

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

```
public class Exercicio1 {

    public static void main(String[] args) {

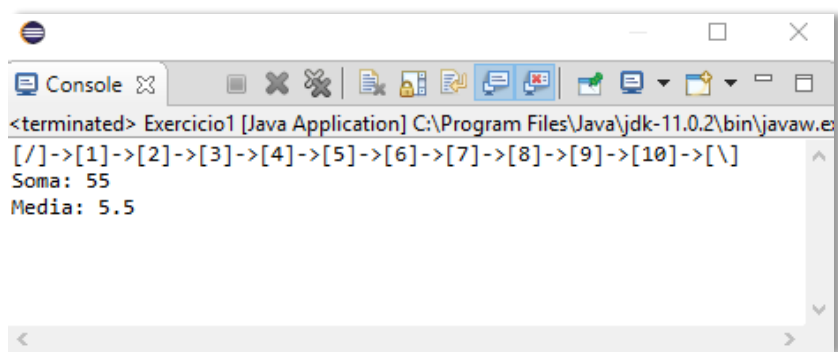
        CLista lista = new CLista();

        int[] numeros = {1,2,3,4,5,6,7,8,9,10};
        int soma = 0;

        for (int i = 0; i < numeros.length; i++) {
            lista.insereFim(numeros[i]);
            soma += numeros[i];
        }
        lista.imprimeFormatoLista();
        System.out.println("Soma: " + soma);
        System.out.println("Media: " + ((float)soma/numeros.length));
    }

}
```

Saída do Programa



```
/*                      Exercício 2
/* Faça um programa em Java que preencha uma
/* CFile com 10 números inteiros.
/* Calcule e imprima a soma desses números.
/*/
```

```
public class Exercicio2 {

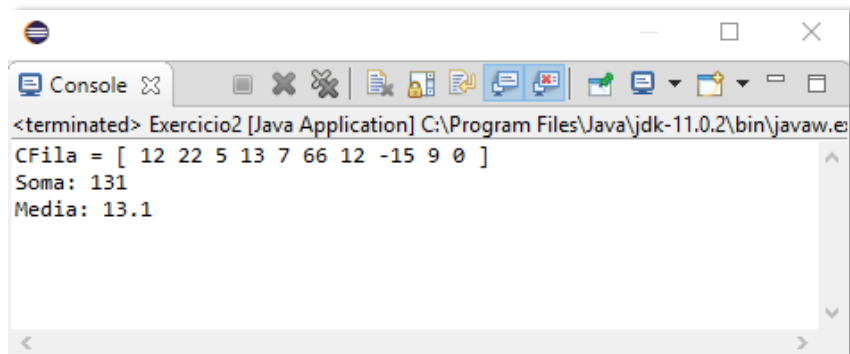
    public static void main(String[] args) {

        CFile fila = new CFile();

        int[] numeros = { 12, 22, 5, 13, 7, 66, 12, -15, 9, 0 };
        int soma = 0;
        for (int i = 0; i < numeros.length; i++) {
            fila.enfileira(numeros[i]);
            soma += numeros[i];
        }

        System.out.print("CFile = " );
        fila.mostra();
        System.out.println("Soma: " + soma);
        System.out.println("Media: " + ((float) soma / numeros.length));
    }
}
```

Saída do Programa



The screenshot shows a Java console window titled "Saída do Programa". The console output is as follows:

```
<terminated> Exercicio2 [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.e
CFile = [ 12 22 5 13 7 66 12 -15 9 0 ]
Soma: 131
Media: 13.1
```

```

/*
                                Exercício 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 class Exercicio3 {

    public static void main(String[] args) {

        Object[] valores = {"Inteiro",2,"4",5.4,true,6,1,false,"9",10.01};
        int soma = 0;

        for (int i = 0; i < valores.length; i++) {
            lista.insereFim(valores[i]);
            soma = soma + valores[i];
        }

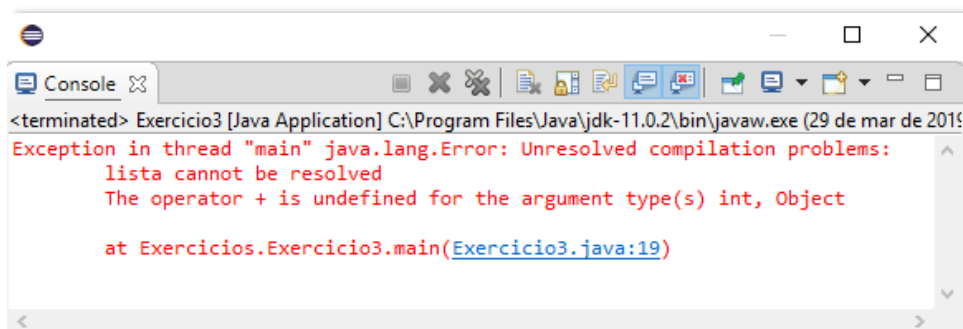
        System.out.println("Soma: " + soma);
        System.out.println("Media: " + ((float)soma/valores.length));

        // The Operator + is undefined for the arguments type(s) int, object;
    }

}

