

## SKAITĻU TEORIJAS ALGORITMS 38.nodarbība

### LIELĀKAIS KOPIĢAIS DALĪTĀJS (LKD)

- Par veselu skaitļu lielāko kopīgo dalītāju sauc lielāko veselo skaitli, ar kuru dotie skaitļi dalās bez atlikuma.
- Piemēram  $LKD(9,15)=3$
- Atrašana:
  - Sadalījums pirmreizinātājos
  - Eiklīda algoritms
  - ...

### EIKLĪDA ALGORITMS (VĀRDISKAIS)

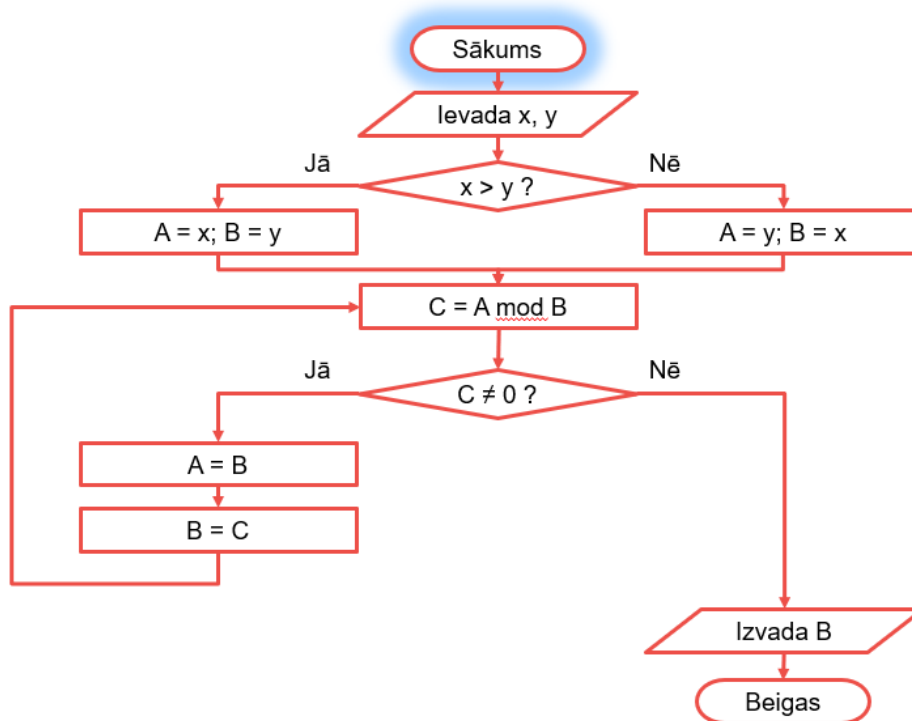
**Soli:**

1. Piešķir mainīgajam A lielāko un B mazāko skaitli
2. Dali A ar B, iegūstot atlikumu C
3. Ja C nav vienāds ar 0, tad:
  - a) A piešķir B vērtību
  - b) B piešķir C vērtību
  - c) atgriezies uz 2. soli
4. LKD ir mainīgajā B esošā vērtība

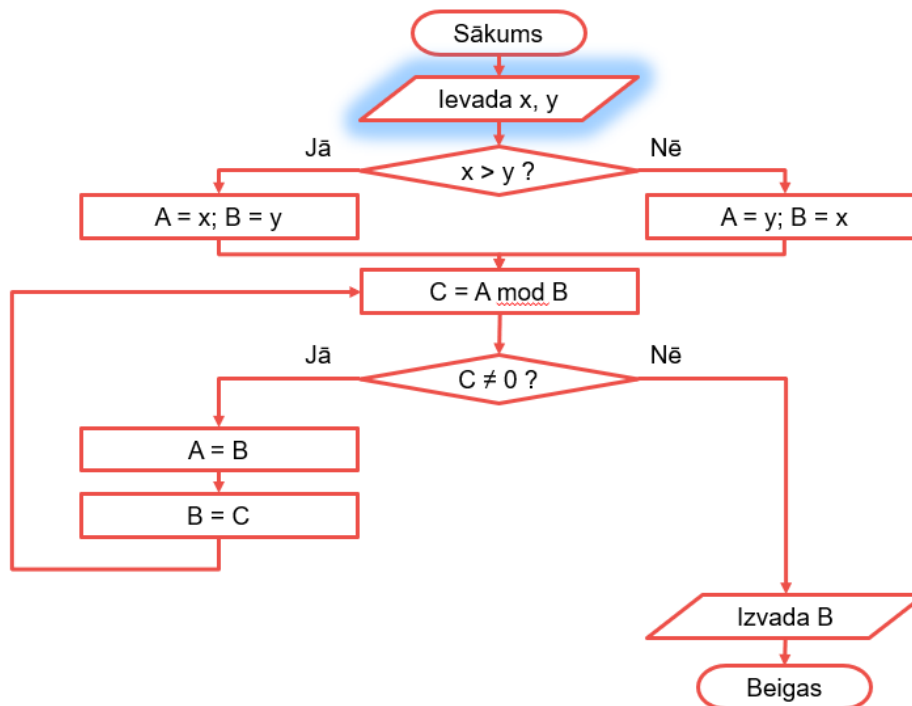
Solis	A	B	C
1.	15	9	
2.			6
3.	9	6	
2.			3
3.	6	3	
2.			0
3.			
4.	Atbilde = 3		

