

**Listas**

Lista de tipos primitivos:

```
int c[];
```

```
c = new int[ 12 ];
```

```
double d[] = new double[ 30 ];
```

```
int b[] = { 3, 5, 7, 9 };
```

```
int b[] = new int[ 4 ];
```

```
b[ 0 ] = 3;
```

```
b[ 1 ] = 5;
```

```
b[ 2 ] = 7;
```

```
b[ 3 ] = 9;
```

## Construção

## Lista de objetos:

```
Label llb[] = new Label[ 3 ];  
    llb[ 0 ] = new Label( "Vetor a" );  
    llb[ 1 ] = new Label( "Vetor b" );  
    llb[ 2 ] = new Label( "Produto" );
```

**Construção**

Utilização do comprimento em um laço:

```
for( int i = 0; i < llb.length; i++){  
    llb[ i ].setBackground( Color.blue );  
}
```

## Referência e valor

Listas são objetos e, na linguagem Java, objetos são sempre passados aos métodos por referência, ao passo que tipos primitivos são sempre passados por valor

# Comprimento

Matrizes:

```
int b[][];
```

```
b = new int[ 2 ][ 3 ];
```

```
int c[][] = { { 1, 2, 3 }, { 4, 5, 6 } };
```

Mais gerais:

```
int b[][];
```

```
b = new int[ 2 ][ ];
```

```
b[ 0 ] = new int[ 2 ];
```

```
b[ 1 ] = new int[ 3 ];
```

```
int c[][] = { { 1, 2 }, { 3, 4, 5 } };
```

## Índices múltiplos

Faça um aplicativo que leia um vetor de 8 posições inteiras e exiba-o contando quantos números pares foram inseridos.

Utilize a classe Keyboard.java anexada abaixo para capturar o teclado:

[Keyboard.java](#)

**Prática**