



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

AD1 de Programação III

1º semestre de 2013

Nome:

Pólo:

Exercício (ENTREGAR OS ARQUIVOS EM MÍDIA, PARA FINS DE TESTE, JUNTAMENTE COM A AD IMPRESSA):

Considere que você é dono de uma empresa de software e que sua empresa foi contratada pela empresa educacional CEDEXY para escrever um software que fornece informações a respeito dos seus discentes matriculados.

Existem dois tipos de avaliações nesta entidade educacional: Avaliações a Distância (AD) e Avaliações Presenciais (AP). AD é uma avaliação digital e impressa, cujo o enunciado é entregue aos alunos com dois meses de antecedência, e com data de entrega fixada (pelo CEDEXY). Já a AP é uma avaliação em papel e individual, realizada com data e hora previamente marcada. Ambas possuem notas que quantificam o aprendizado do aluno. Se o discente não entrega a AD ou não faz a AP, a sua nota nesta avaliação é zero.

Durante o semestre o aluno faz as seguintes avaliações: AD_1 e AP_1 , AD_2 e AP_2 . Com a AD_1 e AP_1 calcula-se a média $N_1 = (2 \times AD_1 + 8 \times AP_1) / 10$. Com a AD_2 e AP_2 computa-se a média $N_2 = (2 \times AD_2 + 8 \times AP_2) / 10$. Com as médias N_1 e N_2 calcula-se a média $N = (N_1 + N_2) / 2$. Se N é maior ou igual a seis, o aluno está aprovado e esta é sua NOTA FINAL. Senão, o aluno faz a AP_3 (avaliação presencial com toda a matéria) e, para ser aprovado, $(\text{maior_nota_entre}\{N_1, N_2\} + AP_3) / 2$ deve ser maior ou igual a cinco.

Com base nestas informações, escreva:

(a) As classes avaliação, AP e AD, sendo a AP e a AD subclasses de avaliação;

(b) A classe disciplina que tem um nome, um código único que a distingue das demais e uma carga horária;

(c) A classe aluno que tem um nome, uma matrícula única que o distingue dos demais e um CR. Além disto, cada aluno pode se inscrever em, no máximo, cinco disciplinas. As notas estão relacionadas as disciplinas cursadas; e

(d) Escreva um programa de teste que crie alunos e disciplinas. Seu programa deve informar o CR e em quais matérias o aluno foi aprovado ou reprovado (com as respectivas notas). O cálculo de CR será feito com base na carga horária do curso, obedecendo a seguinte fórmula: $CR = [(CH_1 \times N_1) + (CH_2 \times N_2) + \dots + (CH_n \times N_n)] / (CH_1 + CH_2 + \dots + CH_n)$, onde:

n = número de disciplinas cursadas;

CH_i = carga horária da disciplina; e

N_i = nota final da disciplina, $i = 1, 2, \dots n$.

RESPOSTA:

```
import java.util.*;
import java.text.DecimalFormat;

abstract class Avaliacao{
    String nome_aval;
    GregorianCalendar data;
    double nota;

    Avaliacao(String na, GregorianCalendar d){
        nome_aval = na;
        data = d;
        nota = 0.0;
    }

    double getNota(){ return nota; }

    void setNota(double n){ nota = n; }

    public String toString(){
        DecimalFormat df = new DecimalFormat("0.00");
        return "\t\t" + nome_aval + ": " + df.format(nota) + "\n";
    }
}

class AvalDist extends Avaliacao{
    AvalDist(String n, GregorianCalendar d){ super(n, d); }
}

class AvalPres extends Avaliacao{
    AvalPres(String n, GregorianCalendar d){ super(n, d); }
}

class Disciplina{
    String nome_disc;
    static int prox_codigo = 1;
    int codigo;
    int CH;
    Avaliacao provas[];
    double nota_final;
    String sit;

    Disciplina(String n, int h){
        nome_disc = n;
        codigo = prox_codigo++;
        CH = h;
        provas = new Avaliacao[5];
    }
}
```

```

    Disciplina(String n, int h, Avaliacao ad1, Avaliacao ap1, Avaliacao
ad2, Avaliacao ap2, Avaliacao ap3){
        this(n, h);
        provas[0] = ad1;
        provas[1] = ap1;
        provas[2] = ad2;
        provas[3] = ap2;
        provas[4] = ap3;
    }

    void insereNota(Avaliacao a, double nota){
        for(int i = 0; i < provas.length; i++){
            if(a == provas[i]){
                provas[i].setNota(nota);
                return;
            }
        }
    }

    void calculaNotaFinal(){
        double N1, N2, NF;
        N1 = ((provas[0].getNota() * 2) + (provas[1].getNota() * 8)) / 10.0;
        N2 = ((provas[2].getNota() * 2) + (provas[3].getNota() * 8)) / 10.0;
        nota_final = (N1 + N2) / 2.0;
        if(nota_final >= 6.0){
            sit = "aprovado";
            return;
        }
        if(N1 > N2) nota_final = (N1 + provas[4].getNota()) / 2.0;
        else nota_final = (N2 + provas[4].getNota()) / 2.0;
        if(nota_final >= 5.0) sit = "aprovado";
        else sit = "reprovado";
    }

    double getNotaFinal(){ return nota_final; }

    int getCH(){ return CH; }

    public String toString(){
        DecimalFormat df = new DecimalFormat("0.00");
        String resp = "\t" + nome_disc + " (codigo: " + codigo + " - " + CH
+ "h): " + df.format(nota_final) + " (" + sit + ")" + "\n";
        for(int i = 0; i < provas.length; i++) resp += provas[i].toString();
        return resp;
    }
}

class Discente{
    String nome;
    int mat;
    static int prox_mat = 1;
    double CR;
    int disc_livre;
    Disciplina disc[];
}

