

Nome: João Paulo de Oliveira

11611BCC046

5º Aula prática

Uberlândia

2016

1.Código fonte:

- Controleacademico.dominio.cursos:

- Disciplina.java:

```
package controleacademico.dominio.cursos;
import java.util.*;

public class Disciplina {
    private String nome;
    private ArrayList<Disciplina> prerequisitos;

    public Disciplina (String nome) {
        this.nome = nome;
    }

    void criaPrerequisito (Disciplina d) {
        prerequisitos.add(d);
    }

    public String getNome() {
        return nome;
    }
}
```

- GradeCurricular.java:

```
package controleacademico.dominio.cursos;
import java.util.Vector;

public class GradeCurricular {
    private Vector<Disciplina> disciplinas = new Vector<Disciplina>();

    public void incluiDisciplina (Disciplina d) {
        disciplinas.add(d);
    }

    void removeDisciplina (Disciplina d) {
        disciplinas.remove(d);
    }

    public int nroDisciplinas() {
        return disciplinas.size();
    }

    public Disciplina getDisciplina(int i){
        return disciplinas.get(i);
    }
}
```

```
}  
}
```

- Controleacademico.dominio.matriculas:

- Historico.java:

```
package controleacademico.dominio.matriculas;  
import java.util.*;  
import controleacademico.dominio.pessoas.Aluno;  
import controleacademico.dominio.cursos.Disciplina;  
import controleacademico.dominio.matriculas.MatriculaAluno;  
  
public class Historico {  
    private ArrayList<MatriculaAluno> disciplinasMatriculadas;  
    private Aluno aluno;  
  
    public Historico (Aluno a) {  
        aluno = a;  
        disciplinasMatriculadas = new ArrayList<MatriculaAluno>();  
    }  
  
    public void matriculaDisciplina (OfertaTurma oferta) {  
        MatriculaAluno m = new MatriculaAluno (aluno, oferta);  
        this.disciplinasMatriculadas.add(m);  
        oferta.matriculaAluno(m);  
    }  
  
    public boolean constaAprovacao (Disciplina d) {  
        for (MatriculaAluno m: disciplinasMatriculadas) {  
            if (m.aprovado(d))  
                return true;  
        }  
        return false;  
    }  
  
    public void registraResultado (Disciplina d, int nota) {  
        for (MatriculaAluno m: disciplinasMatriculadas) {  
            if (m.getOferta().nomeDisciplina().equals(d.getNome()) &&  
m.getStatus() == StatusMatriculaDisciplina.CURSANDO)  
                m.registraNota(nota);  
        }  
    }  
}
```

- MatriculaAluno.java:

```
package controleacademico.dominio.matriculas;
```

```
import controleacademico.dominio.pessoas.Aluno;
```

```
import controleacademico.dominio.cursos.Disciplina;
```

```
public class MatriculaAluno {
```

```
    private Aluno aluno;
```

```
    private OfertaTurma oferta;
```

```
    private StatusMatriculaDisciplina status;
```

```
    private float nota;
```

```
    MatriculaAluno (Aluno a, OfertaTurma o) {
```

```
        aluno = a;
```

```
        oferta = o;
```

```
        status = StatusMatriculaDisciplina.CURSANDO;
```

```
    }
```

```
    Aluno alunoMatriculado () {
```

```
        return aluno;
```

```
    }
```

```
    OfertaTurma turmaMatriculada() {
```

```
        return oferta;
```

```
    }
```

```
    void registraNota (float n) {
```

```
        this.nota = n;
```

```
        if (this.nota >= 60)
```

```
            status = StatusMatriculaDisciplina.APROVADA;
```

```
        else
```

```
            status = StatusMatriculaDisciplina.REPROVADA;
```

```
    }
```

```
    void trancaMatricula () {
```

```
        this.status = StatusMatriculaDisciplina.TRANCADA;
```

```
    }
```

```
    boolean aprovado (Disciplina d) {
```

```
        return d.getNome().equals(oferta.nomeDisciplina()) && status ==
```

```
        StatusMatriculaDisciplina.APROVADA;
```

```
    }
```

```

    public OfertaTurma getOferta(){
        return oferta;
    }
    public StatusMatriculaDisciplina getStatus(){
        return status;
    }
}

```

○ **OfertaTurma:**

```

package controleacademico.dominio.matriculas;
import java.util.*;

```

```

import controleacademico.dominio.pessoas.Professor;
import controleacademico.dominio.cursos.Disciplina;
import controleacademico.dominio.pessoas.Aluno;

```

```

public class OfertaTurma {
    private int ano;
    private Semestre semestre;
    private Disciplina disciplina;
    private Vector<MatriculaAluno> alunosMatriculados;
    private Professor professor;

    public OfertaTurma (int ano, Semestre s, Disciplina d) {
        this.ano = ano;
        this.semestre = s;
        this.disciplina = d;
        alunosMatriculados = new Vector<MatriculaAluno>();
    }
    void matriculaAluno (MatriculaAluno m) {
        alunosMatriculados.add(m);
    }

    void cancelaMatricula (Aluno a) {
        alunosMatriculados.remove(a);
    }

    public void vinculaProfessor (Professor p) {
        professor = p;
        professor.ficaAlocadoTurma(this);
    }

    public String nomeDisciplina() {
        return this.disciplina.getNome();
    }
}

