

## **Java nuoseklių procesų komunikavimas-2 (Java CSP)**

# Programų pavyzdžiai

## Funkcijos reikšmių siuntimas

```
Guard[] g = {kinA, kinB, kinC};
Alternative alt = new Alternative(g);
a = b = c = 0; n = false;
for ( ; ; ) {
    switch (alt.Select()) {
        case 0:  a = kinA.read(); n = true; break;
        case 1:  b = kinB.read(); n = true; break;
        case 2:  c = kinC.read(); n = true; break;
    }
    if (n)
        { r = f(a, b, c); kout.write(r); }
}
```

## Dekoratyvinis sodas (1)

```
public class DekSodas {  
    final static int N = ...; // vartų skaičius  
    valdProc valdymas;  
    VartuProc[] vartProcesai = new VartuProc[N];  
    Parallel visi = new Parallel();  
    Any2OneChannelInt vartai;  
    public void vykdyti() {  
        visi.addProcess(valdymas);  
        visi.addProcess(vartProcesai);  
        visi.run();  
    }  
    public static void main(String[] args) {  
        new DekSodas().vykdyti();  
    } }  
}
```

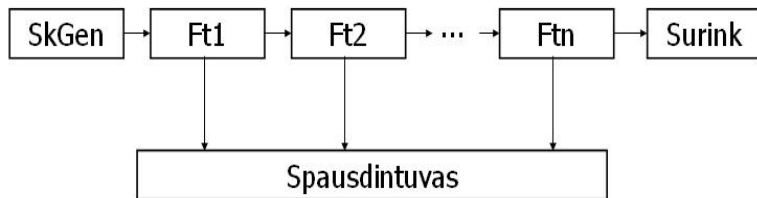
## Dekoratyvinis sodas (2)

```
class VartuProc implements CProcess {  
    private final ChannelOutputInt v;  
    private final int kiekKartų, kiekLeisti;  
    ...  
    public void run() {  
        for(int i=0; i<kiekKartų; i++)  
            v.write(kiekLeisti);  
        v.write(-1);  
    } }  
}
```

## Dekoratyvinis sodas (3)

```
class valdProc implements CProcess {  
    ChannelInputInt vartai;  
    int N; int bendrasKiekis;  
    ...  
    public void run() {  
        int uždaryta = 0; bendrasKiekis = 0;  
        while (uždaryta < N) {  
            int lank = vartai.read();  
            if (lank < 0) uždaryta++;  
            else bendrasKiekis += lank;  
        } } }
```

# Pirminiai skaičiai (1)



## Pirminiai skaičiai (2)

```
import org.jcsp.lang.*;

class Filtras implements CSPProcess { ... }
class SkaičiųGeneratorius implements CSPProcess
    { ... }
class Spausdintuvas implements CSPProcess
    { ... }
class Surinkėjas implements CSPProcess { ... }

public class PirminiaiSkaičiai { ... }

public static void main(String[] args) {
    new PirminiaiSkaičiai().run();
}
```



## Pirminiai skaičiai (3a)

```
class Filtr implements CProcess {  
    private final ChannelInputInt išKairės;  
    private final ChannelOutputInt dešinė;  
    private final ChannelOutputInt žemyn;  
  
    Filtr(ChannelInputInt išKairės,  
          ChannelOutputInt dešinė,  
          ChannelOutputInt žemyn) {  
        this.išKairės = išKairės;  
        this.dešinė = dešinė;  
        this.žemyn = žemyn;  
    }  
  
    public void run() { ... }  
}
```

## Pirminiai skaičiai (3b)

```
// class Filtras
public void run() {
    int n, m;
    n = išKairės.read(); žemyn.write(n);
    for (;;) {
        m = išKairės.read();
        if ((m % n) != 0) dešinė.write(m);
    }
}
```

## Pirminiai skaičiai (4)

```
class SkaičiųGeneratorius implements CSPProcess {  
    private final ChannelOutputInt dešinèn;  
  
    SkaičiųGeneratorius(ChannelOutputInt dešinèn)  
    {  
        this.dešinèn = dešinèn;  
    }  
  
    public void run() {  
        for (int sk=2; ;sk++) dešinèn.write(sk);  
    }  
}
```

## Pirminiai skaičiai (5)

```
class Spausdintuvas implements CProcess {  
    private final ChannelInputInt[] išViršaus;  
    private int p, N;  
  
    Spausdintuvas( ... ) {  
        ...  
    }  
  
    public void run() {  
        for (int i=0; i<N; i++) {  
            p = išViršaus[i].read();  
            ... // p - pirminis skaičius  
        }  
    }  
}
```

