aluno matrículaAluno;
    private Curso matrículaCurso;
    private Bolsa bolsa;
    private Data inícioData;
    private float nota_final;

    public Matrícula(Aluno matrículaAluno, Curso matrículaCurso, Bolsa bolsa,
        Data inícioData) {
        this.matrículaAluno = matrículaAluno;
        this.matrículaCurso = matrículaCurso;
        this.bolsa = bolsa;
        this.inícioData = inícioData;
        nota_final = -1.0f;
    }
}
```

Exemplo

```
public Aluno getAluno() {
    return matrículaAluno;
}

public Curso getCurso() {
    return matrículaCurso;
}

public Bolsa getBolsa() {
    return bolsa;
}

public Data getDataInício() {
    return inícioData;
}
```

Exemplo

```
public float getNotaFinal() {
    return nota_final;
}

public void setNotaFinal(float nota_final) {
    this.nota_final = nota_final;
}

public String toString() {
    String dados = "Aluno: " + matrículaAluno.toString()
        + "\n    Curso: " + matrículaCurso.toString()
        + "\n    data início: " + inícioData;
    if (bolsa != Bolsa.nenhuma) dados += " - bolsa: " + bolsa;
    if (nota_final >= 0) dados += " - nota final: " + nota_final;
    return dados;
}
}
```

Exemplo

```
package controle;

import entidade.Aluno;
import entidade.Curso;
import entidade.Endereço;
import entidade.Matrícula;
import entidade.Matrícula.Bolsa;
import java.util.ArrayList;
import java.util.HashMap;
import util.Data;

public class EscolaInformática {

    private static HashMap<String, Curso> hashCurso = new HashMap();
        // indexação pelo título
    private static HashMap<String, Aluno> hashAluno = new HashMap();
        // indexação pelo CPF
    private static ArrayList<Matrícula> listaMatrícula = new ArrayList();
}
```

Exemplo

```
public static void main(String[] args) {
    cadastrarCursos();
    cadastrarAlunos();
    cadastrarMatrículas();
    imprimirMatrículas(listaMatrícula);
}

private static void inserirCurso (Curso novoCurso) {
    String chave_curso = novoCurso.getTítulo();
    if (hashCurso.get(chave_curso) == null) hashCurso.put(chave_curso, novoCurso);
    else System.out.println("Curso já cadastrado");
}

private static void inserirAluno (Aluno novoAluno) {
    String chave_aluno = novoAluno.getCpf();
    if (hashAluno.get(chave_aluno) == null) hashAluno.put(chave_aluno, novoAluno);
    else System.out.println("Aluno já cadastrado");
}
```

Exemplo

```
private static void inserirMatrícula(String cpf_aluno, String título_curso,
    Matrícula.Bolsa bolsa, Data inícioData) {
    Aluno matrículaAluno = hashAluno.get(cpf_aluno);
    Curso matrículaCurso = hashCurso.get(título_curso);
    if (matrículaAluno == null) {
        System.out.println("Matrícula mal sucedida: aluno não cadastrado");
        return;
    }
    if (matrículaCurso == null) {
        System.out.println("Matrícula mal sucedida: curso não cadastrado");
        return;
    }
    listaMatrícula.add(new Matrícula(matrículaAluno, matrículaCurso, bolsa, inícioData));
}
```

Exemplo

```
private static void cadastrarCursos() {
    inserirCurso(new Curso("Java Básico", 8, 10));
    inserirCurso(new Curso("Java Aplicações Locais", 8, 10));
    inserirCurso(new Curso("Java Aplicações Web", 8, 10));
    inserirCurso(new Curso("Banco de Dados Básico", 8, 8));
    inserirCurso(new Curso("Banco de Dados Avançado", 8, 8));
    Curso novo1Curso = hashCurso.get("Java Básico");
    Curso novo2Curso = hashCurso.get("Java Aplicações Locais");
    Curso novo3Curso = hashCurso.get("Java Aplicações Web");
    Curso novo4Curso = hashCurso.get("Banco de Dados Básico");
    Curso novo5Curso = hashCurso.get("Banco de Dados Avançado");
    novo2Curso.inserirPréRequisito(novo1Curso);
    novo2Curso.inserirPréRequisito(novo4Curso);
    novo3Curso.inserirPréRequisito(novo2Curso);
    novo5Curso.inserirPréRequisito(novo4Curso);
}
```

Exemplo

```
private static void cadastrarAlunos() {
    inserirAluno(new Aluno("Ana Julia Parra", "31.845.917",
        "212.234.571-32", new Data(15, 10, 1982), 'F',
        new Endereço("Rua Arco Verde", 171, "apto 301", "Água Boa",
        "Dourados", "79810-015")));
    inserirAluno(new Aluno("Ana Ligia Silveira", "32.870.923",
        "312.434.775-30", new Data(8, 8, 1985), 'F',
        new Endereço("Rua Chapéu Velho", 303, "", "Rouxinol",
        "Dourados", "79820-017")));
    inserirAluno(new Aluno("André Oliveira", "41.825.341",
        "531.331.740-71", new Data(20, 3, 1993), 'M',
        new Endereço("Rua Sino da Mata", 303, "", "Brejão",
        "Carapó", "733100-000")));
}
```

Exemplo

