

Instruções de entrega: entregar as respostas em papel para o professor na aula do dia 05/abr. A entrega deverá ser individual.

Nos exercícios considere as classes UML e representações das Figura 1 a Figura 7.

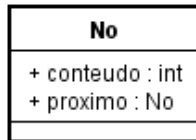


Figura 1 – Diagrama UML da classe No para lista encadeada.

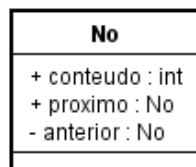


Figura 2 – Diagrama UML da classe No para lista duplamente encadeada.

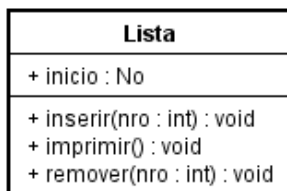


Figura 3 – Diagrama UML da classe Lista.

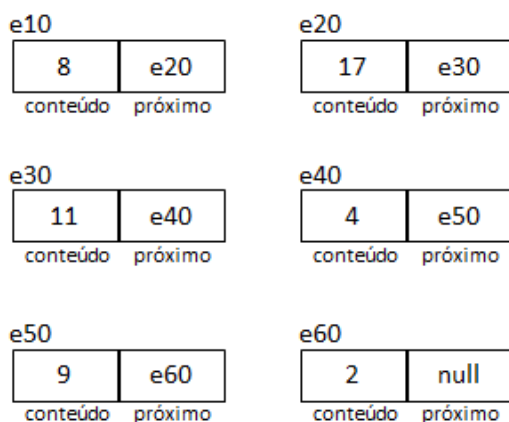


Figura 4 – Representação da lista encadeada na memória, onde cada nó é um objeto do tipo No da Figura 1.

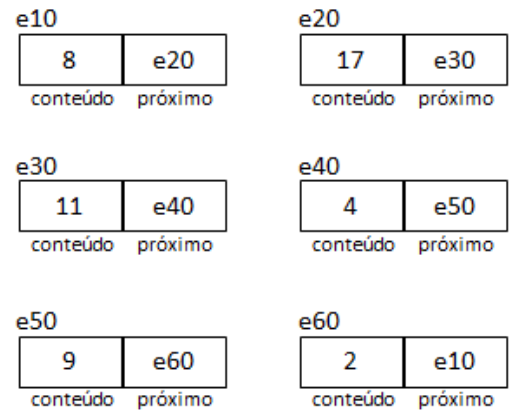


Figura 5 – Representação da lista encadeada circular na memória, onde cada nó é um objeto do tipo No da Figura 1.

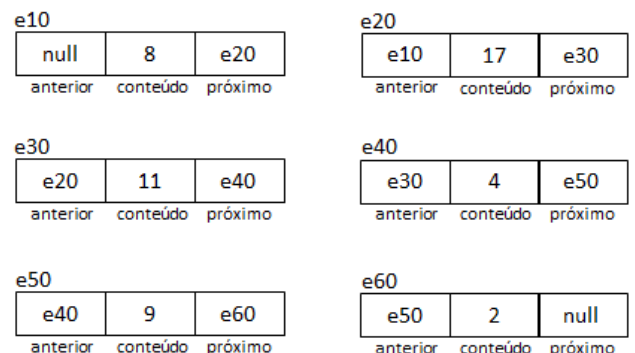


Figura 6 – Representação da lista duplamente encadeada na memória, onde cada nó é um objeto do tipo No da Figura 2.

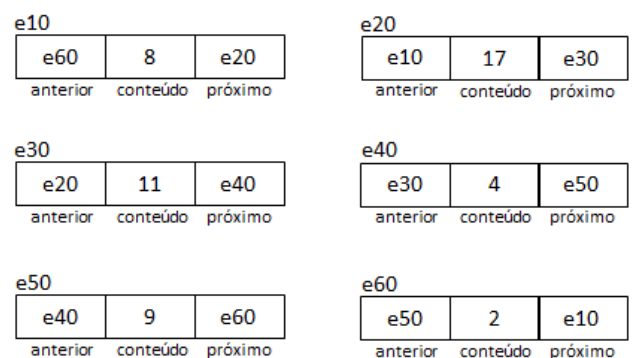


Figura 7 – Representação da lista circular duplamente encadeada na memória, onde cada nó é um objeto do tipo No da Figura 2.