```

Saída do Programa



```

/*                                Exercício 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,
 * conforme abaixo:
 *
 * class Aluno {
 *     String nome;
 *     int matricula;
 * }
 */

public class Exercicio4 {

    public static void main(String[] args) {

        CLista lista = new CLista();

        Aluno aluno1 = new Aluno("Luis", 1234);
        lista.insereFim(aluno1);

        Aluno aluno2 = new Aluno("Antonio", 1337);
        lista.insereFim(aluno2);

        Aluno aluno3 = new Aluno("Marina", 2334);
        lista.insereFim(aluno3);

        Aluno aluno4 = new Aluno("Luisa", 8254);
        lista.insereFim(aluno4);

        Aluno aluno5 = new Aluno("Flavio", 1454);
        lista.insereFim(aluno5);

        Aluno aluno6 = new Aluno("Rodrigo", 2323);
        lista.insereFim(aluno6);

        /* Pegar Classe Aluno */
    }
}

public class Aluno {
    String nome;
    int matricula;

    public String getNome() {
        return nome;
    }

    public int getMatricula() {
        return matricula;
    }

    public Aluno(String nome, int matricula) {
        super();
        this.nome = nome;
        this.matricula = matricula;
    }
}

```

```

/*                                Exercício 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 Exercicio5 {

    public static void main(String[] args) {

        CLista lista = new CLista();

        Aluno aluno1 = new Aluno("Luis", 1234);
        lista.insereFim(aluno1);

        Aluno aluno2 = new Aluno("Antonio", 1337);
        lista.insereFim(aluno2);

        Aluno aluno3 = new Aluno("Marina", 2334);
        lista.insereFim(aluno3);

        Aluno aluno4 = new Aluno("Luisa", 8254);
        lista.insereFim(aluno4);

        Aluno aluno5 = new Aluno("Flavio", 1454);
        lista.insereFim(aluno5);

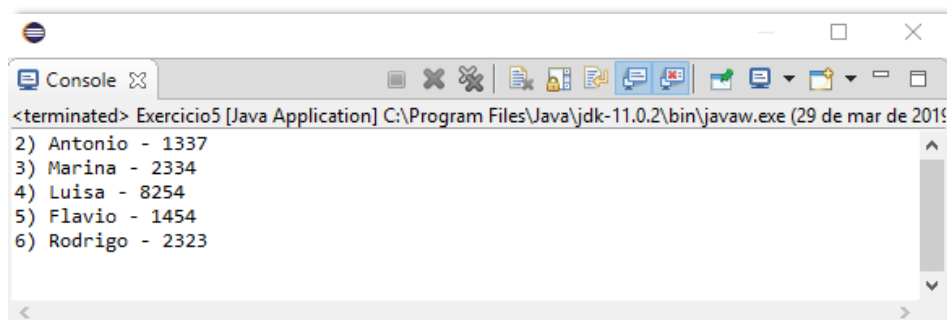
        Aluno aluno6 = new Aluno("Rodrigo", 2323);
        lista.insereFim(aluno6);

        int tamLista = lista.quantidade();

        for (int i = 1; i <= tamLista; i++) {
            Object aluno = lista.retornaPrimeiro();
            System.out.println(i + ") " + ((Aluno) aluno).getNome() + " - " +
            ((Aluno) aluno).getMatricula());
            lista.removeComecoSemRetorno();
        }
    }
}

```

Saída do Programa



```

<terminated> Exercicio5 [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (29 de mar de 2019)
2) Antonio - 1337
3) Marina - 2334
4) Luisa - 8254
5) Flavio - 1454
6) Rodrigo - 2323

```

```

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

```

```

public class Exercício6 {

    public static void main(String[] args) {

        CPilha pilha = new CPilha();
        pilha.empilha(1);
        pilha.empilha(-1);
        pilha.empilha(2);
        pilha.empilha(1);

        CFile fila = new CFile();
        fila.enqueue(1);
        fila.enqueue(-1);
        fila.enqueue(4);
        fila.enqueue(-1);
        fila.enqueue(4);
        fila.enqueue(4);

        CLista lista = new CLista();
        lista.insereComeço(1);
        lista.insereComeço(-1);
        lista.insereComeço(3);
        lista.insereComeço(-1);
        lista.insereComeço(3);
        lista.insereComeço(3);

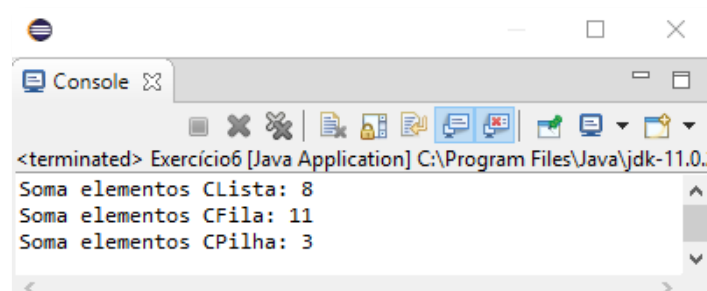
        System.out.println("Soma elementos CLista: " + somaElementos(lista));
        System.out.println("Soma elementos CFile: " + somaElementos(fila));
        System.out.println("Soma elementos CPilha: " + somaElementos(pilha));
    }

    public static int somaElementos(Object tipo) {
        int somaElementos = 0;
        int quantItems = 0;

        if (tipo instanceof CLista) {
            quantItems = ((CLista) tipo).quantidade();
            for (int i = 1; i <= quantItems; i++) {
                somaElementos += (int) ((CLista) tipo).removeRetornaComeço();
            }
        } else if (tipo instanceof CPilha) {
            quantItems = ((CPilha) tipo).quantidade();
            for (int i = 1; i <= quantItems; i++) {
                somaElementos = somaElementos + (int) ((CPilha) tipo).desempilha();
            }
        } else if (tipo instanceof CFile) {
            quantItems = ((CFile) tipo).quantidade();
            for (int i = 1; i <= quantItems; i++) {
                somaElementos += (int) ((CFile) tipo).desenqueue();
            }
        }
        return somaElementos;
    }
}

```

Saída do Programa



```

/*                                Exercício 7
 * Crie uma função que calcule o número de elementos positivos de uma estrutura
 * de dados passada como parâmetro
 */

```

```

public class Exercício7 {
    public static void main(String[] args) {

        CPilha pilha = new CPilha();
        pilha.empilha(1); pilha.empilha(-1); pilha.empilha(2); pilha.empilha(1);

        CFila fila = new CFila();
        fila.enfileira(1); fila.enfileira(-1); fila.enfileira(4); fila.enfileira(-1);
        fila.enfileira(4); fila.enfileira(4);

        CLista lista = new CLista();
        lista.insereComeco(1); lista.insereComeco(-1); lista.insereComeco(3);
        lista.insereComeco(-1); lista.insereComeco(3); lista.insereComeco(3);

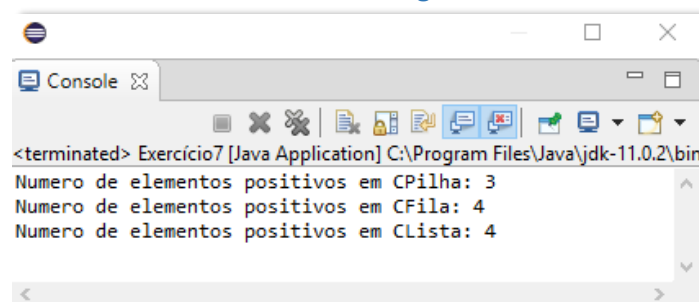
        System.out.println("Numero de elementos positivos em CPilha: " +
            numElementosPositivos(pilha));
        System.out.println("Numero de elementos positivos em CFila: " +
            numElementosPositivos(fila));
        System.out.println("Numero de elementos positivos em CLista: " +
            numElementosPositivos(lista));
    }

    public static int numElementosPositivos(Object tipo) {
        int contElementos = 0, elemento, quantItems;

        if (tipo instanceof CLista) {
            quantItems = ((CLista) tipo).quantidade();
            for (int i = 1; i <= quantItems; i++) {
                elemento = (int) ((CLista) tipo).removeRetornaComeco();
                if (elemento > 0)
                    contElementos++;
            }
        } else if (tipo instanceof CPilha) {
            quantItems = ((CPilha) tipo).quantidade();
            for (int i = 1; i <= quantItems; i++) {
                elemento = (int) ((CPilha) tipo).desempilha();
                if (elemento > 0)
                    contElementos++;
            }
        } else if (tipo instanceof CFila) {
            quantItems = ((CFila) tipo).quantidade();
            for (int i = 1; i <= quantItems; i++) {
                elemento = (int) ((CFila) tipo).desenfileira();
                if (elemento > 0)
                    contElementos++;
            }
        }
        return contElementos;
    }
}

```

Saída do Programa



```

<terminated> Exercício7 [Java Application] C:\Program Files\Java\jdk-11.0.2\bin
Numero de elementos positivos em CPilha: 3
Numero de elementos positivos em CFila: 4
Numero de elementos positivos em CLista: 4

```

```

/*                                Exercício 8
 * Crie uma função que calcule o número de ocorrências em uma estrutura de dados
 * de um elemento passado como parâmetro
 */
public class Exercício8 {

    public static void main(String[] args) {

        CPilha pilha = new CPilha();
        pilha.empilha(1); pilha.empilha(-1); pilha.empilha(2); pilha.empilha(1);

        CFile fila = new CFile();
        fila.enfileira(1); fila.enfileira(-1); fila.enfileira(4); fila.enfileira(-1);
        fila.enfileira(4); fila.enfileira(4);

        CLista lista = new CLista();
        lista.insereComeco(1); lista.insereComeco(-1); lista.insereComeco(3);
        lista.insereComeco(-1); lista.insereComeco(3); lista.insereComeco(3);

        System.out.println("Numero de ocorrências do item 1 na pilha instanciada: " +
            numDeOcorrencias(pilha, 1));
        System.out.println("Numero de ocorrências do item 4 na fila instanciada: " +
            numDeOcorrencias(fila, 4));
        System.out.println("Numero de ocorrências do item 3 na lista instanciada: " +
            numDeOcorrencias(lista, 3));

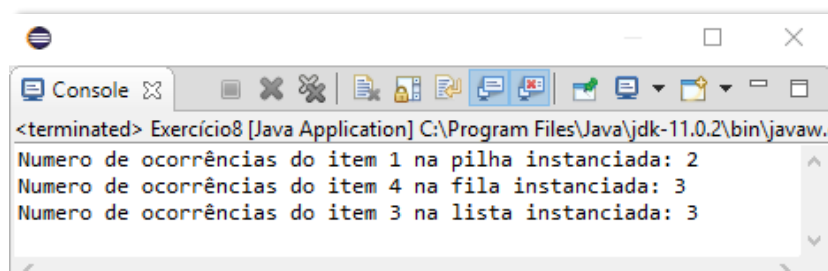
    }

    public static int numDeOcorrencias(Object tipo, Object item) {
        int contElementos = 0, quantItems;
        Object elemento;

        if (tipo instanceof CLista) {
            quantItems = ((CLista) tipo).quantidade();
            for (int i = 1; i <= quantItems; i++) {
                elemento = ((CLista) tipo).removeRetornaComeco();
                if (elemento.equals(item))
                    contElementos++;
            }
        } else if (tipo instanceof CPilha) {
            quantItems = ((CPilha) tipo).quantidade();
            for (int i = 1; i <= quantItems; i++) {
                elemento = ((CPilha) tipo).desempilha();
                if (elemento.equals(item))
                    contElementos++;
            }
        } else if (tipo instanceof CFile) {
            quantItems = ((CFile) tipo).quantidade();
            for (int i = 1; i <= quantItems; i++) {
                elemento = ((CFile) tipo).desenfileira();
                if (elemento.equals(item))
                    contElementos++;
            }
        }
        return contElementos;
    }
}

```

Saída do Programa



```

<terminated> Exercício8 [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.
Numero de ocorrências do item 1 na pilha instanciada: 2
Numero de ocorrências do item 4 na fila instanciada: 3
Numero de ocorrências do item 3 na lista instanciada: 3

```



```

/*                                Exercício 9
 * Crie 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
 * opcao 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;
 */

```

```

public class Exercício9 {
    public static void main(String[] args) {

        CLista testeA;
        System.out.println(" * Lista crescente de 0 até n-1: ");
        testeA = criaCLista(0,13);
        testeA.imprimeFormatoLista();

        CFila testeQ;
        System.out.println("\n * Lista decrescente de n-1 até 0: ");
        testeQ = criaCFila(1,20);
        testeQ.mostra();

        CPilha testeS;
        System.out.println("\n * Lista de itens aleatórios com n elementos: ");
        testeS = criaCPilha(2, 25);
        testeS.mostra();
    }

    public static CLista criaCLista(int opcao, int n) {
        int element;
        CLista lista = new CLista();

        Random gerador = new Random();

        if (opcao == 0) {
            element = 0;
            for (int i = 0; i <= n - 1; i++) {
                lista.insereFim(element);
                element++;
            }
            return lista;
        } else if (opcao == 1) {
            element = n - 1;
            for (int i = n; i > 0; i--) {
                lista.insereFim(element);
                element--;
            }
            return lista;
        } else if (opcao == 2) {
            for (int i = 0; i <= n - 1; i++) {
                lista.insereFim(gerador.nextInt(100));
            }
            return lista;
        }
        return lista;
    }

    public static CFila criaCFila(int opcao, int n) {
        int element;
        CFila fila = new CFila();

        Random gerador = new Random();

        if (opcao == 0) {
            element = 0;

```

```

        for (int i = 0; i <= n - 1; i++) {
            fila.enfileira(element);
            element++;
        }
        return fila;
    } else if (opcao == 1) {
        element = n - 1;
        for (int i = n; i > 0; i--) {
            fila.enfileira(element);
            element--;
        }
        return fila;
    } else if (opcao == 2) {
        for (int i = 0; i <= n - 1; i++) {
            fila.enfileira(gerador.nextInt(100));
        }
        return fila;
    }
    return fila;
}

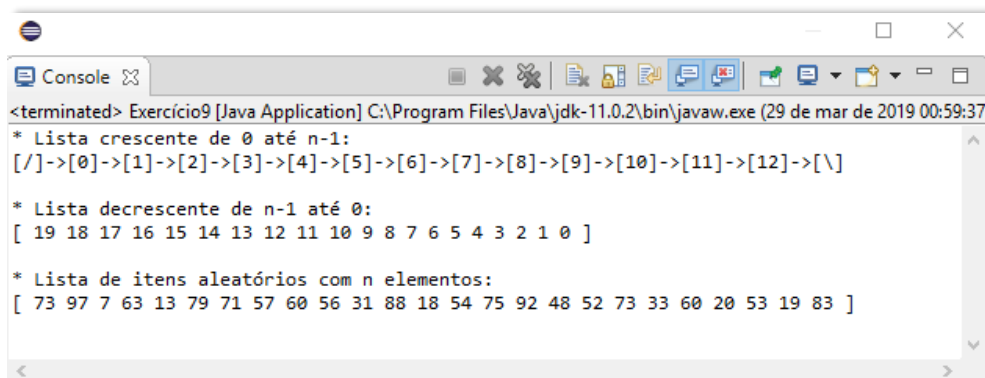
public static CPilha criaCPilha(int opcao, int n) {
    int element;
    CPilha pilha = new CPilha();

    Random gerador = new Random();

    if (opcao == 0) {
        element = 0;
        for (int i = 0; i <= n; i++) {
            pilha.empilha(element);
            element++;
        }
        return pilha;
    } else if (opcao == 1) {
        element = n - 1;
        for (int i = n; i >= 0; i--) {
            pilha.empilha(element);
            element--;
        }
        return pilha;
    } else if (opcao == 2) {
        for (int i = 0; i <= n; i++) {
            pilha.empilha(gerador.nextInt(100));
        }
        return pilha;
    }
    return pilha;
}
}