```

```

Discente(String n){
    nome = n;
    mat = prox_mat++;
    disc_livre = 0;
    disc = new Disciplina[5];
    CR = 0.0;
}

void insereDisc(Disciplina d){
    if(disc_livre == disc.length) return;
    disc[disc_livre++] = d;
}

void insereNotaDiscente(Disciplina d, Avaliacao a, double nota){
    for(int i = 0; i < disc_livre; i++){
        if(disc[i] == d){
            disc[i].insereNota(a, nota);
            return;
        }
    }
}

void calculaCR(){
    int totalCH = 0;
    for(int i = 0; i < disc_livre; i++){
        disc[i].calculaNotaFinal();
        CR += disc[i].getNotaFinal() * disc[i].getCH();
        totalCH += disc[i].getCH();
    }
    CR /= totalCH;
}

public String toString(){
    DecimalFormat df = new DecimalFormat("0.00");
    String resp = nome + " (" + mat + "): " + df.format(CR) + "\n";
    for(int i = 0; i < disc_livre; i++){
        resp += disc[i].toString();
    }
    return resp;
}

}

public class AD1_2013_1{
    public static void main(String[] args){
        GregorianCalendar GCAD1 = new GregorianCalendar(2013, 3, 11);
        GregorianCalendar GCAP1 = new GregorianCalendar(2013, 3, 11, 10,
59);
        GregorianCalendar GCAD2 = new GregorianCalendar(2013, 3, 11);
        GregorianCalendar GCAP2 = new GregorianCalendar(2013, 3, 11, 10,
59);
        GregorianCalendar GCAP3 = new GregorianCalendar(2013, 3, 11, 10,
59);

        Avaliacao AD1_ProgIII = new AvalDist("AD1", GCAD1);
        Avaliacao AP1_ProgIII = new AvalPres("AP1", GCAP1);
        Avaliacao AD2_ProgIII = new AvalDist("AD2", GCAD2);
        Avaliacao AP2_ProgIII = new AvalPres("AP2", GCAP2);
    }
}

```

```

    Avaliacao AP3_ProgIII = new AvalPres("AP3", GCAP3);
    Disciplina ProgIII = new Disciplina("Programacao III", 80,
AD1_ProgIII, AP1_ProgIII, AD2_ProgIII, AP2_ProgIII, AP3_ProgIII);

    Avaliacao AD1_APSI = new AvalDist("AD1", GCAD1);
    Avaliacao AP1_APSI = new AvalPres("AP1", GCAP1);
    Avaliacao AD2_APSI = new AvalDist("AD2", GCAD2);
    Avaliacao AP2_APSI = new AvalPres("AP2", GCAP2);
    Avaliacao AP3_APSI = new AvalPres("AP3", GCAP3);
    Disciplina APSI = new Disciplina("Arquitetura e Projeto de Sistemas
I", 80, AD1_APSI, AP1_APSI, AD2_APSI, AP2_APSI, AP3_APSI);

    Avaliacao AD1_ES = new AvalDist("AD1", GCAD1);
    Avaliacao AP1_ES = new AvalPres("AP1", GCAP1);
    Avaliacao AD2_ES = new AvalDist("AD2", GCAD2);
    Avaliacao AP2_ES = new AvalPres("AP2", GCAP2);
    Avaliacao AP3_ES = new AvalPres("AP3", GCAP3);
    Disciplina ES = new Disciplina("Engenharia de Software", 80,
AD1_ES, AP1_ES, AD2_ES, AP2_ES, AP3_ES);

    Avaliacao AD1_RCI = new AvalDist("AD1", GCAD1);
    Avaliacao AP1_RCI = new AvalPres("AP1", GCAP1);
    Avaliacao AD2_RCI = new AvalDist("AD2", GCAD2);
    Avaliacao AP2_RCI = new AvalPres("AP2", GCAP2);
    Avaliacao AP3_RCI = new AvalPres("AP3", GCAP3);
    Disciplina RCI = new Disciplina("Redes de Computadores I", 120,
AD1_RCI, AP1_RCI, AD2_RCI, AP2_RCI, AP3_RCI);

    Avaliacao AD1_APSII = new AvalDist("AD1", GCAD1);
    Avaliacao AP1_APSII = new AvalPres("AP1", GCAP1);
    Avaliacao AD2_APSII = new AvalDist("AD2", GCAD2);
    Avaliacao AP2_APSII = new AvalPres("AP2", GCAP2);
    Avaliacao AP3_APSII = new AvalPres("AP3", GCAP3);
    Disciplina APSII = new Disciplina("Arquitetura e Projeto de
Sistemas II", 80, AD1_APSII, AP1_APSII, AD2_APSII, AP2_APSII,
AP3_APSII);

    Avaliacao AD1_CG = new AvalDist("AD1", GCAD1);
    Avaliacao AP1_CG = new AvalPres("AP1", GCAP1);
    Avaliacao AD2_CG = new AvalDist("AD2", GCAD2);
    Avaliacao AP2_CG = new AvalPres("AP2", GCAP2);
    Avaliacao AP3_CG = new AvalPres("AP3", GCAP3);
    Disciplina CG = new Disciplina("Computacao Grafica", 80, AD1_CG,
AP1_CG, AD2_CG, AP2_CG, AP3_CG);

    Avaliacao AD1_RCII = new AvalDist("AD1", GCAD1);
    Avaliacao AP1_RCII = new AvalPres("AP1", GCAP1);
    Avaliacao AD2_RCII = new AvalDist("AD2", GCAD2);
    Avaliacao AP2_RCII = new AvalPres("AP2", GCAP2);
    Avaliacao AP3_RCII = new AvalPres("AP3", GCAP3);
    Disciplina RCII = new Disciplina("Redes de Computadores II", 120,
AD1_RCII, AP1_RCII, AD2_RCII, AP2_RCII, AP3_RCII);

    int i;
    double nota;

