



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

AD2 de Programação III

1º semestre de 2015.

Nome:

Matrícula:

Pólo:

Obs: A solução para o exercício proposto deve ser entregue em formato digital.

Considere a questão 1 da AP1 deste semestre, a qual fornecia as 2 classes abaixo:

Representa o total de despesas num mês	Representa o total de despesas num dia
<pre>class DespesaMes { private int mes; // mes da despesa private float valor; // valor da despesa public DespesaMes(int mes, float valor) { this.mes = mes; this.valor = valor; } public int getMes() { return mes; } public float getValor() { return valor; } }</pre>	<pre>class DespesaDia extends DespesaMes { private int dia; // dia da despesa public DespesaDia(int dia, int mes, float valor) { super(mes, valor); this.dia = dia; } public int getDia() { return dia; } }</pre>

e pedia que criássemos uma terceira classe, a qual deveria conter, dentre outras coisas, um vetor contendo as despesas realizadas e um método que realizasse a soma das despesas. Do gabarito, uma possível solução seria a classe abaixo:

<pre>class Despesas { DespesaMes despesas []; String cpf; public Despesas(DespesaMes[] despesas, String cpf) { this.despesas = despesas; this.cpf = cpf; } public String getCpf () { return this.cpf; } }</pre>

```

    public DespesaMes totalizaMes (int mes) {
        float valor = 0;
        for (int i=0; i<despesas.length; i++)
        {
            if (despesas[i].getMes() == mes)
                valor = valor + despesas[i].getValor();
        }
        return new DespesaMes(mes, valor);
    }
}

```

Considerando estas classes, faça:

- Crie um método toString na classe DespesaMes, o qual retorna uma versão string da despesa. Retorne uma string com formato baseado no seguinte exemplo: “Mes: 2, Valor: 1500.0”
- Crie um método main numa classe nova, chamada Principal; neste método, crie um vetor do tipo DespesaMes que contenha as seguintes despesas:

Dia	Mês	Valor
-	2	1000
10	2	500
10	3	500
-	4	750

- Crie, ainda no método main, um objeto do tipo Despesas e imprima o total para o mês 2.
- Crie uma classe DespesaAntiga. Suponha que, para estas, não há registro de data, apenas do valor do que foi pago e alguma observação sobre o uso. Permita que objetos deste tipo possam também ser inseridos no vetor criado no item b). Deve ser evitado que relações de herança estranhas, incorretas do ponto de vista de modelagem, aconteçam, como uma classe que herda campos de outra mas nunca os utiliza. Faça as adequações necessárias, podendo mexer nas classes, para que isso funcione.

RESPOSTA:

```

abstract class Despesa {
    private float valor;

    public Despesa(float valor) {
        this.valor = valor;
    }

    public float getValor() {
        return valor;
    }
}

```

```

        abstract public int getMes();
    }

    class DespesaMes extends Despesa {
        private int mes;
        // mes da despesa

        // valor da despesa
        public DespesaMes(int mes, float valor) {
            super(valor);
            this.mes = mes;
        }

        public int getMes() {
            return mes;
        }

        // item a)
        public String toString() {
            return "Mes: " + this.mes + " Valor: " + this.getValor() + "\n";
        }
    }

    class DespesaDia extends DespesaMes {
        private int dia;

        // dia da despesa
        public DespesaDia(int dia, int mes, float valor) {
            super(mes, valor);
            this.dia = dia;
        }

        public int getDia() {
            return dia;
        }
    }

    class Despesas {
        Despesa despesas [];
        String cpf;

        public Despesas(Despesa[] despesas, String cpf) {
            this.despesas = despesas;
            this.cpf = cpf;
        }

        public String getCpf () {
            return this.cpf;
        }

        public DespesaMes totalizaMes (int mes) {
            float valor = 0;
            for (int i=0; i<despesas.length; i++)
            {
                if (despesas[i].getMes() == mes)
                    valor = valor + despesas[i].getValor();
            }
            return new DespesaMes(mes, valor);
        }
    }

```

```

// item d)

class DespesaAntiga extends Despesa {
    String observacao;

    public DespesaAntiga (float valor, String obs) {
        super(valor);
        this.observacao = obs;
    }

    public int getMes() {
        return -1;
    }
}

public class AD2_2015_1 {
    public static void main(String[] args) {

        // item b)
        Despesa desp[] = { new DespesaMes(2, 1000), new DespesaDia(10,2,500), new
DespesaDia(10,3,500), new DespesaMes(4,750) };

        // item c)
        Despesas totalmes = new Despesas(desp, "001001001-01");
        System.out.println(totalmes.totalizaMes(2));
    }
}

```