## EIKLĪDA ALGORITMS (BLOKSHĒMA, 1. VARIANTS)

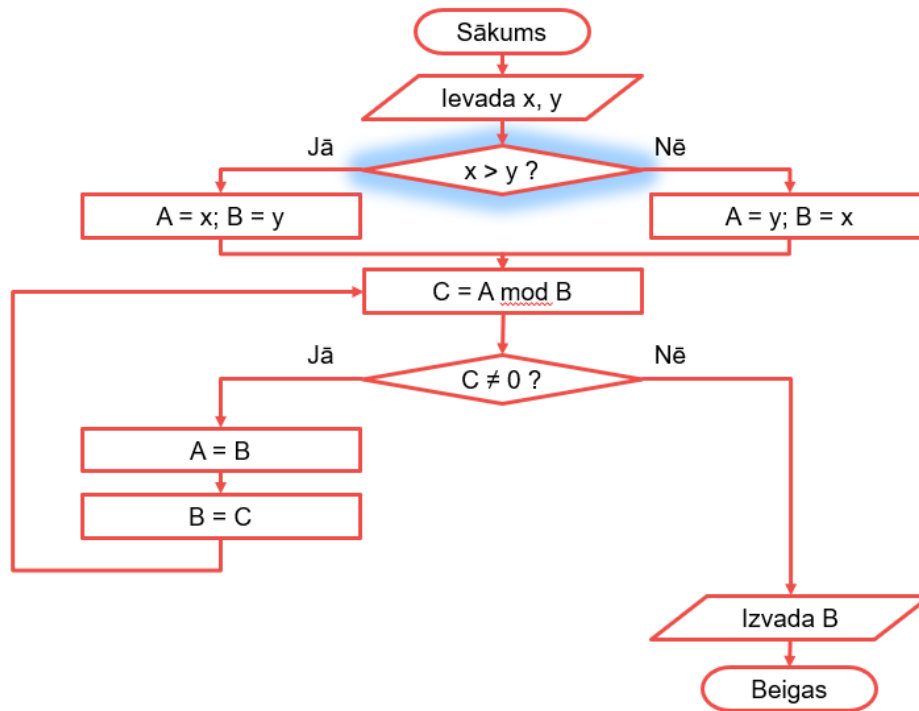
1.



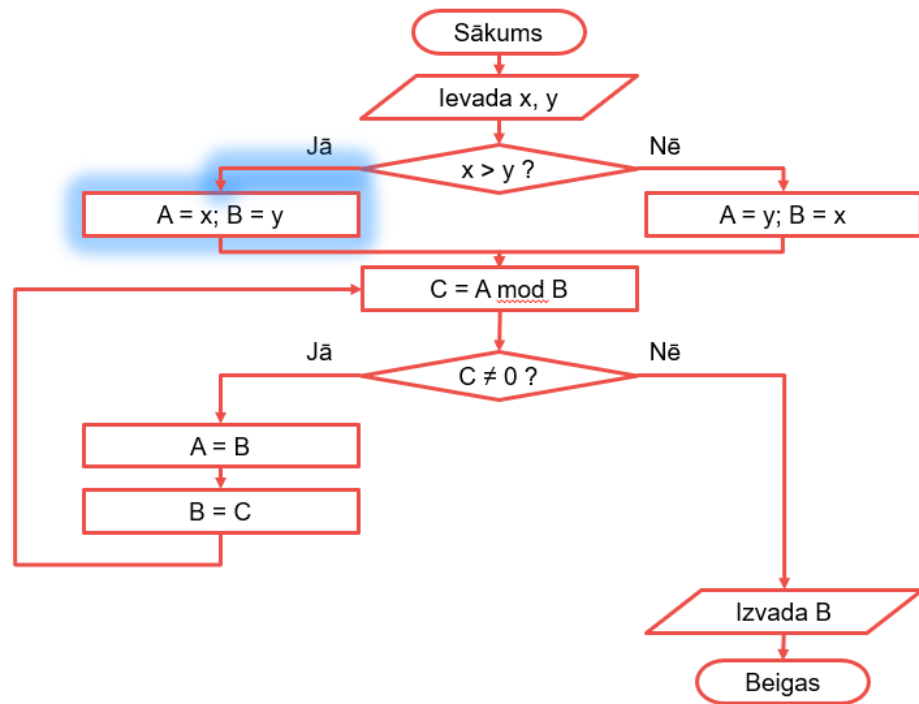
2.