```

Saída do Programa



```

<terminated> Exercício9 [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (29 de mar de 2019 00:59:37)
* Lista crescente de 0 até n-1:
[/]->[0]->[1]->[2]->[3]->[4]->[5]->[6]->[7]->[8]->[9]->[10]->[11]->[12]->[\]

* Lista decrescente de n-1 até 0:
[ 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 ]

* Lista de itens aleatórios com n elementos:
[ 73 97 7 63 13 79 71 57 60 56 31 88 18 54 75 92 48 52 73 33 60 20 53 19 83 ]

```

```

/*
                                Exercício 10
* Crie as funções static void imprimeLista(CListaA), static void imprimeCFila(CFilaQ) e static void
* imprimeCPilha(CPilhaS).
* 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 class Exercício10 {
    public static void main(String[] args) {

        CLista lista1 = new CLista();
        for (int i = 0; i < 18; i++) {
            lista1.insereFim(i);
        }

        CFila fila1 = new CFila();
        for (int i = 0; i < 25; i++) {
            fila1.enfileira(i);
        }

        CPilha pilha1 = new CPilha();
        for (int i = 24; i >= 0; i--) {
            pilha1.empilha(i);
        }
        System.out.println("CLista: ");
        imprimeLista(lista1);
        System.out.println("\n");

        System.out.println("CFila: ");
        imprimeCFila(fila1);
        System.out.println("\n");

        System.out.println("CPilha: ");
        imprimeCPilha(pilha1);
        System.out.println("\n");
    }

    public static void imprimeLista(CLista A) {
        CLista listaFake = new CLista();
        int contPulaLinha = 0;

        int tamLista = ((CLista) A).quantidade();
        for (int i = 1; i < tamLista; i++) {
            listaFake.insereComeco(A.removeRetornaComeco());
            System.out.print(A.retornaPrimeiro());
            if (i < tamLista - 1)
                System.out.print(" | ");
            contPulaLinha++;
            if (contPulaLinha >= 10) {
                System.out.println();
                contPulaLinha = 0;
            }
        }

        System.out.print(" \n ");
        for (int i = 1; i < tamLista; i++) {
            A.insereComeco(listaFake.removeRetornaComeco());
        }
        A.imprimeFormatoLista();
    }

    public static void imprimeCFila(CFila Q) {
        CFila filaFake = new CFila();
        int contPulaLinha = 0;

        int tamFila = ((CFila) Q).quantidade();
        for (int i = 1; i < tamFila; i++) {
            filaFake.enfileira(Q.desenfileira());
            System.out.print(Q.peek());
        }
    }
}

```

```

        if (i < tamFila -1)
            System.out.print(" | ");
        contPulaLinha++;
        if (contPulaLinha >= 10) {
            System.out.println();
            contPulaLinha = 0;
        }
    }

    System.out.print(" \n ");
    for(int i = 1; i < tamFila; i++) {
        Q.enqueue(filaFake.desenfileira());
    }
    Q.mostra();
}

public static void imprimeCPilha(CPilha S) {
    CPilha pilhaFake = new CPilha();
    int contPulaLinha = 0;

    int tamFila = ((CPilha) S).quantidade();
    for (int i = 1; i < tamFila; i++) {
        pilhaFake.empilha(S.desempilha());
        System.out.print(S.peak());
        if (i < tamFila -1)
            System.out.print(" | ");
        contPulaLinha++;
        if (contPulaLinha >= 10) {
            System.out.println();
            contPulaLinha = 0;
        }
    }

    System.out.print(" \n ");
    for(int i = 1; i < tamFila; i++) {
        S.empilha(pilhaFake.desempilha());
    }
    S.mostra();
}
}

```

Saída do Programa

```

<terminated> Exercício10 [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (29 de mar de 2019 01:09:50)

Lista:
1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
11 | 12 | 13 | 14 | 15 | 16 | 17
[/]->[0]->[1]->[2]->[3]->[4]->[5]->[6]->[7]->[8]->[9]->[10]->[11]->[12]->[13]->[14]->[15]->[16]->[17]->[\]

CFila:
1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
21 | 22 | 23 | 24
[ 24 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 ]

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

```