Tipos de dados abstractos em Java

classes, interface

Paulo Mateus

Departamento de Matemática

IST

2018

Objectivos

- Interface em Java vs tipos de dados abstractos
- Classes
- Implementação de tipos de dados abstractos

Tipos de dados abstractos

- Fila de inteiros (FIFO)
- Enriquecimento sobre os tipos/classes em Java
- Assinatura
- Operações:

nova: → fila
insere: fila, int → fila
primeiro: fila → int
vazia: fila → boolean
retira: fila → fila

Especificação equacional

- Termos de tipo Fila

  Tlista: Conjunto indutivamente definido da seguinte forma:
  - Conjunto de variáveis **X** está contida em Tlista
  - Op:t1, ..., tk ➔ lista
    tj é um termo em Tj para j=1...k, então
    Op(t1,...,tk) é um termo em Tlista

- Obs: Tlista é a álgebra livremente gerada a partir da assinatura e conjunto de variáveis **X.**

- Uma *equação* é uma igualdade entre termos do mesmo tipo

  t1=t2

- Uma especificação equacional é um conjunto de equações.

Especificação equacional de listas

primeiro(insere(nova,N))=N
primeiro(insere(insere(W,N1),N2)=primeiro(insere(W,N1))

vazia(nova)=true
vazia(insere(W,N))=false

retira(insere(nova,N))=nova
retira(insere(insere(W,N1),N2)=insere(retira(insere(W,N1)),N2)

Especificação equacional de listas

primeiro(insere(nova,N))=N
primeiro(insere(insere(W,N1),N2)=primeiro(insere(W,N1))

vazia(nova)=true
vazia(insere(W,N))=false

retira(insere(nova,N))=nova
retira(insere(insere(W,N1),N2)= insere(retira(insere(W,N1)),N1)

Assinatura = interface

```
public interface Fila {

    //(i) Desnecessário implementar a nova pois há sempre um construtor new

    // (ii) Todas as funções que retornam e recebem uma fila devem retornar void
   // e retirar essa fila do argumento
   // Um objecto tem estado  e comportamento, estas funções alteram
   //o estado e não retornam nada.

    void insere(int x);
    int primeiro();
    void retira();
    boolean vazia();

}
```

Implementação de Filas – Estática

```
public class Filaest implements Fila {
        private int[] vec;
        private int pos,max;
        public Filaest(int i){ //construtor nova?
                vec=new int[i];
                max=i;
                pos=0;
        }
        public void insere(int x) {
                if(pos<max){
                        vec[pos]=x;
                        pos++;
                }

        }
```

Implementação de Filas - Estática

```
public int primeiro() {
        if (pos>0) return vec[0];
        return -1;
}

public void retira() { //retira primeiro
        if (pos>0) {
         int i=1;
         while (i<pos-1) {
           vec[i-1]=v[i];
           i++;
         }
         pos--;
        }
}

public boolean vazia() {
        return pos==0;
}

}
```

Implementação de Filas

```
class Node {
        int val;
        Node next;

        Node(int v){
                val=v;
                next=null;
        }
}
```

Implementação de Filas

```
public class Filadyn implements Fila {

        private Node last;

        Filadyn() {
                last=null;
        }
        public void insere(int x) {
                Node n=new Node(x);
                n.next=last;
                last=n;

        }
```