## Pirminiai skaičiai (6)

```
class Surinkėjas implements CSPProcess {  
    private final ChannelInputInt išKairės;  
  
    Surinkėjas( ... ) {  
        ...  
    }  
  
    public void run() {  
        for (;;) int sk = išKairės.read();  
    }  
}
```

## Pirminiai skaičiai (7a)

```
class PirminiaiSkaiciai {  
  
    final static int N = ...; // pirminių sk.kiekis  
    ... dešinèn = new One2OneChannelInt[N+1];  
    ... žemyn    = new One2OneChannelInt[N];  
    private final Parallel rasti;  
  
    PirminiaiSkaiciai() {  
        ... // Kanalo sukūrimas  
        ... // Procesų sukūrimas  
    }  
  
}
```

## Pirminiai skaičiai (7b)

```
// Kanalių sukūrimas
for (int i = 0; i < N+1; ++i)
    dešinèn[i] = Channel.one2oneInt();
for (int i = 0; i < N; ++i)
    žemyn[i] = Channel.one2oneInt();
... žemynIn = new AltingChannelInputInt[N];
for (int i = 0; i < N; ++i)
    žemynIn[i] = žemyn[i].in();
```

## Pirminiai skaičiai (7c)

```
// Procesų sukūrimas
... sg = new SkaičiųGeneratorius(dešinėn[0].out());
... sur = new Surinkėjas(dešinėn[N].in());
... sp = new Spausdintuvas(N, žemynIn);

... filtrai = new Filtras[N];
for (int i = 0; i < N; ++i)
    filtrai[i] = new Filtras(dešinėn[i].in(),
                             dešinėn[i+1].out(), žemyn[i].out());

rasti = new Parallel (
    new CSPProcess[] {
        sg, sur, sp,
        new Parallel(filtrai)
    }
);
```



## Matricos ir vektoriaus sandauga (1)

**$B = A \times X$ , kur:**

**$A = A(n, n); \quad X = X(n); \quad B = B(n).$**

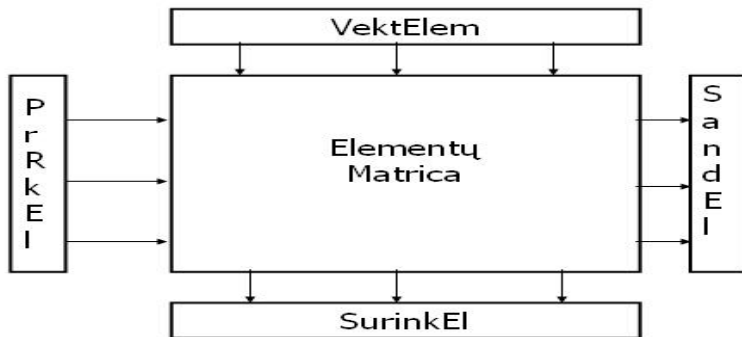
**$B[1] = \Sigma (A[1, j] \times X[j], j = 1, 2, \dots, n);$**

**$B[2] = \Sigma (A[2, j] \times X[j], j = 1, 2, \dots, n);$**

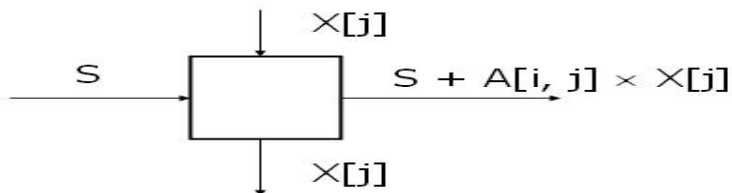
**...**

**$B[n] = \Sigma (A[n, j] \times X[j], j = 1, 2, \dots, n).$**

## Matricos ir vektoriaus sandauga (2)



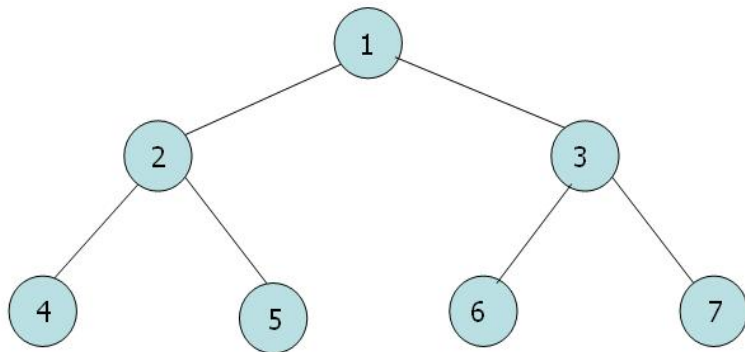
## Matricos ir vektoriaus sandauga (3)



## Matricos ir vektoriaus sandauga (4)

Programa?

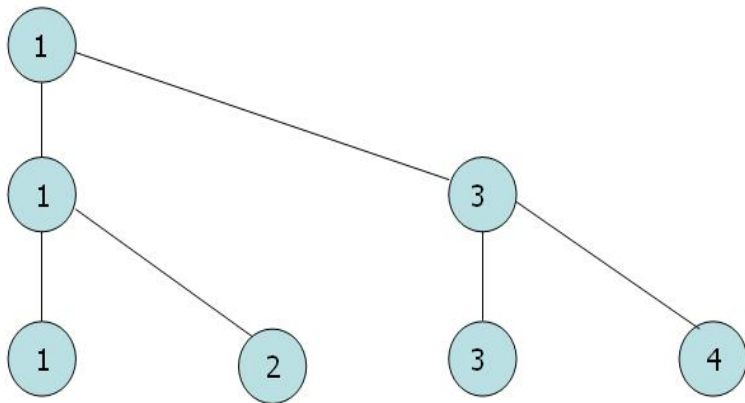
# Masyvo rikiavimas (1 variantas)



## Masyvo rikiavimas (1 variantas)

Programa?

## Masyvo rikiavimas (2 variantas)



## Masyvo rikiavimas (2 variantas)

Programa?



# Procesų nutraukimas

## Parallel procesų rinkinys

- baigia darbą, kai baigiami visi rinkinio procesai;
- neturi metodų, kurie galėtų nustatyti, ar kuris nors procesas dar "gyvas";
- dažniausiai turi procesą(-us), kurio(-ių) veikimas apibrėžia kitų procesų darbo trukmę;
- apie tai, kad reikia baigti darbą, procesas gali pranešti žinute.

## Pirminiai skaičiai: su procesų nutraukimu (1)

```
class Spausdintuvas implements CProcess {  
    ...  
    private final ChannelOutputInt bg;  
  
    Spausdintuvas( ... ) {  
        ...  
    }  
  
    public void run() {  
        for (int i=0; i<N; i++) {  
            p = išViršaus[i].read();  
            ... // p - pirminis skaičius  
        }  
        bg.write(0);  
    }  
}
```

## Pirminiai skaičiai: su procesų nutraukimu (2a)

```
class SkaičiųGeneratorius implements CSPProcess {  
    private final AltingChannelInputInt bg;  
    ...  
    final Skip skip = new Skip();  
    final Guard[] guards = {bg, skip};  
    ... alt = new Alternative (guards);  
    int sk = 2; boolean baigti = false;  
    ...  
}
```

## Pirminiai skaičiai: su procesų nutraukimu (2b)

```
while (!baigti) {  
    switch (alt.priSelect()) {  
        case 0:  
            dešinèn.write(bg.read()); baigti = true;  
            break;  
        case 1:  
            dešinèn.write(sk); sk++; break;  
    }  
}
```

## Pirminiai skaičiai: su procesų nutraukimu (3)

```
// class Filtras
public void run() {
    int n, m = 10;
    n = išKairės.read();
    if (n > 0) {
        žemyn.write(n);
        while (m > 0) {
            m = išKairės.read();
            if ((m==0) || ((m%n) !=0))
                dešinèn.write(m);
        }
    }
    else dešinèn.write(0);
}
```

## Pirminiai skaičiai: su procesų nutraukimu (4)

```
class Surinkėjas implements CProcess {  
    ...  
    public void run() {  
        int sk = 10;  
        while (sk > 0) sk = išKairės.read();  
    }  
}
```

# Kanalų modeliai. Išvados

- Naudojami aktyvūs ir pasyvūs procesai.
- Kanalas – apibrėžto tipo tarpininkas tarp dviejų procesų.
- Komunikavimas ir sinchronizavimas – viena konstrukcija.
- Duomenų perdavimas viena kryptimi.
- Komunikavimas "one-to-one": galima sužinoti, kas siuntė.
- Komunikavimas "many-to-one": galima supaprastinti programos kodą.



# Klausimai pakartojimui

- 1 Kuo pasižymi any2one kanalai?
- 2 Kuo pasižymi one2one kanalai?
- 3 Kiek procesų dalyvauja pirminių skaičių generavime?
- 4 Kiek procesų dalyvauja matricos ir vektoriaus daugyboje?
- 5 Koks pagrindinis skirtumas tarp pateiktų masyvo rikiavimo variantų?
- 6 Koku būdu procesai gali pranešti apie darbo pabaigą?
- 7 Palyginkite one-to-one ir any-to-one komunikavimo variantus: kada kuris geriau tinka?