



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Programação III

AP1 1º semestre de 2016.

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)

O Cadastro de Pessoas Físicas (CPF) é o registro de um cidadão na Receita Federal brasileira no qual devem estar todos os contribuintes, ou seja, pessoas físicas nacionais e estrangeiras com negócios no Brasil. Ao ser emitido, um CPF gera um número de onze algarismos no formato:

ABC.DEF.GHI-JK

sendo os dois últimos dígitos (*J* e *K*) verificadores para evitar erros de digitação.

Um “algoritmo de módulo 11” é adotado para obter os dígitos de verificação:

$$r = (s \bmod 11) \bmod 10, \quad \text{para } s = \sum_{i=1}^9 (i \cdot d_i)$$

onde $(a \bmod b)$ denota o resto (inteiro) após a divisão de a por b , $(a \cdot b)$ denota o produto de a por b e d_i é o i -ésimo dígito considerado no cálculo do dígito resultante r . No caso do cálculo de J , $d_1 = A$, $d_2 = B$, ..., $d_9 = I$. No cálculo de K , $d_1 = B$, $d_2 = C$, e assim por diante.

Nesta questão:

(3.0 pontos) Escreva a classe CPF, cuja instância representa um número de CPF válido. A classe deve conter:

- **(0.5 ponto)** Um ÚNICO atributo do tipo String que armazena um número de CPF válido no formato *XXX.XXX.XXX-XX*, onde *X* corresponde a um caractere numérico
- **(1.0 ponto)** Um construtor de inicialização que espera receber um valor inteiro e, se este CPF for válido, ele deve ser armazenado no formato *XXX.XXX.XXX-XX*.
- **(0.5 ponto)** O método `toString()` sobrescrito para retornar o número de CPF representado, isto é no formato *XXX.XXX.XXX-XX*.
- **(1.0 ponto)** Um método que verifique a tentativa de instanciação de um número de CPF inválido.

(1.0 ponto) Escreva uma classe que teste sua classe CPF. Nessa classe, o método *main* receberá, como parâmetro de entrada, um CPF, no formato numérico, criará instância de CPF, se este número for válido, e imprimirá o resultado.

RESPOSTA:

```
class CPF {
    String numero = "";
    public CPF(String numero){
        if ((numero == null) || (numero.length() != 11)){
            this.numero = null;
            return;
        }
        int[] d = new int[11];
        char[] resp = new char[14];
        for (int j = 0, i = 0; j < numero.length(); ) {
            if((i == 3) || (i == 7)) resp[i++] = '.';
            else if (i == 11) resp[i++] = '-';
            else{
                char c = numero.charAt(j);
                if ((c < '0') || (c > '9')){
                    numero = null;
                    return;
                }
                else{
                    resp[i++] = c;
                    d[j++] = (int)c - '0';
                }
            }
        }
    }
    if ((modulo11(d, 0) != d[9]) || (modulo11(d, 1) != d[10])) {
        this.numero = null;
        return;
    }
}
```

```

    }
    for(int i = 0; i < 14; i++) this.numero += resp[i];
}

public String toString() {
    return "O CPF valido e " + this.numero;
}

private int modulo11(int[] d, int primeiro) {
    int s = 0;
    for (int i = 1, j = primeiro; i <= 9; i++, j++) {
        s += i * d[j];
    }
    return ((s % 11) % 10);
}

public String getNumero(){ return numero; }
}

public class Q1_AP1_2016_1{
    public static void main(String[] args){
        CPF cpf = new CPF(args[0]);
        if(cpf.getNumero() != null) System.out.println(cpf);
    }
}

```

Questão 2) (3.0 pontos)

Considere a questão da AD1-2016.1 que pedia para simular o comportamento de uma pilha usando vetores. Um exemplo resumido da classe Pilha podia ser esse:

```

class Pilha {
    public Pilha() {
        // Inicialização da pilha
    }

    public boolean empty () { ... }

    public void push (int v) { ... }

    public int pop () { ... }
}

```

Implemente nesta classe (observe que não é necessário saber a estrutura interna que simula a pilha):

- Um método `push(int v[], int qtd)`, o qual recebe um vetor de números (`v`) e o tamanho do vetor (`qtd`) e empilha todos os valores na pilha.

- b. Um método `push(Pilha p)`, o qual recebe um vetor de números (`v`) e o tamanho do vetor (`qtd`) e empilha todos os valores na pilha.

RESPOSTA:

```
class Pilha {
    ...
    public void push (int v[], int qtd) {
        for (int i=0; i<qtd; i++) {
            push(v[i]);
        }
    }

    public void push (Pilha p) {
        while (! p.empty()) {
            this.push(p.pop());
        }
    }
}
```

Questão 3) (3.0 pontos)

Considere o código abaixo:

```
class Entrada {
    private String fonte;
    private String traducao;

    public Entrada(String fonte, String traducao) {
        this.fonte = fonte;
        this.traducao = traducao;
    }

    public String getFonte () {
        return this.fonte;
    }

    public String getTraducao () {
        return this.traducao;
    }
}

public class AP1_2016_1_Q3 {
    public static void main(String[] args) {
        Dicionario dic = new Dicionario();
        dic.adiciona(new Entrada("table", "mesa"));
        dic.adiciona(new Entrada("pen", "caneta"));
        dic.adiciona(new Entrada("car", "carro"));
        dic.adiciona(new Entrada("dog", "cachorro"));

        System.out.println(dic.traduz("pen"));
    }
}
```

Implemente a classe Dicionario, considerando a maneira como esta é utilizada no método main() acima.

RESPOSTA:

```
class Dicionario {
    private Entrada entradas[];
    private int tam;
    private static int MAX = 1000;

    public Dicionario() {
        entradas = new Entrada [MAX];
        tam = 0;
    }

    public void adiciona (Entrada e) {
        if (tam < MAX-1) {
            entradas[tam] = e;
            tam++;
        }
    }

    public String traduz (String termo) {
        for (int i=0; i<tam; i++)
            if (entradas[i].getFonte() == termo) {
                return entradas[i].getTraducao();
            }
        return "";
    }
}
```