- 1) Considerando a lista representada na Figura 4 e que 1ª instrução a seguir irá imprimir o valor 11. Marque a alternativa que contém o valor a ser impresso ao executar a 2ª instrução.

```
System.out.println( no conteudo );
System.out.println( no.proximo.proximo conteudo );
```

- a) 2 d) null
b) 4 e) Código apresenta erro
c) 9

- 2) Considerando a lista representada na Figura 4 e que 1ª instrução println a seguir irá imprimir o valor 11. Marque a alternativa que contém o valor a ser impresso ao executar a 2ª instrução println.

```
No temp = no.proximo;
System.out.println( no conteudo );
lista.remove(temp.conteudo);
System.out.println( no.proximo.proximo.conteudo );
```

- a) 2 d) null
b) 4 e) Código apresenta erro
c) 9

- 3) Considerando que a lista representada na Figura 4 está acessível pela variável lista. Marque a alternativa que contém os valores a serem impressos ao executar as instruções a seguir.

```
No a = lista.inicio;
No b = a.proximo;
No c = b.proximo;
System.out.println( a.conteudo );
System.out.println( b.conteudo );
System.out.println( c.conteudo );
```

- a) 8 17 11 d) 8 8 8
b) 0 0 0 e) Código apresenta erro
c) 8 17 17

- 4) Considerando que foi programado o método a seguir na classe Lista da Figura 3:

```
void imprimir(No no){
    if( no != null ){
        System.out.print(no.conteudo+";");
```

```
        imprimir(no.proximo);
    }
    System.out.print("aqui+";");
}
```

Considere também que a lista representada na Figura 4 está acessível pela variável lista.

Marque a alternativa que contém a quantidade de vezes que o texto aqui será impresso na tela ao executar a instrução a seguir:

```
lista.imprimir(lista.inicio);
```

- a) 0 d) 7
b) 5 e) Código apresenta erro
c) 6

- 5) Considerando que foi programado o método a seguir na classe Lista da Figura 3:

```
void imprimir(No no){
    System.out.print("aqui+";");
    if( no != null ){
        System.out.print(no.conteudo+";");
        imprimir(no.proximo);
    }
}
```

Considere também que a lista representada na Figura 4 está acessível pela variável lista.

Marque a alternativa que contém a quantidade de vezes que o texto aqui será impresso na tela ao executar a instrução a seguir:

```
lista.imprimir(lista.inicio);
```

- a) 0 d) 7
b) 5 e) Código apresenta erro
c) 6

- 6) Considerando que foi programado o método a seguir na classe Lista da Figura 3:

```
int size(int cont, No no){
    No ultimo = inicio;
    while( ultimo != null ){
        cont++;
        ultimo = ultimo.proximo;
    }
    return cont;
}
```

Considere também que a lista representada na Figura 4 está acessível pela variável lista.

Marque a alternativa que contém o valor que será impresso na tela ao executar a instrução a seguir:

```
System.out.print(lista.size(0, lista.inicio));
```

- a) 0
- b) 5
- c) 6
- d) 7
- e) Código apresenta erro

7) Considerando que foi programado o método a seguir na classe Lista da Figura 3:

```
int size(int cont, No no){
    if( no.proximo != null ){
        return size(++cont, no.proximo);
    }
    return cont;
}
```

Considere também que a lista representada na Figura 4 está acessível pela variável **lista**.

Marque a alternativa que contém o valor que será impresso na tela ao executar a instrução a seguir:

```
System.out.print(lista.size(0, lista.inicio));
```

- a) 0
- b) 5
- c) 6
- d) 7
- e) Código apresenta erro

8) Altere a instrução **++cont** para **cont++** no método `size(cont:int, no:No)` do Exercício 7 e marque a alternativa que contém o valor que será impresso na tela ao executar a instrução a seguir:

```
System.out.print(lista.size(0, lista.inicio));
```

- a) 0
- b) 5
- c) 6
- d) 7
- e) Código apresenta erro

9) Considerando que a variável **ultimo** aponta para o último nó da lista representada na Figura 5. Marque a alternativa que contém o código que insere o **no** a seguir no final dessa lista.

```
No no = new No();
no.conteudo = 22;
```

- a) ultimo.proximo = no;
no = ultimo.proximo;
- b) ultimo = no;

- c) no.proximo = ultimo.proximo;
ultimo.proximo = no;
- d) ultimo = no;
no.proximo = ultimo.proximo;
- e) ultimo.proximo = no;
no.proximo = ultimo.proximo;