```

```

Discente jose = new Discente("Jose das Couves");
jose.inserDisc(ProgIII);
nota = Math.random() * 10;
jose.inserNotaDiscente(ProgIII, AD1_ProgIII, nota);
nota = Math.random() * 10;
jose.inserNotaDiscente(ProgIII, AP1_ProgIII, nota);
nota = Math.random() * 10;
jose.inserNotaDiscente(ProgIII, AD2_ProgIII, nota);
nota = Math.random() * 10;
jose.inserNotaDiscente(ProgIII, AP2_ProgIII, nota);
nota = Math.random() * 10;
jose.inserNotaDiscente(ProgIII, AP3_ProgIII, nota);

jose.inserDisc(APSI);
nota = Math.random() * 10;
jose.inserNotaDiscente(APSI, AD1_APSI, nota);
nota = Math.random() * 10;
jose.inserNotaDiscente(APSI, AP1_APSI, nota);
nota = Math.random() * 10;
jose.inserNotaDiscente(APSI, AD2_APSI, nota);
nota = Math.random() * 10;
jose.inserNotaDiscente(APSI, AP2_APSI, nota);
nota = Math.random() * 10;
jose.inserNotaDiscente(APSI, AP3_APSI, nota);

jose.inserDisc(ES);
nota = Math.random() * 10;
jose.inserNotaDiscente(ES, AD1_ES, nota);
nota = Math.random() * 10;
jose.inserNotaDiscente(ES, AP1_ES, nota);
nota = Math.random() * 10;
jose.inserNotaDiscente(ES, AD2_ES, nota);
nota = Math.random() * 10;
jose.inserNotaDiscente(ES, AP2_ES, nota);
nota = Math.random() * 10;
jose.inserNotaDiscente(ES, AP3_ES, nota);

jose.calculaCR();
System.out.println(jose);

Discente sicrano = new Discente("Sicrano da Silva");
sicrano.inserDisc(APSII);
nota = Math.random() * 10;
sicrano.inserNotaDiscente(APSII, AD1_APSII, nota);
nota = Math.random() * 10;
sicrano.inserNotaDiscente(APSII, AP1_APSII, nota);
nota = Math.random() * 10;
sicrano.inserNotaDiscente(APSII, AD2_APSII, nota);
nota = Math.random() * 10;
sicrano.inserNotaDiscente(APSII, AP2_APSII, nota);
nota = Math.random() * 10;
sicrano.inserNotaDiscente(APSII, AP3_APSII, nota);

```

```
sicrano.inserDisc(CG);
nota = Math.random() * 10;
sicrano.inserNotaDiscente(CG, AD1_CG, nota);
nota = Math.random() * 10;
sicrano.inserNotaDiscente(CG, AP1_CG, nota);
nota = Math.random() * 10;
sicrano.inserNotaDiscente(CG, AD2_CG, nota);
nota = Math.random() * 10;
sicrano.inserNotaDiscente(CG, AP2_CG, nota);
nota = Math.random() * 10;
sicrano.inserNotaDiscente(CG, AP3_CG, nota);

sicrano.calculaCR();
System.out.println(sicrano);
}
}
```