

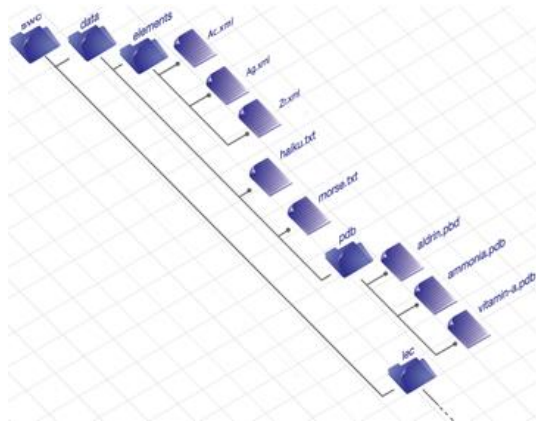


# 8. prednáška (9.11.2011)

## Midterm a súbory prakticky

alebo

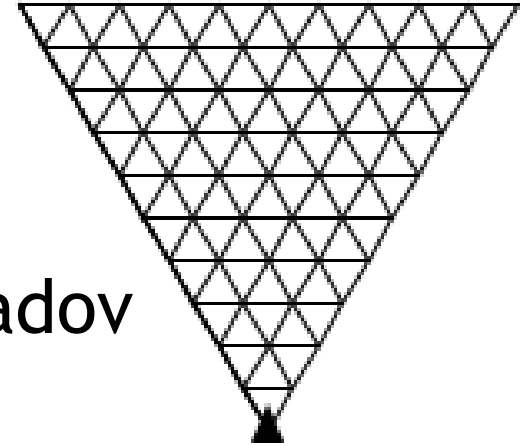
Vstupy zo súborov  
sú super!





# Midterm - Pyramída

- Do triedy `Midtermarka` pridajte metódu `pyramida`, ktorá nakreslí pyramídu z rovnostranných trojuholníkov so zadaným počtom radov a so zadanou dĺžkou strany malých rovnostranných trojuholníkov. Korytnačka sa na začiatku nachádza v jednej zo strán trojuholníka a je nasmerovaná v smere výšky pyramídy.



```
public void pyramida(int pocetRadov, double strana)
```



## Midterm – Exponent rozkladu

- Z matematiky je známe, že každé celé číslo  $n$  vieme jednoznačne zapísať v tvare  $a * 2^i$ , kde  $a$  je nepárne číslo. Do triedy `Midtermarka` pridajte metódu `exponentRozkladu`, ktorá pre zadané číslo vráti exponent  $i$  (pri dvojke) v tomto jedinečnom rozklade - t.j. exponent najväčšej mocniny dvojky, ktorá delí zadané číslo.
- *Napríklad pre číslo 100 má metóda vrátiť 2 lebo  $100 = 25 * 2^2$ , pre číslo 27 má vrátiť 0 pretože  $27 = 27 * 2^0$ , pre číslo 40 číslo 3 pretože  $40 = 5 * 2^3$ . Môžete predpokladať, že parametrom je kladné číslo.*

```
public int exponentRozkladu(int cislo)
```



# Midterm – Spoločný prefix

- Prefixom reťazca nazývame ľubovoľný jeho podreťazec, ktorým tento reťazec začína. Napríklad reťazec "skola" má tieto prefixy: "", "s", "sk", "sko", "skol", "skola".
- Do triedy `Midtermarka` pridajte metódu `spolocnyPrefix`. Táto metóda dostane ako parametre referencie dva reťazce a vráti najdlhší taký reťazec (referenciu na taký reťazec), ktorý je prefixom oboch zadaných reťazcov.
- Príklady:
  - najdlhší spoločný prefix reťazcov "promocia" a "program" je reťazec "pro"
  - najdlhší spoločný prefix reťazcov "kolovratok" a "kolo" je reťazec "kolo"
  - najdlhší spoločný prefix reťazcov "program" a "kolo" je reťazec "",

```
public String spolocnyPrefix(String r1, String r2)
```



# Midterm – Korytnačí salaš

- Vytvorte triedu `MidtermPane`, ktorá rozširuje triedu `WinPane`. Po vytvorení kresliacej plochy triedy `MidtermPane` nech sa v nej vytvorí 8 korytnačiek triedy `Midtermarka` (alebo triedy `Turtle`) na náhodných pozíciách.
- Do triedy `MidtermPane` pridajte metódu `vsetkyVKosiari`, ktorá vráti, či **všetky** korytnačky v kresliacej ploche sa nachádzajú vo vnútri obdĺžnika rovnobežného s kresliacou plochou a ktorého ľavý horný roh má súradnice  $(x, y)$ , jeho šírka je  $s$  a výška  $v$ .

```
public boolean vsetkyVKosiari(double x, double y, double s, double v)
```



# Práca s textovými súbormi

- Práca s **obsahom textových súborov** sa vždy skladá z 3 krokov.
  - otvorenie súboru, ktoré sa udeje pri vzniku nejakého čítača (napr. `Scanner`) alebo zapisovača (napr. `PrintWriter`).
  - práca s obsahom súboru, teda čítanie alebo zapisovanie
  - zatvorenie súboru



# Práca s textovými súbormi

- Prácu s textovým súborom vždy realizujeme v rámci **try-catch** bloku
- **Musíme** odchytať možnú výnimku `FileNotFoundException`
  - Vyhodí sa, aj keď existuje adresár s týmto menom
  - Pri zápise: Keď adresár v ktorom má nový súbor vzniknúť neexistuje
  - Pri čítaní: Keď súbor ktorý ideme čítať neexistuje
- V bloku **finally** zatvárame súbor !