```

```
}  
}
```

- Semestre.java:

```
package controleacademico.dominio.matriculas;
```

```
public enum Semestre {  
    PRIMEIRO, SEGUNDO;  
}
```

- StatusMatriculaDisciplina.java:

```
package controleacademico.dominio.matriculas;
```

```
public enum StatusMatriculaDisciplina {  
    CURSANDO, TRANCADA, REPROVADA, APROVADA;  
}
```

- Controleacademico.dominio.pessoas:

- Aluno.java:

```
package controleacademico.dominio.pessoas;
```

```
import controleacademico.dominio.cursos.GradeCurricular;  
import controleacademico.dominio.matriculas.Historico;  
import controleacademico.dominio.matriculas.OfertaTurma;  
import controleacademico.dominio.cursos.Disciplina;
```

```
public class Aluno {  
    private String nome;  
    private GradeCurricular grade;  
    private Historico historico;
```

```
    public Aluno (String n, GradeCurricular g) {  
        nome = n;  
        grade = g;  
        historico = new Historico(this);  
    }
```

```
    public boolean verificaSeFormou () {  
        for (int i = 0; i < grade.nroDisciplinas(); i++) {  
            //for (Disciplina d: grade.disciplinas) {  
                if(! historico.constaAprovacao(grade.getDisciplina(i)))  
                    return false;  
            }  
        }  
        return true;  
    }
```

```

    }
    public String getNome(){
        return (nome);
    }
    public void matriculaAluno(OfertaTurma oferta){
        this.historico.matriculaDisciplina(oferta);
    }
    public void registraNota(Disciplina d, int nota){
        this.historico.registraResultado(d, nota);
    }
}

```

- Professor.java:

```

package controleacademico.dominio.pessoas;
import java.util.*;

import controleacademico.dominio.matriculas.OfertaTurma;

public class Professor {
    private String nome;
    private String departamento;
    private ArrayList<OfertaTurma> turmasOfertadas;

    public Professor (String nome, String d) {
        this.nome = nome;
        departamento = d;
    }

    public void ficaAlocadoTurma (OfertaTurma o) {
        turmasOfertadas.add(o);
        o.vinculaProfessor(this);
    }
}

```

- Controleacademico.dominio.ui:

- ControleAcademico.java:

```

package controleacademico.ui;
import javax.swing.*;
import controleacademico.dominio.cursos.Disciplina;
import controleacademico.dominio.cursos.*;
import controleacademico.dominio.matriculas.*;
import controleacademico.dominio.pessoas.*;

public class ControleAcademico {

```

```

public static void main(String[] args) {
    Disciplina pp = new Disciplina("PP");
    Disciplina aed1 = new Disciplina("AED1");
    Disciplina poo1 = new Disciplina("POO1");
    GradeCurricular grade = new GradeCurricular();
    grade.incluiDisciplina(pp);
    //pp.incluiNaGrade(grade); Seria um bad smell: feature envy
    grade.incluiDisciplina(aed1);
    grade.incluiDisciplina(poo1);
    Professor profMarcelo = new Professor("Marcelo", "FACOM");
    Aluno alunoSeverino = new Aluno("Severino", grade);

    JOptionPane.showMessageDialog(null,"O aluno " +
alunoSeverino.getNome()+ " esta' " +
(alunoSeverino.verificaSeFormou() ? "formado!" : "ainda cursando"));

    OfertaTurma ofertaPP = new OfertaTurma(2015,
Semestre.SEGUNDO, pp);
    alunoSeverino.matriculaAluno(ofertaPP);
    JOptionPane.showMessageDialog(null,"O aluno " +
alunoSeverino.getNome() + " esta' " +
(alunoSeverino.verificaSeFormou() ? "formado!" : "ainda cursando"));

    OfertaTurma ofertaAED1 = new OfertaTurma(2016,
Semestre.PRIMEIRO, aed1);
    alunoSeverino.matriculaAluno(ofertaAED1);

    OfertaTurma ofertaPOO1 = new OfertaTurma(2016,
Semestre.PRIMEIRO, poo1);
    alunoSeverino.matriculaAluno(ofertaPOO1);

    JOptionPane.showMessageDialog(null,"O aluno " +
alunoSeverino.getNome() + " esta' " +
(alunoSeverino.verificaSeFormou() ? "formado!" : "ainda cursando"));

    alunoSeverino.registraNota(pp, 60);
    JOptionPane.showMessageDialog(null,"O aluno " +
alunoSeverino.getNome() + " esta' " +
(alunoSeverino.verificaSeFormou() ? "formado!" : "ainda cursando"));
    alunoSeverino.registraNota(aed1, 60);
    JOptionPane.showMessageDialog(null,"O aluno " +
alunoSeverino.getNome() + " esta' " +
(alunoSeverino.verificaSeFormou() ? "formado!" : "ainda cursando"));

```



```

        alunoSeverino.registraNota(pool, 60);
        JOptionPane.showMessageDialog(null, "O aluno " +
        alunoSeverino.getNome() + " esta' " +
        (alunoSeverino.verificaSeFormou() ? "formado!" : "ainda cursando"));
    }
}

```

2. Print do funcionamento:



