

1) Faça um programa em Java que preencha uma CLista com 10 números inteiros. Calcule e imprima a soma desses números.

```
public static CLista preencher ( ){
    CLista L = new CLista ( );
    int dado = 0;

    for (int i = 0; i < 10; i++){
        System.out.println("Entre com um dado: ");
        dado = in.nextInt( );
        L.insereComeco(dado);
    }//end for
    return L;
} //end preencher ( )

public static int soma (CLista L){
    CLista L2  = L;
    Object item = null;
    int soma    = 0;
    int dado    = 0;
    int quantidade = L.quantidade( );

    while(quantidade > 0){
        dado = (int) L2.removeRetornaComeco( );
        soma = soma + dado;
        quantidade --;
    }//end while
    return soma;
} //end soma

public static void questao01 ( ){
    CLista L = new CLista( );
    int soma = 0;

    L    = preencher( );
    soma = soma(L);

    System.out.println("SOMA: "+soma);
} //end questao01( )
```

2) Faça um programa em Java que preencha uma CFila com 10 números inteiros. Calcule e imprima a soma desses números.

```
public static CFila preencherF ( ){
    CFila F = new CFila ( );
    int dado = 0;

    for (int i = 0; i < 10; i++){
        System.out.println("Entre com um dado: ");
        dado = in.nextInt( );
        F.enqueue(dado);
    } //end for
    return F;
} //end preencher ( )
```

```

public static int soma (CFila F){
    CFila F2      = F;
    Object item    = null;
    int dado       = 0;
    int soma       = 0;
    int quantidade = F.quantidade( );

    while (quantidade > 0){
        item = F.desenfileira( );
        dado = (int) item;
        soma = soma + (int) dado;
        quantidade--;
    } //end for
    return soma;
} //end soma ( )

```

```

public static void questao02 ( ){
    CFila F  = new CFila( );
    int soma = 0;

    F      = preencherF( );
    soma = soma(F);

    System.out.println("SOMA: "+soma);
} //end questao02( )

```

3) Faça um programa em Java que preencha uma CLista com 10 valores de tipos diferentes (String, double, int, etc...). Tente calcular e imprimir a soma desses elementos. Que erro ocorre?

```

public static CLista variada ( ){
    CLista L = new CLista( );

    boolean bool = true;
    L.insereComeco(bool);
    boolean bool2 = false;
    L.insereComeco(bool2);
    char caract1 = '-';
    L.insereComeco(caract1);
    char caract = 'a';
    L.insereComeco(caract);
    int dado = 10;
    L.insereComeco(dado);
    int value = 79;
    L.insereComeco(value);
    String valor = "valor";
    L.insereComeco(valor);
    String vazia = " ";
    L.insereComeco(vazia);
    double num = 1.85;
    L.insereComeco(num);
    double num2 = 7.92;
    L.insereComeco(num2);

    return L;
} //end preencher ( )

```

```

public static void questao03( ){
    CLista L      = new CLista( );
    int quantidade = 0;
    Object item    = null;
    int dado       = 0;
    int soma       = 0;

    L = variada( );
    quantidade = L.quantidade( );

    while(quantidade > 0){
        item = L.removeRetornaComeco( );
        dado = (int) item;
        soma = soma + dado;
        quantidade --;
    } //end while
    System.out.println("Soma: "+soma);
} //end questao03( )

```

**RESPOSTA:** Percebe-se que ao compilar o código não se é apresentado nenhum erro aparente, porém, ao executar o mesmo é apresentado ao usuário o seguinte erro:

Exception in thread "main" java.lang.ClassCastException:  
java.base/java.lang.Double cannot be cast to java.base/java.lang.Integer

4) Faça um programa em Java que preencha uma CLista com os dados de 5 alunos. Os dados dos alunos estão armazenados em objetos da classe Aluno.

```

import java.util.ArrayList;
import java.util.Scanner;

class Aluno {
    public String nome;
    public int matricula;

    public Aluno ( ) {
        this.nome     = "";
        this.matricula = 0;
    } //end construtor

    public Aluno (String name, int matricula) {
        this.nome     = name;
        this.matricula = matricula;
    } //end construtor
} //end class

```

5) Altere o programa feito na questão 4 para que ele imprima os dados dos 5 alunos conforme o layout a seguir:

```

1 – nome aluno – matrícula
2 – nome aluno – matrícula
...
5 – nome aluno – matrícula

```

```

public class MainCLista {
private static Scanner in = new Scanner (System.in);

    public static void teste ( ) {
        CLista L = new CLista ( );
        Aluno a = new Aluno ("Marlon", 601020);
        Aluno b = new Aluno ("Raissa", 604697);
        Aluno c = new Aluno ("Willian", 613416);
        Aluno d = new Aluno ("Wilton", 623451);
        Aluno e = new Aluno ("Kelven", 613418);
        Object data1, data2, data3, data4, data5;

        data1 = (a.nome + a.matricula);
        data2 = (b.nome + b.matricula);
        data3 = (c.nome + c.matricula);
        data4 = (d.nome + d.matricula);
        data5 = (e.nome + e.matricula);

        L.insereFim(data1);
        L.insereFim(data2);
        L.insereFim(data3);
        L.insereFim(data4);
        L.insereFim(data5);
        L.imprime( );
    } //end teste( )

    public static void main (String [ ] args) {
        teste( );
    } //end main( )
} //end class MainCLista( )

```

6) Crie uma função que receba a estrutura de dados (CLista, CFila e CPilha) como parâmetro e retorne a soma de seus elementos. Obs: considere que todos seus dados são do tipo int.

```

public static int somarTudo (CLista L, CFila F, CPilha P){
    Object L1, F1, P1;
    CLista Laux = L;
    CFila Faux = F;
    CPilha Paux = P;
    int somarTudo = 0;
    int q1 = L.quantidade( );
    int q2 = F.quantidade( );
    int q3 = P.quantidade( );
    int amount1 = 0;
    int amount2 = 0;
    int amount3 = 0;
    int item1, item2, item3;

    //somar dados da Lista
    while(q1 > 0){
        L1 = Laux.removeRetornaComeco( );
        item1 = (int) L1;
        amount1 = amount1 + (int) item1;
        q1 --;
    } //end while

    //somar dados da Pilha

```

```

while(q2 > 0){
    P1    = Paux.desempilha( );
    item2  = (int) P1;
    amount2 = amount2 + (int) item2;
    q2 --;
} //end while

//somar dados da Fila
while(q3 > 0){
    F1    = Faux.desenfileira( );
    item3  = (int) F1;
    amount3 = amount3 + (int) item3;
    q3 --;
} //end while

//somar tudo
somarTudo = amount1 + amount2 + amount3;
return somarTudo;
} //end somarTudo( )

public static void questao06 ( ) {
    CFila F = new CFila ( );
    CPilha P = new CPilha ( );
    CLista L = new CLista ( );

    L = preencher ( );
    System.out.println("");
    F = preencherF ( );
    System.out.println("");
    P = preencherP ( );
    System.out.println("SOMA DE TODOS OS TERMOS: "+somarTudo(L,F,P));

} //end questao06( )

```

7) Crie uma função que calcule o número de elementos positivos de uma estrutura de dados passada como parâmetro.

```

public static int numElementosPosF (CFila F){
    CFila Faux    = F;
    int quantidade = F.quantidade( );
    int amount     = 0;
    int item       = 0;
    Object dado    = null;

    while (quantidade > 0){
        dado = Faux.desenfileira( );
        item = (int) dado;
        if(item > 0){
            amount = amount+1;
        } //end if
        quantidade --;
    } //end while
    return amount;
} //end numElementosPosF( )

```

```

public static int numElementosPosP (CPilha P){
    CPilha Paux    = P;
    int quantidade = P.quantidade( );
    int amount      = 0;
    int item        = 0;
    Object dado     = null;

    while (quantidade > 0){
        dado = Paux.desempilha( );
        item = (int) dado;
        if(item > 0){
            amount = amount+1;
        }//end if
        quantidade --;
    }//end while
    return amount;
} //end numElementosPosP( )

```

```

public static int numElementosPosL (CLista L){
    CLista Laux    = L;
    int quantidade = L.quantidade( );
    int amount      = 0;
    int item        = 0;
    Object dado     = null;

    while (quantidade > 0){
        dado = Laux.removeRetornaComeco( );
        item = (int) dado;
        if(item > 0){
            amount = amount+1;
        }//end if
        quantidade --;
    }//end while
    return amount;
} //end numElementosPosF( )

```

```

public static void questao07( ){
    int dado = 0;

    System.out.println("DIGITE A OPCAO DESEJADA: ");
    System.out.println("1 - CFila");
    System.out.println("2 - CLista");
    System.out.println("3 - CPilha");

    dado = in.nextInt( );

    if(dado == 1){
        System.out.println("");
        System.out.println("CFILA: ");
        CFila F = new CFila( );
        int num = 0;
        F = preencherF( );
        num = numElementosPosF(F);
        System.out.println("Qtde de numeros positivos nesta Fila: "+num);
    } else if (dado == 2){
        System.out.println("");
        System.out.println("CLISTA: ");
    }
}

```

```

        CLista L = new CLista( );
        int num = 0;
        L = preencher( );
        num = numElementosPosL(L);
        System.out.println("Qtde de numeros positivos nesta Lista: "+num);
    } else if (dado == 3){
        System.out.println("");
        System.out.println("CPILHA: ");
        CPilha P = new CPilha( );
        int num = 0;
        P = preencherP( );
        num = numElementosPosP(P);
        System.out.println("Qtde de numeros positivos nesta Pilha: "+num);
    } else {
        System.out.println("ERRO: OPCA0 INVALIDA.");
    }
} //end else
} //end questao07( )

```

8) Crie uma função que calcule o número de ocorrências em uma estrutura de dados de um elemento passado como parâmetro.

```

public static int numOcorrenciasF (CFila F, int n){
    CFila Faux    = F;
    Object dado    = null;
    int item       = 0;
    int ocorrencias = 0;
    int quantidade = F.quantidade( );

    while (quantidade > 0){
        dado = Faux.desenfileira( );
        item = (int) dado;
        if(item == n){
            ocorrencias++;
        } // end if
        quantidade--;
    } //end while

    return ocorrencias;
} //end numOcorrenciasF ( )

```

```

public static int numOcorrenciasP (CPilha P, int n){
    CPilha Paux    = P;
    Object dado     = null;
    int item        = 0;
    int ocorrencias = 0;
    int quantidade  = P.quantidade( );

    while (quantidade > 0){
        dado = Paux.desempilha( );
        item = (int) dado;
        if (item == n){
            ocorrencias++;
        } //end if
        quantidade--;
    } //end while

    return ocorrencias;
} //end numOcorrenciasP ( )

```

```

public static int numOcorrenciasL (CLista L, int n){
    CLista Laux    = L;
    Object dado    = null;
    int item       = 0;
    int ocorrencias = 0;
    int quantidade = L.quantidade();

    while (quantidade > 0){
        dado = Laux.removeRetornaComeco();
        item = (int) dado;
        if(item == n){
            ocorrencias++;
        }//end if
        quantidade--;
    }//end while
    return ocorrencias;
}

public static void questao08 (){
    CFila F = new CFila ();
    CLista L = new CLista ();
    CPilha P = new CPilha ();
    int ocorrenciasF = 0;
    int ocorrenciasP = 0;
    int ocorrenciasL = 0;

    System.out.println("FILA:");
    F = preencherF ();
    System.out.println("LISTA:");
    L = preencher ();
    System.out.println("PILHA:");
    P = preencherP ();

    //contar o numero de ocorrencias
    ocorrenciasF = numOcorrenciasF (F,5);
    ocorrenciasL = numOcorrenciasL (L,4);
    ocorrenciasP = numOcorrenciasP (P,0);

    System.out.println ("Ocorrencias P: " +ocorrenciasP+ " |Ocorrencias F: " +ocorrenciasF+
        " |Ocorrencias L: " +ocorrenciasL);
}

```

**9) Cria as funções** static CLista criaCLista(int opcao, int n), static CFila criaCFila(int opcao, int n), static CPilha criaCPilha (int opcao, int n) **que criam e retornam, respectivamente, uma CLista, CFila e CPilha. O parâmetro pode ter os seguintes valores:**

- 0 – gera a coleção com n elementos, de 0 até n-1
- 1 – gera a coleção em ordem decrescente, de n-1 até 0
- 2 – gera a coleção com n elementos aleatórios

**Exemplo da chamada das funções:**

```

static void main(String[] args){
    CLista testeA;
    CFila testeQ;
    CPilha testeS;

    testeA = criaCLista(2, 400);
    // Gera uma CLista aleatória com 400 elementos
    testeQ = criaCFila(0, 250);
}

```



```
// Gera uma CLista com 250 elementos em ordem crescente
testeS = criaPilha(1, 20);
// Gera uma CPilha com vinte elementos em ordem decrescente
...
} //end main ( )
```

```
public static CLista criaCLista (int opcao, int n){
    int dado = 0;
    CLista L = new CLista( );
    Random gerador = new Random( );

    if(opcao == 0){
        for (int i = 0; i < n; i++){
            L.insereComeco(i);
        } //end for
    } else if (opcao == 1){
        for (int i = n-1; i >= 0; i--){
            L.insereComeco(i);
        } //end for
    } else if (opcao == 2){
        for (int i = 0; i < n; i++){
            dado = gerador.nextInt(100);
            dado = Math.abs(dado);
            L.insereComeco(dado);
        } //end for
    } else {
        System.out.println("ERRO: OPCAO INVALIDA.");
    } //end else

    return L;
} //end criaCLista( )
```

```
public static CPilha criaPilha (int opcao, int n){
    int dado = 0;
    CPilha P = new CPilha( );
    Random gerador = new Random( );

    if (opcao == 0){
        for (int i = 0; i < n; i++){
            P.empilha(i);
        } //end for
    } else if (opcao == 1){
        for (int i = n-1; i >= 0; i--){
            P.empilha(i);
        } //end for
    } else if (opcao == 2){
        for (int i = 0; i < n; i++){
            dado = gerador.nextInt(100);
            dado = Math.abs(dado);
            P.empilha(dado);
        } //end for
    } else {
        System.out.println("ERRO: OPCAO INVALIDA.");
    } //end else

    return P;
} //end criaPilha( )
```

```

public static CFila criaFila (int opcao, int n){
    int dado = 0;
    CFila F = new CFila( );
    Random gerador = new Random( );

    if (opcao == 0){
        for (int i = 0; i < n; i++){
            F.enfileira(i);
        } //end for
    } else if (opcao == 1){
        for (int i = n-1; i >= 0; i--){
            F.enfileira(i);
        } //end for
    } else if (opcao == 2){
        for (int i = 0; i < n; i++){
            dado = gerador.nextInt(100);
            dado = Math.abs(dado);
            F.enfileira(dado);
        } //end for
    } else {
        System.out.println("ERRO: OPCAO INVALIDA.");
    } //end else

    return F;
} //end criaFila( )

```

```

public static void questao09( ){
    CFila F      = new CFila ( );
    CLista L     = new CLista( );
    CPilha P     = new CPilha( );
    int opcao    = 0;
    int quantidade = 0;
    int opcao2   = 0;

    System.out.println("Digite a opcao desejada: ");
    System.out.println("");
    System.out.println("0 - Gera a colecao com n elementos, de 0 ate n-1.");
    System.out.println("1 - Gera a colecao com n elementos em ordem decrescente, de n-1 ate 0.");
    System.out.println("2 - Gera a colecao com n elementos aleatorios.");
    opcao = in.nextInt( );
    System.out.println("");
    System.out.println("Entre com a quantidade de dados que deseja: ");
    quantidade = in.nextInt( );
    System.out.println("");
    System.out.println("Qual e' o tipo de celula desejada: ");
    System.out.println("110 - Cfila");
    System.out.println("120 - CPilha");
    System.out.println("130 - Clista");
    opcao2 = in.nextInt( );
    System.out.println("");

    if(opcao2 == 110){
        System.out.println("FILA:");
        F = criaFila(opcao, quantidade);
        imprimeCFila(F);
        System.out.println("");
    } else if (opcao2 == 120){
        System.out.println("PILHA:");
        P = criaPilha(opcao, quantidade);
        imprimeCPilha(P);
    }
}

```

```

        System.out.println("");
    } else if (opcao2 == 130){
        System.out.println("LISTA:");
        L = criaCLista(opcao, quantidade);
        imprimeCLista(L);
        System.out.println("");
    } else {
        System.out.println("ERRO: OPCAO INVALIDA.");
    }
} //end else
} //end questao09( )

```

**10) Crie as funções static void imprimeCLista(CLista A), static void imprimeCFila(CFila Q) e static void imprimeCPilha(CPilha S). As funções devem imprimir 10 elementos por linha, separados por barra "|". Obs: As funções devem ser não destrutivas (ou seja, após a impressão, os dados da coleção devem continuar na mesma ordem original).**

**Exemplo:**

```

1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10
11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20
21 | 22 | 23 | 24

```

```

public static void imprimeCPilha(CPilha P){
    CPilha Paux = P;
    Object item = null;
    int dado = 0;
    int quantidade = P.quantidade( );

    while(quantidade > 0){
        if (quantidade > 1){
            item = Paux.desempilha( );
            dado = (int) item;
            System.out.print(""+dado+"|");
        } else {
            System.out.print(""+dado);
        }
        //end else
        quantidade --;
    }
    //end while
} //end imprimeCPilha( )

```

```

public static void imprimeCFila(CFila F){
    CFila Faux = F;
    Object item = null;
    int dado = 0;
    int quantidade = F.quantidade( );

    while(quantidade > 0){
        if (quantidade > 1){
            item = Faux.desenfileira( );
            dado = (int) item;
            System.out.print(""+dado+"|");
        } else {
            System.out.print(""+dado);
        }
        //end else
        quantidade --;
    }
}

```

```
    }//end while  
} //end imprimeCFila( )
```

```
public static void imprimeCLista(CLista L){  
    CLista Laux  = L;  
    Object item   = null;  
    int dado      = 0;  
    int quantidade = L.quantidade( );  
  
    while(quantidade > 0){  
        if(quantidade > 1){  
            item = Laux.removeRetornaComeco( );  
            dado = (int) item;  
            System.out.print(""+dado+"|");  
        }else{  
            System.out.print(""+dado);  
        }  
        //end else  
        quantidade --;  
    }  
} //end while  
} //end imprimeCLista( )
```

```
public static void questao10( ){  
    CFila F = new CFila( );  
    F = preencherF( );  
  
    CLista L = new CLista( );  
    L = preencher( );  
  
    CPilha P = new CPilha( );  
    P = preencherP( );  
  
    imprimeCFila(F);  
    System.out.println("");  
    imprimeCLista(L);  
    System.out.println("");  
    imprimeCPilha(P);  
  
} //end questao10( )
```