# Schéma práce s *PrintWriter*-om

```
File subor = new File("C:\\adresare\\subor");
PrintWriter pw = null;
try {
    pw = new PrintWriter(subor);

    // píšeme do pw

} catch (FileNotFoundException e) {
    System.out.println("Súbor " +
        subor.getName() + " som nenašiel");
} finally {
    if (pw != null)
        pw.close();
}
```





# Schéma práce so Scanner-om pri čítaní zo súboru

```
File subor = new File("C:\\adresare\\subor");
Scanner scanner = null;
try {
    scanner = new Scanner(subor);

    // čítame zo scannera

} catch (FileNotFoundException e) {
    System.out.println("Súbor " +
        subor.getName() + " som nenašiel");
} finally {
    if (scanner != null)
        scanner.close();
}
```



# Čítame maticu zo súboru

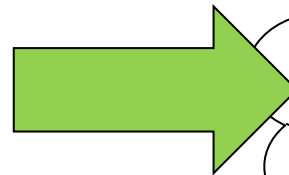
- Vytvorme si metódu, ktorá načíta zo súboru maticu čísiel.
  - V prvom riadku je uvedený počet riadkov a počet stĺpcov

3 4

-1 456 2 0

1 2 1 1

3000 5 6 -7



-1	456	2	0
1	2	1	1
3000	5	6	-7



# Čítame maticu zo súboru

```
public int[][] nacistajMaticu(File subor) {
    Scanner citacSuboru = null;
    try {
        citacSuboru = new Scanner(subor);
        int riadkov = citacSuboru.nextInt();
        int stlpcov = citacSuboru.nextInt();
    }
}
```



náš program

objekty

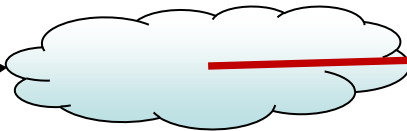
reálny svet

subor



citacSuboru

null



Name	Ext	Size	Modified	rwx
..	<DIR>			
bin	<DIR>		11/05/11 01:08 PM	rw-
src	<DIR>		11/11/10 11:37 AM	rw-
matica2	txt	40 B	11/11/10 10:57 AM	rw-
<b>matica</b>	txt	34 B	11/07/11 07:56 AM	rw-
zoznam	txt	162 B	11/11/10 11:14 AM	rw-

file:///home/gursky/ec  
 File Edit View Bookmarks Tools Settings KrView  
 Save Save As Undo Redo Print  
 matica.txt (Editing) x  
 3 4  
 -1 456 2 0  
 1 2 1 1  
 3000 5 6 -7



# Čítame maticu zo súboru

```
public int[][] nacistajMaticu(File subor) {
    Scanner citacSuboru = null;
    try {
        citacSuboru = new Scanner(subor);
        int riadkov = citacSuboru.nextInt();
        int stlpcov = citacSuboru.nextInt();
    }
}
```



náš program

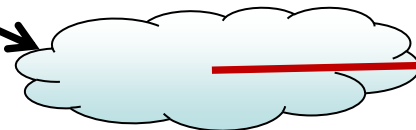
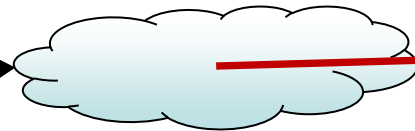
objekty

reálny svet

subor



citacSuboru



Name	Ext	Size	Modified	rwx
..	<DIR>			
bin	<DIR>	11/05/11 01:08 PM		rwx
src	<DIR>	11/11/10 11:37 AM		rwx
matica2	txt	40 B	11/11/10 10:57 AM	rw-
matica	txt	34 B	11/07/11 07:56 AM	rw-
zoznam	txt	162 B	11/11/10 11:14 AM	rw-

file:///home/gursky/ec

File Edit View Bookmarks Tools Settings KrView

Save Save As Undo Redo Print

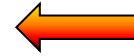
matica.txt (Editing) x

```
34
-1 456 2 0
1 2 1 1
3000 5 6 -7
```



# Čítame maticu zo súboru

```
public int[][] nacistajMaticu(File subor) {
    Scanner citacSuboru = null;
    try {
        citacSuboru = new Scanner(subor);
        int riadkov = citacSuboru.nextInt();
        int stlpcov = citacSuboru.nextInt();
        int[][] pole = new int[riadkov][stlpcov];
        for (int r=0; r < riadkov; r++)
            for (int s=0; s < stlpcov; s++)
                pole[r][s] = citacSuboru.nextInt();
        return pole;
    }
}
```

