```java
import java.lang.ArrayIndexOutOfBoundsException;
public class Vetor implements VetorD {

	private int valores [];

	private int contador;

	public Vetor() {
		valores = new int[1];
		contador = 0;
	}
	@Override
	public void insereFinal(int valor) {
		//insere proxima posicao livre
		criarEspaco();
		valores[this.contador] = valor;
		contador++;
	}

	@Override
	public void insereInicio(int valor) {
		//insere no inicio do vetor
		inserePosicao(0,valor);
	}

	@Override
	public void inserePosicao(int ind, int valor)  {
		//insere na posicao descrita  na

			if(isIndexValid(ind)) {

				criarEspaco();

				moveFrente(ind);

				valores[ind] = valor;

				contador++;
			}


	}


	@Override
	public void get(int ind) {
		if(isIndexValid(ind)) {
			System.out.println("int: "+valores[ind]);
		}

	}

	@Override
	public void set(int ind, int valor) {
		if(isIndexValid(ind)) {
```

```
                valores[ind] = valor;
        }

}

@Override
public void removeFinal() {
        contador--;
}

private void moveTras(int index) {

        if ( isIndexValid ( index ) ) {
          for (int i = index + 1; i < contador; i++) {
              valores[i - 1] = valores[i];
          }
        }
}

@Override
public void removeInicio() {
        int ini = 0;
        moveTras(ini);


}

@Override
public void remove(int indice) {
        if(isIndexValid(indice)) {
                moveTras(indice);
                contador--;

        }

}

@Override
public int size() {
        //retorna o numero de elementos armazenados no vetor
        return contador;

}

@Override
public int capacity() {
        // retorna a capacidade de armazenamento do vetor.
        return (valores.length-1);

}

@Override
public boolean isEmpty() {
        // retorna "true" se o número de elementos for 0 (zero).
        return (contador == 0 );
```

```
	}

	@Override
	public void criarEspaco() {
		//Duplica o tamanho do array.
		if(contador == valores.length -1) {
			int aux[] = new int[valores.length*2];
			for(int i =0 ; i< valores.length; i++) {
				aux[i]=valores[i];
			}
			valores = aux;

		}

	}

	public void moveFrente(int index) {
	    if ( isIndexValid ( index ) ) {

	        for (int i = contador; i > index; i = i- 1) {
	            valores[i] = valores[i - 1];
	        }
	    }
	   }

	public boolean isIndexValid ( int index ) {
		try {
			if(index<0) {throw new ArrayIndexOutOfBoundsException();}

			return ( index >= 0  ) &&  ( index < valores.length );


		}catch( ArrayIndexOutOfBoundsException e) {

			System.out.println("Erro: " + e);

			return false;
		}
	}
}


//----------------------------------------------------------------------Classe aplicação


public class Aplicacao {

	public static void main(String[] args) {
		long tempoInicial = System.currentTimeMillis();

		Vetor t;
		t= new Vetor();
		for (int i = 0 ; i < 100000; i++) {
			t.insereFinal(i);
		}
```

```
            for(int i = 0; i<t.size();i++) {
                  t.get(i);
            }
            System.out.println(t.capacity());
            System.out.println("fim");

            long tempoFinal = System.currentTimeMillis();

            System.out.printf("%.3f ms%n", (tempoFinal - tempoInicial) /
1000d);
      }

}
```

Tempo de Inserção 1,463 ms