

# Programação Visual

TIC - 3PA

## Lista de Exercícios - 02



**Professor:** Edwar Saliba Júnior

- 1) Elabore um programa para simular uma conta de poupança. Crie uma classe *conta-corrente* com os seguintes atributos: *saldo*, *juros* e *correção monetária*, todos do tipo *float*. Crie um método para:
  - depositar,
  - sacar e
  - aniversário.O método *depositar* deve aumentar o valor do saldo.  
O método *sacar* deve diminuir o valor do saldo desde que haja quantidade de dinheiro suficiente para o saque que está sendo solicitado. Caso não haja dinheiro suficiente para o saque solicitado, então emita uma mensagem de advertência ao usuário apresentando o valor do saldo atual.  
O método *aniversário*, quando chamado, deve aumentar o valor do saldo aplicando sobre este o percentual de juros e o percentual de correção monetária. Os percentuais de juros e correção monetária serão definidos pelos usuários do *software*.  
Na classe principal deverá ser criado um *menu* com as três opções possíveis de ações a serem executadas pelo usuário: depósito (que deve receber o valor a ser depositado), saque (que deve receber o valor a ser sacado) e aniversário (deve receber o percentual juros a ser aplicado sobre o saldo, o valor da correção monetária é fixo 0.2% ao mês).
- 2) Elabore um programa onde o usuário seja capaz de cadastrar 100 números inteiros. Após digitados os 100 números, mostre:
  - o somatório de todos os números ímpares existentes no vetor;
  - o somatório dos 50 primeiros números do vetor;
  - o total da soma dos 50 primeiros números menos o somatório dos últimos 50 números.
- 3) Construa um *software* em que o usuário seja capaz de armazenar 30 números num vetor (*v1*). Construa um segundo vetor (*v2*) de mesmo tamanho e copie os número digitados no primeiro vetor para este segundo vetor, porém, a ordem inversa à da digitação. Depois faça o seguinte:
  - construa um terceiro vetor (*v3*) que deverá receber em suas posições a seguinte multiplicação:  $v3[x] = v1[x] * v2[x]$
  - some todos os valores que estiverem em posições ímpares de *v3* e imprima o resultado;
  - some todos os valores ímpares de *v3* e imprima o resultado.
- 4) Elabore um programa para cadastro de 10 alunos. A classe "Aluno" deverá ser composta pelos seguintes atributos: matrícula (*int*), nome (*String*) e telefone (*String*). Na classe principal construa um vetor para o cadastro dos alunos e uma estrutura de *menu* onde o usuário do *software* possa escolher entre as seguintes opções: cadastrar, alterar, excluir, consultar, relatório e sair do programa.
- 5) Escreva um *software* capaz de cadastrar 100 veículos. Para isto, crie uma classe com o nome de "Veiculo" com os seguintes atributos: marca (*String*), modelo

(*String*), ano de fabricação (*int*) e ano do modelo (*int*). Na classe principal, crie um vetor para armazenar os veículos e uma estrutura de *menu* que permita ao usuário do *software* fazer as seguintes operações: cadastro, alteração, exclusão, consulta, relatório e sair do programa. Antes de excluir um valor ou objeto, deve-se sempre solicitar do usuário uma confirmação.

6) Faça um programa para controle de uma concessionária. Este programa deverá ser capaz de cadastrar 100 veículos, 100 clientes e 100 vendas. A classe "Veiculo" deverá possuir os seguintes atributos: marca, modelo, ano de fabricação e ano de modelo. A classe "Cliente" deverá possuir os seguintes atributos: nome, cpf, identidade e endereço. E a classe "Venda" terá os atributos: "Cliente", "Veiculo", valor e data da venda. Crie uma estrutura de *menu* onde o usuário possa: incluir, alterar, excluir, consultar e imprimir relatório das classes acima citadas. Lembre-se de criar um vetor para cada uma das classes e de instanciar os objetos antes que estes sejam colocados no vetor.

