

7. laboratorijas darbs

Ir matrica M un divi pavedieni: **Even** un **Odd**.

Pavediens **Even** apstrādā matricas rindiņas ar indeksiem 0, 2, ...

Pavediens **Odd** apstrādā matricas rindiņas ar indeksiem 1, 3, ...

Pavedienu klases konstruktora parametri:

1. Matrica M.
2. Pirmās apstrādājamās rindiņas indekss `StartRow`. Vērtības: 0, 1.
3. Pavediena vārds `Name`.
4. Pavediena prioritāte `Priority`.
5. Aizture pēc kārtējas rindiņas apstrādes `Delay`.

Iespējama matrica M, kas deklarēta `main()` metodē:

1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7

Iespējamā pavedienu radīšana `main()` metodē:

```
new MatrThread(M, 0, "Even", 2, 0);  
new MatrThread(M, 1, "Odd", 8, 0);
```

1. daļa: Abi pavedieni meklē elementu summas *matricas rindiņās*.

Programmas rezultāti (abu pavedienu prioritāte ir 2):

```
Thread[Even,2,main]. Row: 0, Sum: 8.  
Thread[Odd,2,main]. Row: 1, Sum: 16.  
Thread[Even,2,main]. Row: 2, Sum: 24.  
Thread[Odd,2,main]. Row: 3, Sum: 32.  
Thread[Even,2,main]. Row: 4, Sum: 40.  
Thread[Odd,2,main]. Row: 5, Sum: 48.  
Thread[Even,2,main]. Row: 6, Sum: 56.
```

Programmas rezultāti (pavediena **Odd** prioritāte ir 8, pavediena **Even** prioritāte ir 2):

```
Thread[Odd,8,main]. Row: 1, Sum: 16.  
Thread[Odd,8,main]. Row: 3, Sum: 32.  
Thread[Odd,8,main]. Row: 5, Sum: 48.  
Thread[Even,2,main]. Row: 0, Sum: 8.  
Thread[Even,2,main]. Row: 2, Sum: 24.  
Thread[Even,2,main]. Row: 4, Sum: 40.  
Thread[Even,2,main]. Row: 6, Sum: 56.
```

2. daļa: abi pavedieni meklē elementu summas *savā apstrādājamā matricas daļā*.
Rezultātā no divām summām tiks iegūta *visu matricas elementu* summa.

Programmas rezultāti:

```
Thread[Odd,8,main]. Sum: 96.  
Thread[Even,2,main]. Sum: 128.  
Total sum: 224.
```

Piezīme: uzdevuma 2. daļā pavedienu klasē ieteicams izveidot metodes `getThread()` un `getSum()`.