10) Considerando a lista representada na Figura 5 e que ao executar as instruções a seguir seja impresso na tela os valores 17 e 11, nessa ordem. Marque a alternativa que contém o código que insere o **no** entre os nós **a** e **b** dessa lista.

```
System.out.print( a.conteudo );
System.out.print( b.conteudo );
No no = new No();
no.conteudo = 22;
```

- a) a.proximo = no.proximo;
no.proximo = b.proximo;
- b) no.proximo = b;
- c) b.proximo = no;
- d) a.proximo = no;
no.proximo = b.proximo;
- e) no.proximo = a.proximo;
a.proximo = no;

11) Considerando a lista representada na Figura 5 e que ao executar as instruções a seguir seja impresso na tela os valores 17 e 11, nessa ordem. Marque a alternativa que contém o código que insere o **no** entre os nós que contém esses valores.

```
System.out.print( a.conteudo );
System.out.print( a.proximo.conteudo );
No no = new No();
no.conteudo = 22;
```

- a) a.proximo = no.proximo;
no.proximo = b.proximo;
- b) no.proximo = a.proximo;
- c) a.proximo.proximo = no;
- d) a.proximo = no;
no.proximo = a.proximo.proximo;
- e) no.proximo = a.proximo;
a.proximo = no ;

12) Considerando a lista representada na Figura 5 e que ao executar a instrução a seguir seja impresso na tela o valor 4. Marque a alternativa que contém o código que remove da lista o nó que contém o valor 9.

```
System.out.print( a conteudo );
```

- a) No b = a;
a.proximo = b.proximo;
- b) No b = a.proximo;
b.proximo = a;
- c) a.proximo = a.proximo.proximo;
- d) No b = a.proximo;
b.proximo = a.proximo.proximo;
- e) a.proximo.proximo = a.proximo;

13) Considerando a lista representada na Figura 6 e que ao executar a instrução a seguir seja impresso na tela o valor 4. Marque a alternativa que contém o código que remove da lista este nó.

```
System.out.print( a conteudo );
```

- a) a.anterior = a.proximo;
- b) No b = a.anterior, c = a.proximo;
b.proximo = a.anterior;
c.anterior = a.proximo;
- c) a.proximo = a.proximo.proximo;
- d) a.anterior.proximo = a.proximo;
a.proximo.anterior = a.anterior;
- e) a.proximo.anterior = a.anterior;

14) Considerando a lista representada na Figura 6 e que ao executar a instrução a seguir seja impresso na tela o valor 11. Marque a alternativa que contém o código que insere o **no** a direita do nó **a**.

```
System.out.print( a conteudo );  
No no = new No();  
no.conteudo = 22;
```

- a) a.proximo.anterior = no;
no.proximo = a.proximo;
a.proximo = no;
no.anterior = a;
- b) a.proximo = no;
no.anterior = a;
a.proximo.anterior = no;

```
no.proximo = a.proximo;
```

- c) a.proximo.anterior = no;
a.anterior.proximo = no.anterior;
no.anterior = a;
no.proximo = a.proximo.proximo;
- d) no.proximo = a.proximo.proximo;
no.anterior = a.anterior;
- e) no.anterior = a;
a.proximo = no;
no.proximo = a.proximo;

15) Considere que a variável **inicio** do trecho de código a seguir aponte para um nó qualquer de uma lista circular duplamente encadeada. Marque a alternativa que contém a situação da lista que **não** faz o laço entrar em loop infinito.

```
No a = inicio;  
while( a != null ){  
    a = a.anterior;  
}
```

- a) A lista está vazia.
- b) A lista possui 1 elemento.
- c) A lista possui 2 elementos.
- d) A lista possui 3 elementos.
- e) A lista possui mais de 3 elementos.

16) Considerando a lista representada na Figura 7. Analise o trecho de código a seguir e marque a alternativa correta.

```
No a = inicio;  
while( a != null && a.conteudo != nro ){  
    a = a.proximo;  
}  
System.out.println( a );
```

- a) O código imprime null na tela em qualquer situação.
- b) O código entra em loop infinito somente se a lista não estiver vazia.
- c) O código entra em loop infinito somente se a lista tiver 1 elemento.
- d) O código entra em loop infinito somente se a lista tiver o nro a partir da 2ª posição.
- e) O código entra em loop infinito somente se a lista não estiver vazia e o nro não existir na lista.