



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Programação Orientada a Objetos

AP1 1º semestre de 2018.

Nome –

Assinatura –

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
 2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
 3. Você pode usar lápis para responder as questões.
 4. Ao final da prova devolva as folhas de questões e as de respostas.
 5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.
-

Questão 1) (4.0 pontos)

Você foi chamado pela empresa XXX para fazer um software que realiza conversões em dados inteiros de datas informadas como parâmetro de entrada. Seu programa recebe, na linha de comando, a data de um país e uma informação, que só pode ser BRA, ING e BIN, e a transforma em um número inteiro. Sabendo-se que o ano tem 365 dias, e que cada mês possui, em média, 30 dias, o inteiro obtido na resposta é o número de dias decorridos desde o início do calendário usado no Brasil (Calendário Gregoriano). O formato Binário (BIN) e de data no Brasil (BRA) é dia/mês/ano. Já a data na Inglaterra (ING) é dada por mês/dia/ano.

Além disso, em BRA e ING as datas seguem a representação decimal. Já em BIN, as datas seguem a representação binária. Exemplos de entrada e de saída para auxiliar na resolução da questão são mostrados a seguir:

ENTRADA	SAÍDA
1/10/1000 BRA	365301 dias
10/1/1000 ING	365301 dias
11/1100/1 BIN	728 dias

DICA: Utilize o conceito de herança. Não é permitido o uso de classes relacionadas a DATAS de Java, e de classes que realizam interface com o usuário.

RESPOSTA:

```
class Data{
    int dia, mes, ano;

    Data(){}

    Data(int dd, int mm, int aa){
        dia = dd;
        mes = mm;
        ano = aa;
    }

    int converteDataDias(){ return 365 * ano + 30 * mes + dia; }

    public String toString(){ return converteDataDias() + " dias"; }
}

class DataBra extends Data{
    DataBra(int dd, int mm, int aa){ super(dd, mm, aa); }
}

class DataIng extends Data{
    DataIng(int mm, int dd, int aa){ super(dd, mm, aa); }
}

class DataBin extends Data{
    DataBin(int dd, int mm, int aa){
        super();
        dia = converteBinNum(dd);
        mes = converteBinNum(mm);
        ano = converteBinNum(aa);
    }

    int converteBinNum(int bin){
        String str = "" + bin;
        int base2 = 1, resp = 0, i = 0;
        while (i < str.length()){
            resp = resp + (base2 * (-48 + str.charAt(str.length() - i -
1)));
            base2 *= 2;
            i++;
        }
        return resp;
    }
}

public class Q1_AP1_POO_2018_1{
    public static void main(String[] args){
        String vs[] = args[0].split("/");
        Data data = null;
        if(args[1].equals("BRA"))
            data = new DataBra(Integer.parseInt(vs[0]),
Integer.parseInt(vs[1]), Integer.parseInt(vs[2]));
        else if(args[1].equals("ING"))
            data = new DataIng(Integer.parseInt(vs[0]),
Integer.parseInt(vs[1]), Integer.parseInt(vs[2]));
        else if(args[1].equals("BIN"))
            data = new DataBin(Integer.parseInt(vs[0]),
Integer.parseInt(vs[1]), Integer.parseInt(vs[2]));
        System.out.println(data);
    }
}
```

}

Questão 2) (3.0 pontos)

Considere o programa inicial abaixo para manipular corridas de táxi (convencional e de aplicativos privados).

```
class Corrida {  
    int distancia;  
  
    public Corrida (int distancia) {  
        this.distancia = distancia;  
    }  
  
    public double calculaValor() {  
        return this.distancia * 2.5;  
    }  
}
```

Na classe Corrida, o campo distancia representa qual a distância percorrida numa corrida, a qual é utilizada para seu cálculo padrão (multiplicação pela taxa de quilômetro percorrido: 2.5). Táxis convencionais adicionam uma taxa inicial (bandeirada) com valor de 5.5. Essa taxa é única entre os táxis convencionais, ou seja, seu valor é constante. Além disso, estes têm a informação da bandeira, a qual pode ter valor 1 (dias normais) ou 2 (períodos noturnos, fins de semana e feriados). Quando a bandeira tem valor 2, a taxa de quilômetro percorrido sobe para 3.

