

Roberto da Silva Bigonha Mariza A.S.Bigonha Laboratório de Linguagens de Programação

Departamento de Ciência da Computação

Universidade Federal de Minas Gerais

Agosto de 2018

Todos os direitos reservados Proibida a cópia sem autorização dos autores

@Roberto S. Bigonha e Mariza A.S.Bigonha

Estilo de Programação

Maquinas de Estado

1



OBJETOS COMO MÁQUINAS DE ESTADO



- Objeto é um estado.
- Chamada a método do objeto muda o estado.
 - Exemplo: x.f() f muda o estado de x.
- Objetos oferecem serviços e têm estados.
- Os serviços atuam de forma relevante sobre o seu estado causando uma mudança de estado.
- Propicia implementações interessantes.

@Roberto S. Bigonha e Mariza A.S.Bigonha

Estilo de Programação

Maquinas de Estado

3



Tipo Lista de Inteiros

Lista: IntList:

boolean empty()

informa se a lista está ou não vazia

int numOfElements()

fornece o número de elementos da lista, retorna inteiro

int value(int i)

valor do i-ésimo elemento

void changeValue(int i,int v)

muda valor do i-ésimo da lista para v



Tipo Lista de Inteiros

void insert(int i, int v)

insere valor v após i-ésimo elemento. Se i == 0, insere-o como primeiro

void delete(int i)

remove i-ésimo elemento

int search(int v)}

devolve a posição do elemento de valor v

@Roberto S. Bigonha e Mariza A.S.Bigonha

Estilo de Programação

Maquinas de Estado

_



Estrutura Auxiliar Node

Esta classe é apenas uma estrutura de dados.

```
private static class Node
{
  int value;
  Node right;

  Node(){}
  Node(int value){ this.value = value; }
}
```



Classe IntList

```
class IntList {
    private static class Node {...}
    private Node firstElement;
    private int nElements;

    public IntList( ) { }

    Operações de IntList
}
```

@Roberto S. Bigonha e Mariza A.S.Bigonha

Estilo de Programação

Maquinas de Estado

-



Operações de IntList

```
public boolean empty() {
    return (nElements == 0);
}

public boolean numOfElements() {
    return nElements;
}
```



Operações de IntList

@Roberto S. Bigonha e Mariza A.S.Bigonha

Estilo de Programação

Maquinas de Estado

Λ



Operações de IntList ...



Operações de IntList ...

```
public int search(int v) {
    Node e;
    if vazio() then return 0;
    for (int j = 1, e = firstElement;
        j <= nElements; j++, e = e.right)
        { if (v == e.value) return j }
    return 0;
}</pre>
```

@Roberto S. Bigonha e Mariza A.S.Bigonha

Estilo de Programação

Maquinas de Estado

11



Operações de IntList ...



Operações de IntList ...

@Roberto S. Bigonha e Mariza A.S.Bigonha

Estilo de Programação

Maquinas de Estado

12



Uso da Classe IntLIST



Possíveis Soluções

1. Reescrever search para retornar o endereço do Node e não seu número de ordem.

Desvantagens:

- viola encapsulação
- manipulação direta da representação reduz grau de reúso
- menor nível de abstração

@Roberto S. Bigonha e Mariza A.S.Bigonha

Estilo de Programação

Maquinas de Estado

15



Possíveis Soluções ...

2. Prover primitivas para as suboperações comuns para implementar operações combinadas:

Vantagem: economia de código

Desvantagens:

- proliferação de suboperações específicas
- aumenta o tamanho da interface ao colocar operações deste tipo do lado de fora.
- 3. Mudar a interface do Contrato ou o próprio Contrato.



Objetos como Máquina de Estados

□ A solução proposta corresponde a mudar o contrato
 □ A lista nesta abordagem tem a noção de nodo corrente, o nodo ativo.
 □ Todas as operações têm a noção de estado.
 □ Estado definido pela última operação realizada
 □ Assim, têm-se novas operações:

 inserir a direita do cursor
 inserir a esquerda do cursor
 voltar o valor do elemento referenciado pelo cursor, etc
 □ todas têm custo O(1), exceto search(), que tem custo O(n)