7) Indique o que será impresso pelo programa abaixo:

```
package pvt_exerciciodepuracao;
```

```
public class Cachorro {

    private String nome;
    private String raca; // Raça.
    private boolean pedigree;
    private int idade;

    public Cachorro(String nome, String raca, boolean ped, int idade){
        this.nome = nome;
        this.raca = raca;
        this.pedigree = ped;
        this.idade = idade;
    }

    public void alterar((String nome, String raca, boolean ped, int idade){
        this.nome = nome;
        this.raca = raca;
        this.pedigree = ped;
        this.idade = idade;
    }

    public void consultar(){
        System.out.println("----==[Dados do Cão]====");
        System.out.println("Nome      : " + nome);
        System.out.println("Raça      : " + raca);
        System.out.println("Pedigre:  : " + pedigree);
        System.out.println("Idade    : " + idade);
    }

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

    public void setIdade(int id){
        idade = id;
    }

    public int getIdade(){
```

```
        return(idade);
    }
}
```

```
=====
package pvt_exerciciodepuracao;
```

```
public class PVT_ExercicioDepuracao {
```

```
    public static void main(String[] args) {
```

```
        int i, x;
```

```
        Cachorro caes[];
```

```
        caes = new Cachorro[5];
```

```
        caes[0] = new Cachorro("Rex", "Pastor Alemão", false, 2);
```

```
        caes[1] = new Cachorro("Tica", "Vira Latas", false, 2);
```

```
        caes[2] = new Cachorro("Iang", "Chau Chau", true, 1);
```

```
        caes[3] = new Cachorro("Fran", "Vira Latas", false, 2);
```

```
        caes[4] = new Cachorro("Malú", "Labrador", false, 2);
```

```
        for(i = 0; i < caes.length; i++){
```

```
            System.out.println(caes[i].getNome() + " " + caes[i].getIdade());
```

```
        }
```

```
        caes[0].alterar("Lilica", "Vira Latas", false, 1);
```

```
        i = 0;
```

```
        x = 0;
```

```
        while(i < caes.length){
```

```
            x += caes[i].getIdade();
```

```
            i++;
```

```
        }
```

```
        System.out.println("\n\nResultado: " + x);
```

```
        int y = 0;
```

```
        for(i = 0; i < caes.length; i++){
```

```
            caes[i].setIdade(caes[i].getIdade() + 1);
```

```
            y += caes[i].getIdade();
```

```
        }
```

```
        System.out.println("\n\nResultado: " + y);
```

```
    }
```

```
}
```

8) Indique o que será impresso pelo programa abaixo:

```
package pvt_exerciciodepuracao2;
```

```
public class ControlaVetor {  
  
    private int[] vet1, vet2, vet3;  
    private int pos;  
    private final int TAM = 10;  
  
    public ControlaVetor(){  
        vet1 = new int[TAM];  
        vet2 = new int[TAM];  
        vet3 = new int[TAM];  
        pos = 0;  
    }  
  
    public int getTotal(){  
        int total;  
        for(int j = 0; j < vet1.length; j++){  
            total += vet1[j];  
        }  
        return(total);  
    }  
  
    public void add(int v){  
        vet1[pos] = v;  
        pos++;  
    }  
  
    public void copia(){  
        for(int j = 0; j < vet1.length; j++){  
            vet2[j] = vet1[TAM - (j + 1)];  
        }  
    }  
  
    public void fazVet3(){  
        for(int j = 0; j < vet1.length; j++){  
            vet3[j] = vet1[j] + vet2[j] - j * j;  
        }  
    }  
  
    public int getSomatorioVet3(){  
        int total;  
        for(int j = 0; j < vet3.length; j++){  
            total += vet3[j];  
        }  
        return(total);  
    }  
  
    public void mudaValoresVet2(int val){  
        for(int j = 0; j < vet2.length; j++){  
            vet2[j] += val;  
        }  
    }  
}
```

=====

```

package pvt_exerciciodepuracao2;

public class PVt_ExercicioDepuracao2 {

    public static void main(String[] args) {
        ControlaVetor cv;
        cv = new ControlaVetor();
        int tam = 10,
            i = 0;

        while(i < tam){
            cv.add(i + i - 2);
            i++;
        }

        System.out.println("\n\nTotal vet1: " + cv.getTotal());

        cv.copia();
        cv.fazVet3();

        System.out.println("\n\nResultado vet3: " + cv.getSomatorioVet3());

        cv.mudaValoresVet2(3);
        cv.fazVet3();

        System.out.println("\n\nResultado vet3: " + cv.getSomatorioVet3());
    }
}

```

Mais informações e **exercícios** poderão ser encontrados em:

DEITEL, H. M.; DEITEL, P. J. **Java Como Programar**; tradução Edson Furmankiewicz; revisão técnica Fábio Lucchini. 6. ed., São Paulo: Pearson, 2005.

DEITEL, H. M.; DEITEL, P. J. **Java Como Programar**; 8. ed., São Paulo: Prentice Hall, 2010.