Corridas de aplicativos, por sua vez, tem seu cálculo baseado em outras informações: quantidade de motoristas disponíveis (acima de 5, 20% a menos que o valor padrão – Multiplicação por 0.8); se a corrida pode ser compartilhada com outros clientes (10% a menos); se o veículo requerido é especial (10% a mais), e; condição do trânsito: Bom (10% a menos), Razoável (sem alteração em relação ao valor padrão) e Ruim (aumento de 10%).

Conclua a implementação. A ordem como os percentuais são aplicados, os quais podem causar alguma diferença, são irrelevantes para esta questão. Além disso, crie objetos para que possamos comparar o valor das corridas com distância de 30 entre os 2 tipos de transporte. As corridas ocorrem durante o dia, com trânsito bom, com carros comuns, de forma exclusiva (não compartilhada) e com boa oferta de motoristas (mais que 10).

RESPOSTA:

```
class Corrida {  
    int distancia;  
  
    public Corrida (int distancia) {  
        this.distancia = distancia;  
    }  
  
    public double calculaValor() {  
        return this.distancia * 2.5;  
    }  
}  
  
class TaxiComum extends Corrida {
```

```

    int bandeira; // 1 ou 2
    static double bandeirada = 5.5;

    public TaxiComum (int distancia, int bandeira) {
        super(distancia);
        this.bandeira = bandeira;
    }

    public double calculaValor() {
        if (this.bandeira == 1)
            return TaxiComum.bandeirada + super.calculaValor();
        return TaxiComum.bandeirada + (this.distancia * 3);
    }
}

class AplicativoTransporte extends Corrida {
    int motoristasDisponiveis;
    boolean compartilhado;
    boolean especial;
    String transito; // Bom, Razoável ou Ruim

    public AplicativoTransporte (int distancia, int motoristasDisponiveis,
    boolean compartilhado, boolean especial, String transito) {
        super(distancia);
        this.motoristasDisponiveis = motoristasDisponiveis;
        this.compartilhado = compartilhado;
        this.especial = especial;
        this.transito = transito;
    }

    public double calculaValor() {
        double percentual = 1;
        if (this.motoristasDisponiveis > 5)
            percentual-=0.2;
        if (this.compartilhado)
            percentual-=0.1;
        if (this.especial)
            percentual+=0.1;
        if (this.transito == "Bom")
            percentual-=0.1;
        else
            if (this.transito == "Ruim")
                percentual+=0.1;
        return super.calculaValor() * percentual;
    }
}

public class AP1_2018_2_Q2 {
    public static void main(String[] args) {
        AplicativoTransporte c1 = new AplicativoTransporte(30, 40, false,
        false, "Bom");
        System.out.println(c1.calculaValor());
        Corrida c2 = new Corrida(30);
        System.out.println(c2.calculaValor());
        TaxiComum c3 = new TaxiComum(30, 1);
        System.out.println(c3.calculaValor());
    }
}

```

```
}  
}
```

Questão 3) (3.0 pontos)

Crie uma classe `Lampada` que possui um atributo `ligada`, o qual indica se a lâmpada está ligada ou desligada. Ao construir uma lâmpada, o estado inicial dela é desligada. Para ligar e desligar a lâmpada, os métodos `ligar()` e `desligar()` devem ser chamados, respectivamente. Aliás, esta é a única forma de alterar o estado da lâmpada, já que o atributo `ligada` não pode ser visível fora da classe. Redefina o método `toString()`, da classe `Object`, de forma que a sua chamada retorne “Lâmpada ligada” ou “Lâmpada desligada”, dependendo do seu estado atual. A lâmpada também deve possuir um método `imprime()` que, quando chamado, exibe o retorno do método `toString()`.

Construa uma aplicação que cria uma lâmpada desligada, imprima seu estado, muda o estado dela para ligada e novamente imprime o seu estado atual.

RESPOSTA:

```
class Lampada {  
    private boolean ligada;  
  
    public Lampada () {  
        this.ligada = false;  
    }  
  
    public void ligar() {  
        this.ligada = true;  
    }  
  
    public void desligar() {  
        this.ligada = false;  
    }  
  
    public String toString() {  
        return (this.ligada) ? "Lâmpada ligada" : "Lâmpada desligada";  
    }  
  
    public void imprime() {  
        System.out.println(this);  
    }  
}  
  
public class AP1_2018_2_Q3 {  
    public static void main(String[] args) {  
        Lampada l1 = new Lampada();  
        l1.imprime();  
        l1.ligar();  
        l1.imprime();  
    }  
}
```