@Roberto S. Bigonha e Mariza A.S.Bigonha

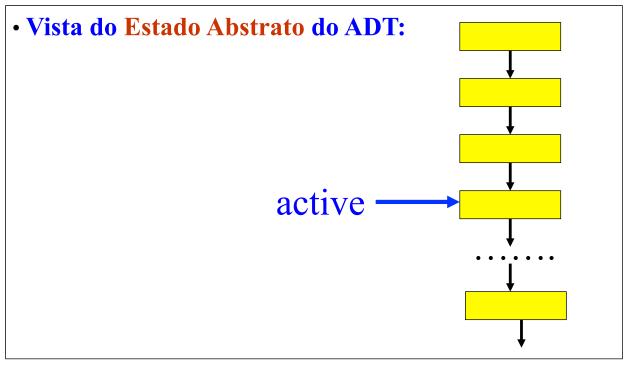
Estilo de Programação

Maquinas de Estado

17

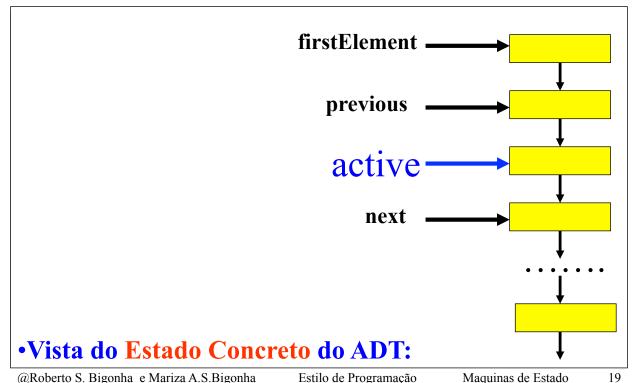


... Objetos como Máquina de Estados





... Objetos como Máquina de Estado





Novo Contrato de IntList

- int numOfElements()
 fornece o número de elementos
- boolean empty()
 informa se a lista está ou não vazia
- void start()
 posiciona o cursor na primeira posição
- void finish()
 posiciona o cursor na última posição



- void forth()avança cursor uma posição
- void back() recua cursor uma posição
- void go(int i) posiciona cursor no i-ésimo elemento

@Roberto S. Bigonha e Mariza A.S.Bigonha

Estilo de Programação

Maquinas de Estado

21



.Novo Contrato de IntList

- boolean offRight() cursor além da última posição?
- boolean offLeft() cursor aquém da primeira posição?
- boolean isFirst() cursor na primeira posição?
- boolean isLast() cursor na última posição?



Novo Contrato de IntList

- □ int value()
 - valor do elemento corrente
- void changeValue(int v)muda valor do elemento corrente
- void insertRight(int v)
 - insere nodo à direita do corrente
- void insertLeft(int v)
 insere nodo à esquerda do corrente

@Roberto S. Bigonha e Mariza A.S.Bigonha

Estilo de Programação

Maquinas de Estado

23



Novo Contrato de IntList

- void insertLeft(int v) insere nodo à esquerda do corrente.
- void delete()
 remove nodo corrente.
- boolean search(int v) move o cursor para elemento de valor v e retorna true; caso contrário, retorna false e deixa offRight() true.
- ...



```
class IntList {
    private static class Node { ... }
    private int nElements, position;
    private Node firstElement, active;
    private Node previous, next;

Operações de IntList
}
```

@Roberto S. Bigonha e Mariza A.S.Bigonha

Estilo de Programação

Maquinas de Estado

25



Estrutura Auxiliar Node

```
private static class Node {
  int value;
  Node right;

  Node(){}
  Node(int value) {
     this.value = value; }
}
```



```
public boolean empty( ){
    return (nElements == 0); }

public int numOfElements( ){
    return nElements; }

public int value( ) throws E1 {
    if (offLeft( ) || offRight( )) throw new E1();
    return active.value ;
}
```

@Roberto S. Bigonha e Mariza A.S.Bigonha

Estilo de Programação

Maquinas de Estado

27



Nova Versão de IntList...

```
public boolean offleft() {
    return (empty() || (position == 0))}

public boolean offRight() {
    return (empty() || (position==nElements+1))}

public boolean isFirst {
    return (position == 1) }

public boolean isLast() {
    return (position == nElements) }
```



```
void start(): posiciona cursor na primeira posição

public void start() throws E2 {
   if (!empty()) {
      previous = null;
      active = firstElement;
      next = active.right;
      position = 1;
   }
}
```

@Roberto S. Bigonha e Mariza A.S.Bigonha

Estilo de Programação

Maquinas de Estado

29



Nova Versão de IntList...

```
void forth(): avança o cursor uma posição

public void forth() throws E3, E4, E5 {
    if (offRight()) throw new E3();
    if (offLeft()){ if (empty()) throw new E4(); start(); }
    else { previous = active; active = next;
        if (next != null) next = next.right;
        position++;
    }
}
```



```
■ void back(): recua o cursor uma posição

public void back () throws E8 {
  if (offLeft()) throw new E8(); go(position - 1);
}

Custo O(1)
```

@Roberto S. Bigonha e Mariza A.S.Bigonha

Estilo de Programação

Maquinas de Estado

31



Nova Versão de IntList...

void go(int i): posiciona o cursor no i-ésimo elemento

```
public void go(int i) throws E6, E7 {
    if (i < 0 || i >= nElements + 1)
        throw new E6();
    if (empty() || (i == 0)) { goOffLeft() }
    else {
        if (position > i) start();
        for (;position!=i && !offRight(); forth());
    }
}
```



■ void goOffLeft(): verifica se o cursor esta aquém da primeira posição, é como se estivesse no inicio da lista

public void goOffLeft() {
 active = null;
 previous = null;
 next = firstElement;
 position = 0;
}

@Roberto S. Bigonha e Mariza A.S.Bigonha

Estilo de Programação

Maquinas de Estado

33



Nova Versão de IntList...

```
boolean search(int v): move o cursor para elemento de
valor v e retorna true; caso contrário, retorna false e
deixa offRight() true.

public boolean search(int v) {
    for (start(); !offRight(); forth()){
        if (active.value == v) {return true};
    }
    return false;
}
```



```
public void insertRight(int v): insere nodo à direita do corrente

public void insertRight(int v) throws E9 {
    Node newNode;
    if (empty() || offRight()) throw new E9()
        newNode = new Node(v);
    insertNodeRight(newNode);
}
Custo O(1)
```

@Roberto S. Bigonha e Mariza A.S.Bigonha

Estilo de Programação

Maquinas de Estado

35

FIM