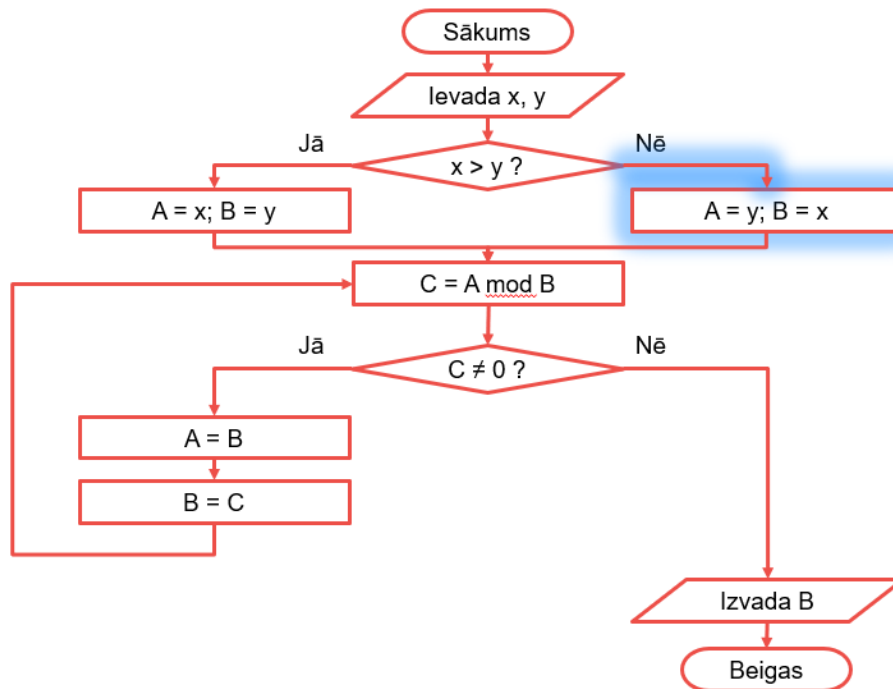
3.



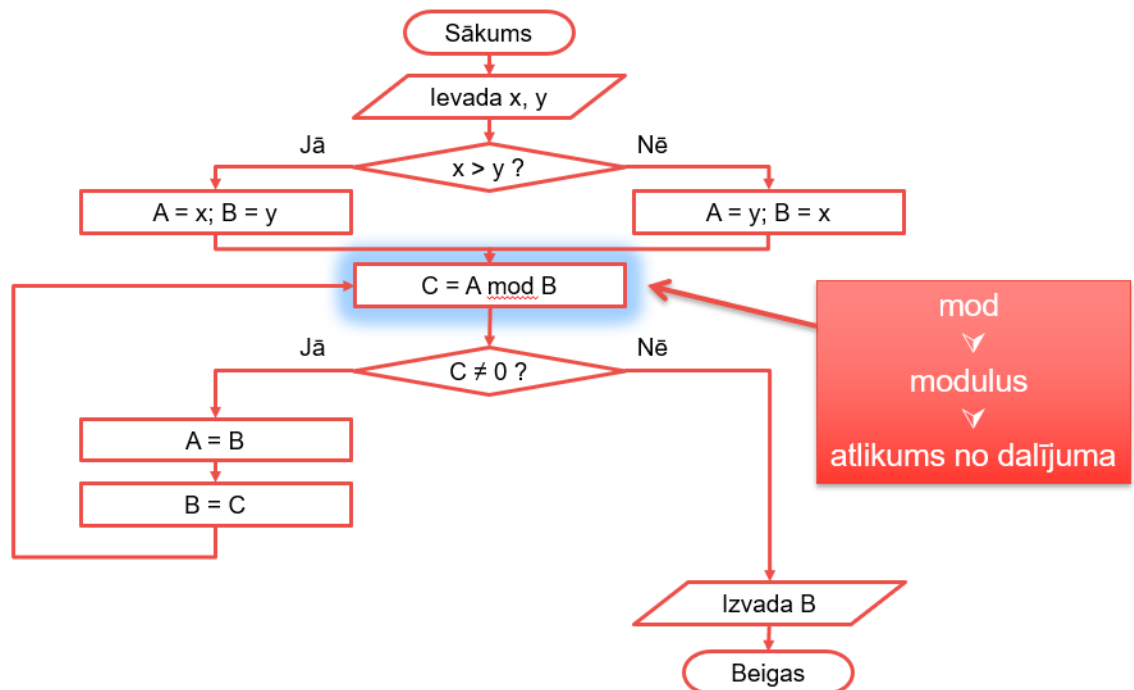
4.



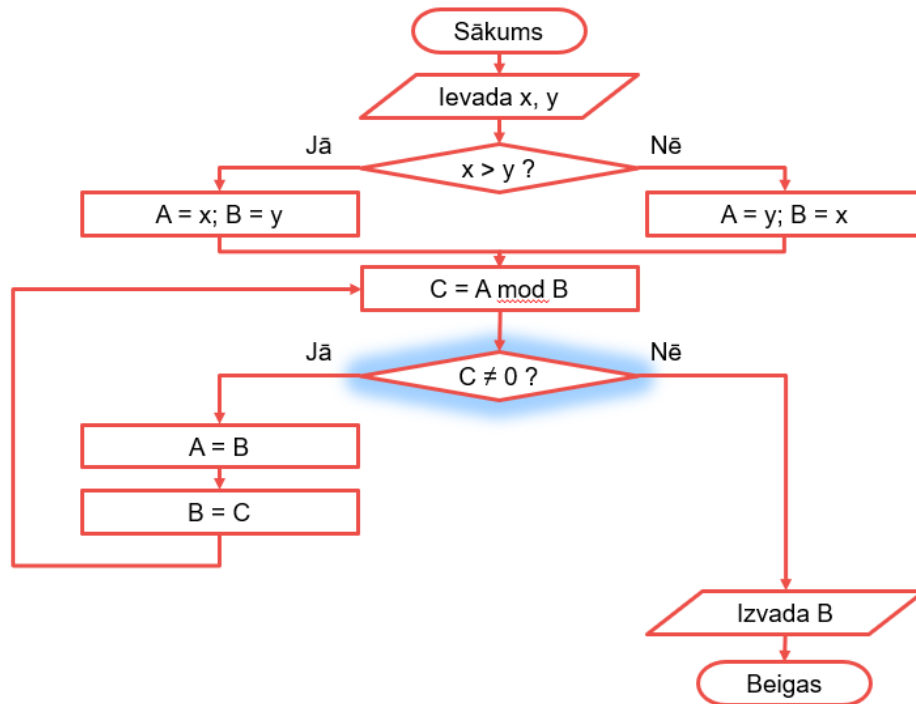
5.



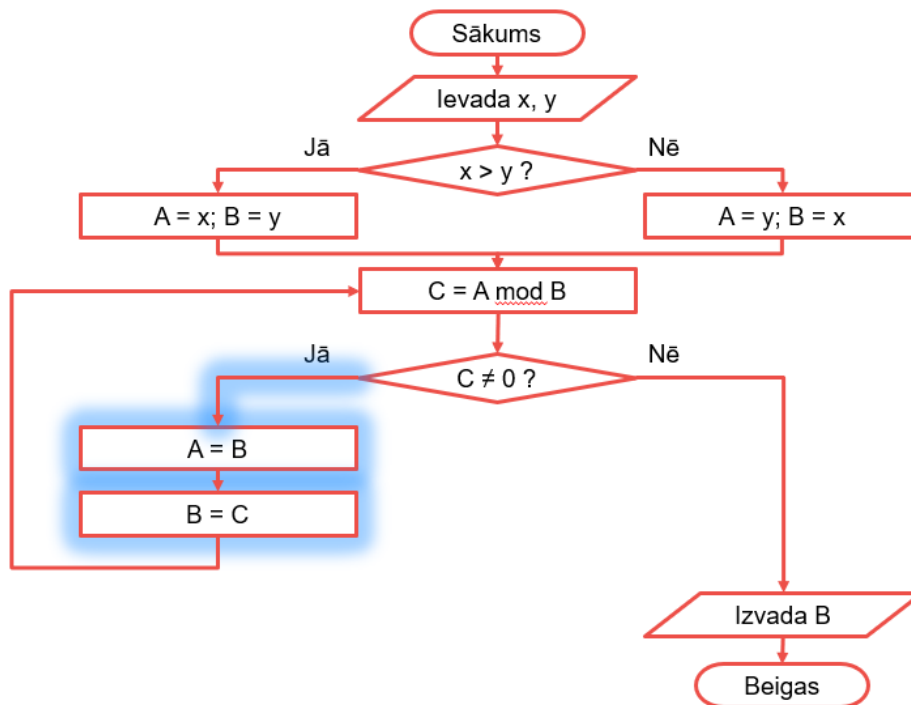
6.



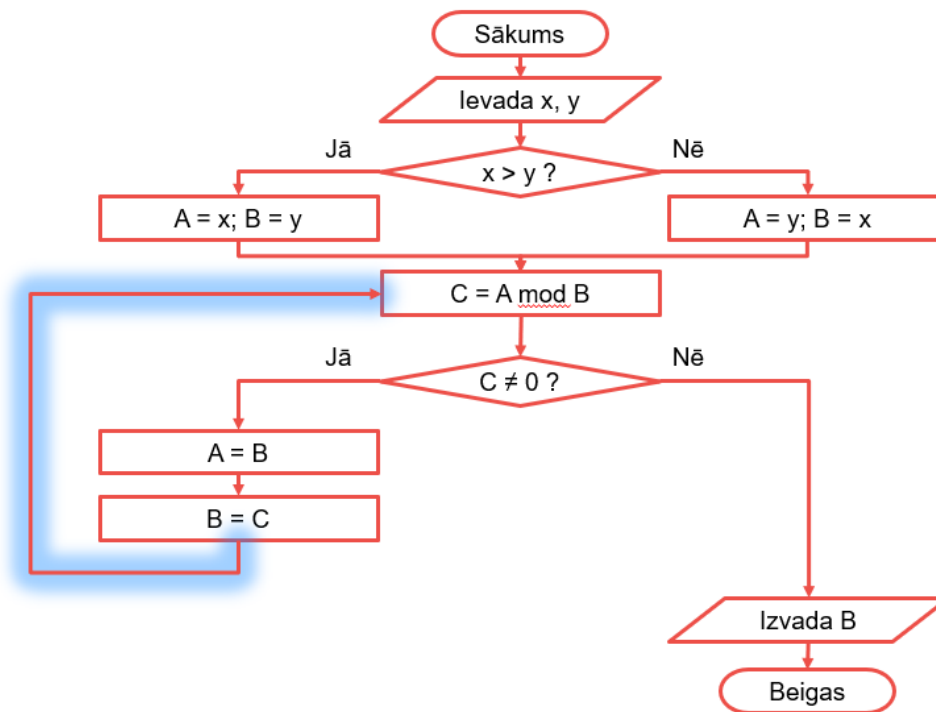
7.



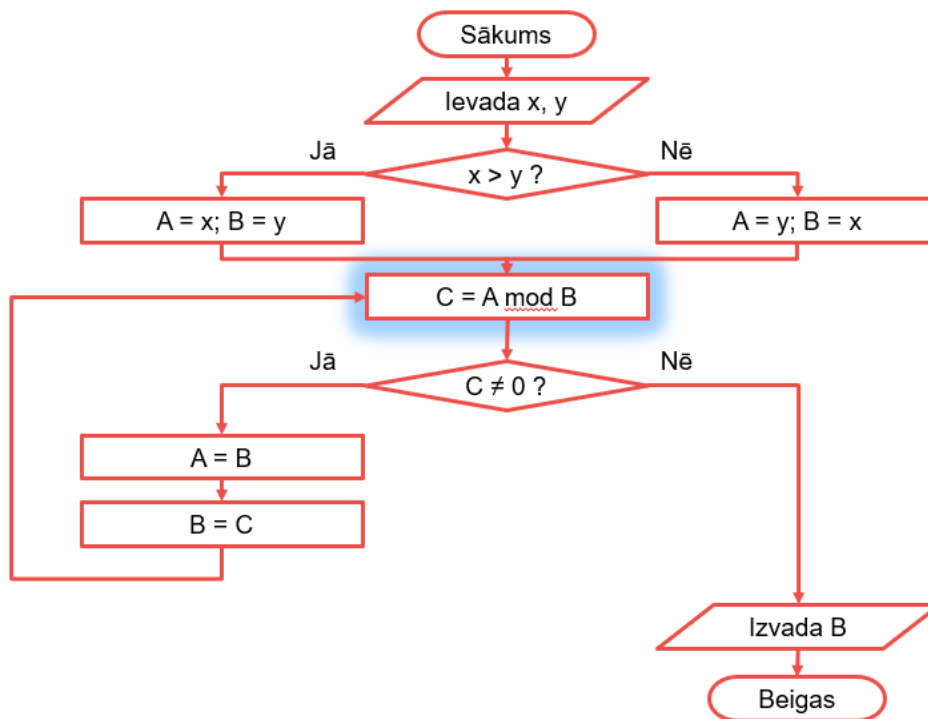
8.



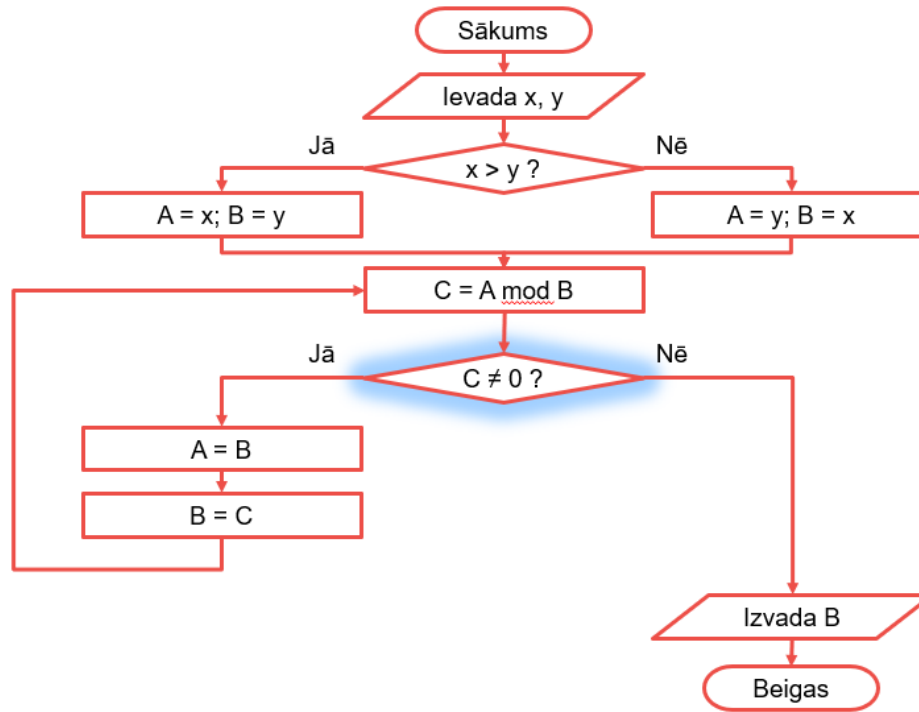
9.



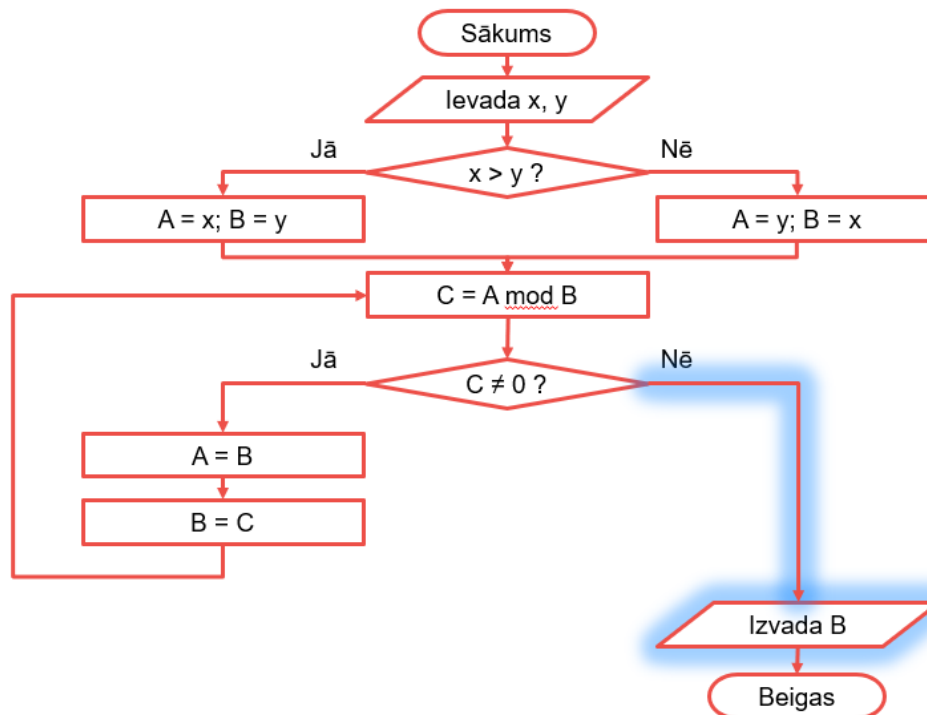
10.



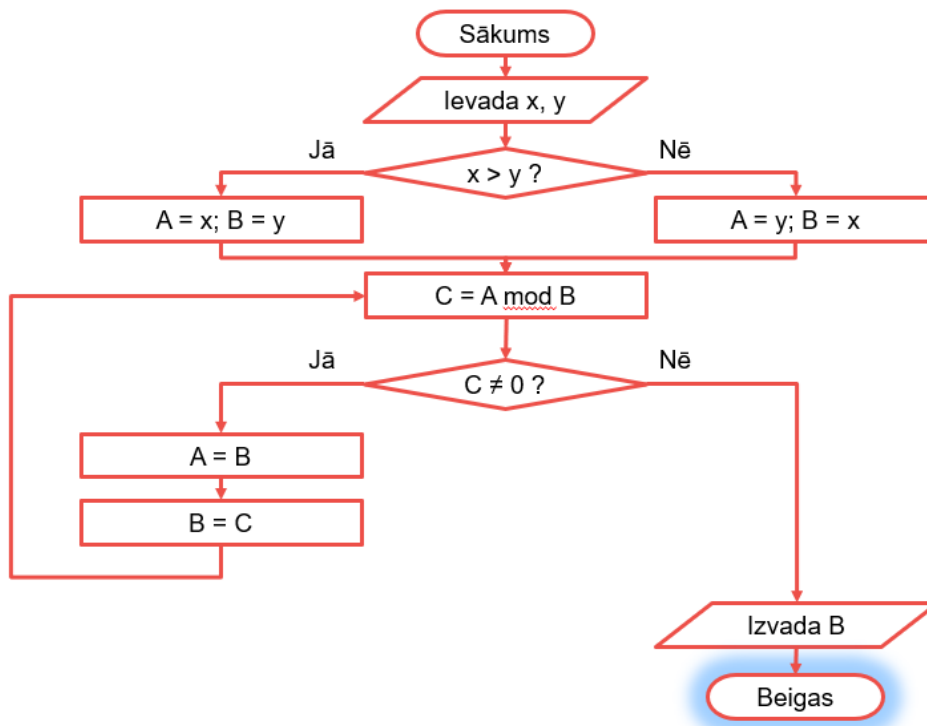
11.



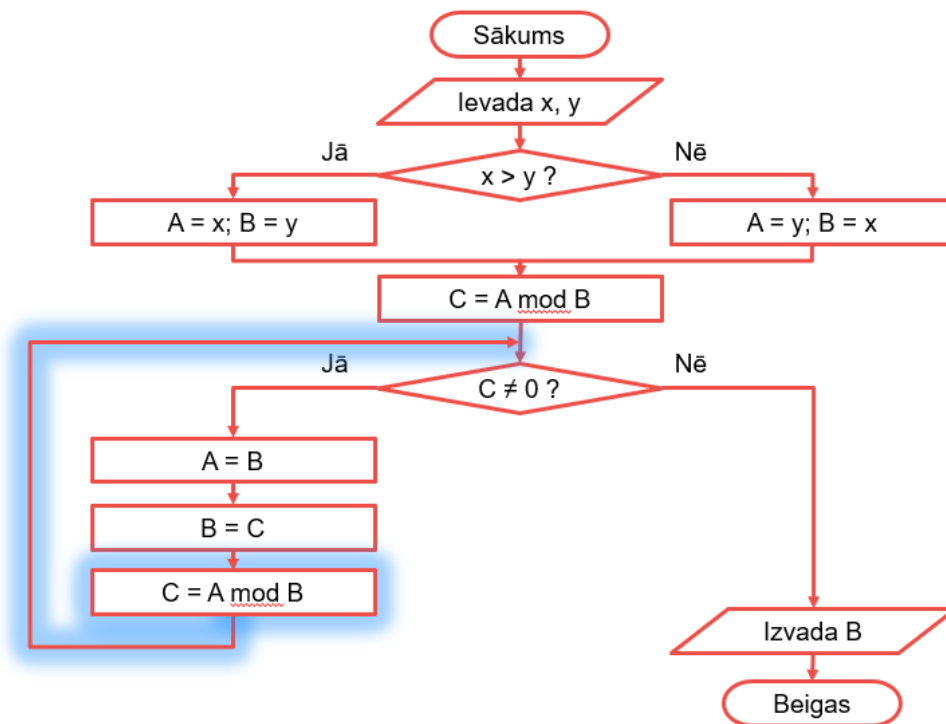
12.



13.



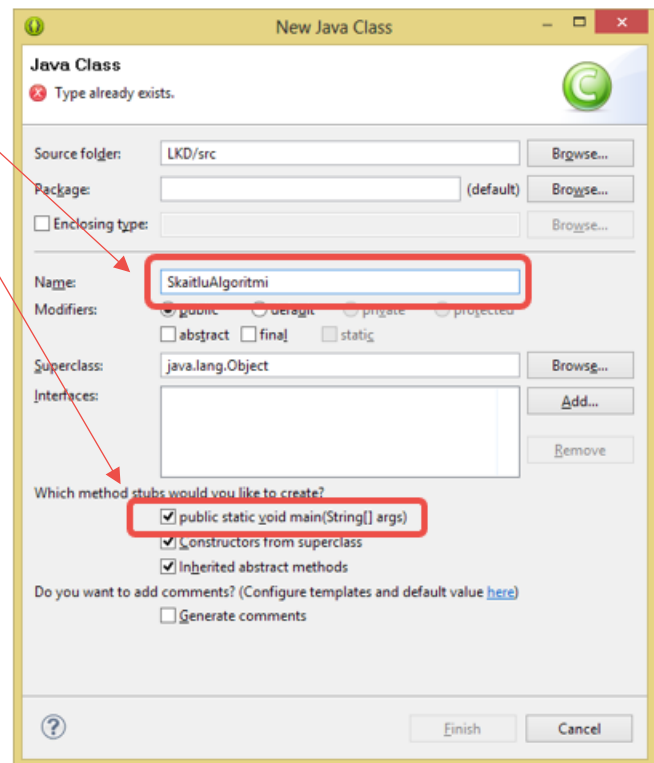
14.





## PROGRAMMA

- Jauns Java konsoles projekts
- Jauna klase SkaitluAlgoritmi
- Atzīmējam, ka veidosim metodi main()



## EIKLĪDA ALGORITMS (JAVA KODS)

```

int lkd( int x, int y ) { // metode lkd() saņem 2 skaitļus
    int a, b, c;
    if( x > y ) { // mainīgajā a jāieliek lielākais skaitlis
        a = x;
        b = y;
    } else {
        a = y;
        b = x;
    }
    c = a % b; // atlikums no dalīšanas (modulus operator)
    while( c != 0 ) { // atkārti, kamēr c nav vienāds 0
        a = b;
        b = c;
        c = a % b;
    }
    return b; // metode atgriež lielāko kopīgo dalītāju
}
    
```

## METODES LKD() IZSAUKŠANA

```
public static void main(String[] args) {
    int sk1, sk2, sk3;
    // Scanner lieto datu ievadišanai no konsoles
    Scanner ievade = new Scanner(System.in);
    // izveido savas klases instanci
    SkaitluAlgoritmi skAlg = new SkaitluAlgoritmi();
    // ievada divus skaitļus
    System.out.println("Programma rēķina LKD");
    System.out.print("Ievadi pirmo skaitli: ");
    sk1 = Integer.parseInt( ievade.next() );
    System.out.print("Ievadi otro skaitli: ");
    sk2 = Integer.parseInt( ievade.next() );
    // aprēķina LKD, izsaucot mūsu metodi
    sk3 = skAlg.lkd( sk1, sk2 );
    // izvada rezultātu konsolē
    System.out.println( "LKD = " + sk3 );
    // aizver ievadišanu no konsoles
    if( ievade != null )
        ievade.close();
}
```

Eclipse:  
Source/Organize Imports  
jeb  
Ctrl+Shift+O

## TESTĒJAM

- Vai pamanījāt, ka programmā ir kļūda?
- Ievadi programmā skaitļus 5 un 0
- Kāds bija rezultāts?
  - Exception in thread "main" java.lang.ArithmeticException: / by zero
  - at SkaitluAlgoritmi.lkd(SkaitluAlgoritmi.java:45)
  - at SkaitluAlgoritmi.main(SkaitluAlgoritmi.java:26)
- `c = a % b; // kļūda, ja b=0`
- `LKD(0, n) = n`

## LABOJAM KĻŪDU (DALĪŠANA AR NULLI)

```
int lkd( int x, int y ) { // metode lkd() saņem 2 skaitļus
    int a, b, c;
    if( x == 0 && y == 0 ) return 0;
    if( x == 0 ) return y;
    if( y == 0 ) return x;
    if( x > y ) { // mainīgajā a jāieliek lielākais skaitlis
        a = x;
        b = y;
    } else {
        a = y;
        b = x;
    }
    c = a % b; // atlikums no dalīšanas (modulus operator)
    while( c != 0 ) { // atkārtoti, kamēr c nav vienāds 0
        a = b;
        b = c;
        c = a % b;
    }
    return b; // metode atgriež lielāko kopīgo dalītāju
}
```

## DAĻSKAITĻU SAĪSINĀŠANA

- Daļskaitļus saīsina dalot skaitītāju un saucēju ar lielāko kopīgo dalītāju
- Piemēram:
  - dots daļskaitlis  $\frac{285}{513}$
  - aprēķinot LKD, iegūst skaitli 57
  - dala skaitītāju un saucēju ar LKD
  - iegūst saīsinātu daļu  $\frac{5}{9}$

```
public static void main(String[] args) {
    int sk1, sk2, sk3;
    Scanner ievade = new Scanner(System.in);
    SkaitluAlgoritmi skAlg = new SkaitluAlgoritmi();
    System.out.println("Programma saīsina daļskaitli");
    System.out.print("Ievadi skaititaju: ");
    sk1 = Integer.parseInt( ievade.next() );
```

```
System.out.print("Ievadi sauceju: ");
sk2 = Integer.parseInt( ievade.next() );
// aprēķina LKD, izsaucot mūsu metodi
sk3 = skAlg.lkd( sk1, sk2 );
if( sk3 != 0 ) { // dala tikai tad, ja LKD nav 0
    sk1 = sk1 / sk3;
    sk2 = sk2 / sk3;
}
System.out.println( "LKD = " + sk3 );
System.out.println( "Saisinata dala = " + sk1 + "/" + sk2 );
if( ievade != null )
    ievade.close();
}
```