```
public static void cadastrarMatrículas() {
    inserirMatrícula("212.234.571-32", "Java Básico",
        Bolsa.completa, new Data(1, 3, 2013));
    inserirMatrícula("312.434.775-30", "Java Básico",
        Bolsa.parcial, new Data(1, 3, 2013));
    inserirMatrícula("212.234.571-32", "Java Aplicações Locais",
        Bolsa.parcial, new Data(1, 6, 2013));
    inserirMatrícula("312.434.775-30", "Java Aplicações Locais",
        Bolsa.nenhuma, new Data(1, 6, 2013));
    inserirMatrícula("212.234.571-32", "Java Aplicações Web",
        Bolsa.parcial, new Data(1, 9, 2013));
    inserirMatrícula("312.434.775-30", "Java Aplicações Web",
        Bolsa.nenhuma, new Data(1, 9, 2013));
    inserirMatrícula("212.234.571-32", "Banco de Dados Básico",
        Bolsa.nenhuma, new Data(1, 3, 2013));
    inserirMatrícula("312.434.775-30", "Banco de Dados Básico",
        Bolsa.nenhuma, new Data(1, 3, 2013));
    inserirMatrícula("531.331.740-71", "Banco de Dados Básico",
        Bolsa.nenhuma, new Data(1, 3, 2013));
    inserirMatrícula("212.234.571-32", "Banco de Dados Avançado",
        Bolsa.nenhuma, new Data(1, 6, 2013));
    inserirMatrícula("531.331.740-71", "Banco de Dados Avançado",
        Bolsa.nenhuma, new Data(1, 6, 2013));
}
```

Exemplo

```
private static ArrayList<Matrícula> selecionarMatrículas() {
    int idade_mínima_aluno = -1;
    String cidade_aluno = "Dourados";
    char curso_sem_prérequisitos = 'X';
    int mínimo_semanas_curso = 10;
    Bolsa bolsa_matrícula = Bolsa.parcial;
    System.out.println("\nFiltros de Matrícula:");
    + "\n - idade mínima do aluno: " + idade_mínima_aluno
    + "\n - cidade do aluno: " + cidade_aluno
    + "\n - curso sem prérequisitos: " + curso_sem_prérequisitos
    + "\n - mínimo de semanas do curso: " + mínimo_semanas_curso
    + "\n - bolsa da matrícula: " + bolsa_matrícula;
    return filtrarMatrículas(idade_mínima_aluno, cidade_aluno, curso_sem_prérequisitos,
        mínimo_semanas_curso, bolsa_matrícula);
}
```

Exemplo

```
private static ArrayList<Matrícula> filtrarMatrículas
(int idade_mínima_aluno, String cidade_aluno, char curso_sem_prérequisitos,
int mínimo_semanas_curso, Bolsa bolsa_matrícula) {
    ArrayList<Matrícula> selecionadas_listaMatrícula = new ArrayList();
    for (Matrícula itemMatrícula : listaMatrícula) {
        Aluno matrículaAluno = itemMatrícula.getAluno();
        Curso matrículaCurso = itemMatrícula.getCurso();
        int n_pré_requisitos = matrículaCurso.getPréRequisitos().size();
        int idade_aluno = matrículaAluno.getDataNascimento().calcularIdade();
        String cidade_aluno1 = matrículaAluno.getEndereço().getCidade();
        int duração_semanas_curso = matrículaCurso.getDuraçãoSemanas();
        Bolsa bolsa_matrícula1 = itemMatrícula.getBolsa();
        if ((idade_mínima_aluno != -1) && (idade_aluno < idade_mínima_aluno)) continue;
        if ((cidade_aluno != null) && !cidade_aluno.equals(cidade_aluno1)) continue;
        if ((curso_sem_prérequisitos == 'T') && (n_pré_requisitos > 0)) continue;
        if ((curso_sem_prérequisitos == 'F') && (n_pré_requisitos == 0)) continue;
        if ((mínimo_semanas_curso != -1)
            && (duração_semanas_curso < mínimo_semanas_curso)) continue;
        if ((bolsa_matrícula != null) && (bolsa_matrícula != bolsa_matrícula1)) continue;
        selecionadas_listaMatrícula.add(itemMatrícula);
    }
    return selecionadas_listaMatrícula;
}
```

Exemplo

```
private static void imprimirMatrículas(String cabeçalho, ArrayList<Matrícula>
listaMatrícula) {
    System.out.println("\n" + cabeçalho);
    for (Matrícula itemMatrícula : listaMatrícula) {
        System.out.println(itemMatrícula.toString());
    }
}
}
```

Lista de Exercícios

2.1 – Adapte o exemplo da apostila:

- a) substituindo as classes Aluno, Curso e Matrícula por Cliente (nome, CPF, CNPJ, email, endereço), Produto (descrição, tipo [eletrodoméstico, eletrônico, roupa]) e Venda (cliente, produto, data, entrega rápida [sim, não]);
- b) substituindo os filtros de impressão por: cidade do cliente, tipo do produto, entrega rápida da venda;
- c) substituindo a classe EscolaInformática por VendaProdutos.

2.2 – Adapte o exemplo da apostila:

- a) substituindo as entidades por classes de sua escolha, utilizando pelo menos: um dado do tipo enumerado e um dado booleano;
- b) e definindo os filtros que julgar necessário, incluído pelo menos um filtro booleano e um enumerado.

Lista de Exercícios

2.3 – Adapte o exemplo da apostila:

- a) divida o método cadastrarMatrículas em 3 métodos que cadastram as matrículas na mesma data: março, junho e setembro;
- b) implemente 3 métodos void atribuirNotas() na classe EscolaInformática, atribuindo notas aprovadas (≥ 7) e reprovadas;
- c) execute o respectivo atribuirNotas(), com as notas associadas às disciplinas concluídas, após cada cadastramento de matrícula;
- d) altere o método inserirMatrícula para aceitar somente matrículas de alunos que tenham sido aprovados nos pré-requisitos.

2.4 – Adapte o exemplo da apostila:

- a) para ordenar as matrículas por nome de alunos;
- b) e para cada nome de aluno, por data decrescente de matrícula.