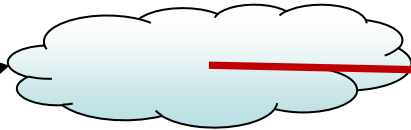


náš program

citacSuboru



objekty



reálny svet

3	4			
-1	4	5	6	2 0
1	2	1	1	
3	0	0	5	6 -7



# Čítame maticu zo súboru

```
public int[][] nacistajMaticu(File subor) {
    Scanner citacSuboru = null;
    try {
        citacSuboru = new Scanner(subor);
        int riadkov = citacSuboru.nextInt();
        int stlpcov = citacSuboru.nextInt();
        int[][] pole = new int[riadkov][stlpcov];
        for (int r=0; r < riadkov; r++)
            for (int s=0; s < stlpcov; s++)
                pole[r][s] = citacSuboru.nextInt();
        return pole;
    }
}
```



náš program

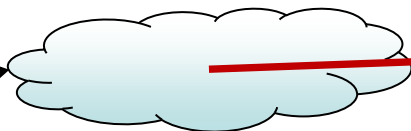
citacSuboru



riadkov



objekty



reálny svet

3	4			
-1	4	5	6	2 0
1	2	1	1	
3	0	0	5	6 -7



# Čítame maticu zo súboru

```
public int[][] nacistajMaticu(File subor) {
    Scanner citacSuboru = null;
    try {
        citacSuboru = new Scanner(subor);
        int riadkov = citacSuboru.nextInt();
        int stlpcov = citacSuboru.nextInt();
        int[][] pole = new int[riadkov][stlpcov];
        for (int r=0; r < riadkov; r++)
            for (int s=0; s < stlpcov; s++)
                pole[r][s] = citacSuboru.nextInt();
        return pole;
    }
}
```



náš program

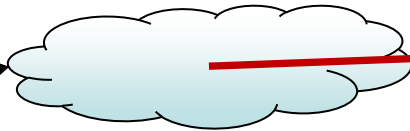
citacSuboru



riadkov stlpcov



objekty



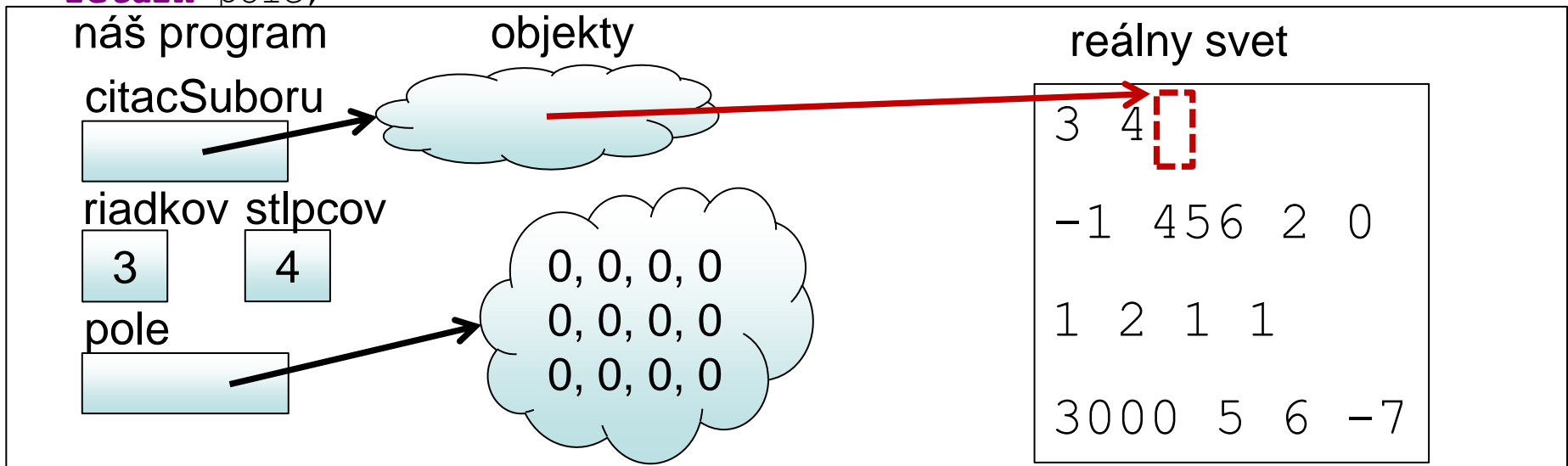
reálny svet

3	4					
-1	4	5	6	2	0	
1	2	1	1			
3	0	0	0	5	6	-7



# Čítame maticu zo súboru

```
public int[][] nacistajMaticu(File subor) {
    Scanner citacSuboru = null;
    try {
        citacSuboru = new Scanner(subor);
        int riadkov = citacSuboru.nextInt();
        int stlpcov = citacSuboru.nextInt();
        int[][] pole = new int[riadkov][stlpcov]; ←
        for (int r=0; r < riadkov; r++)
            for (int s=0; s < stlpcov; s++)
                pole[r][s] = citacSuboru.nextInt();
        return pole;
    }
}
```

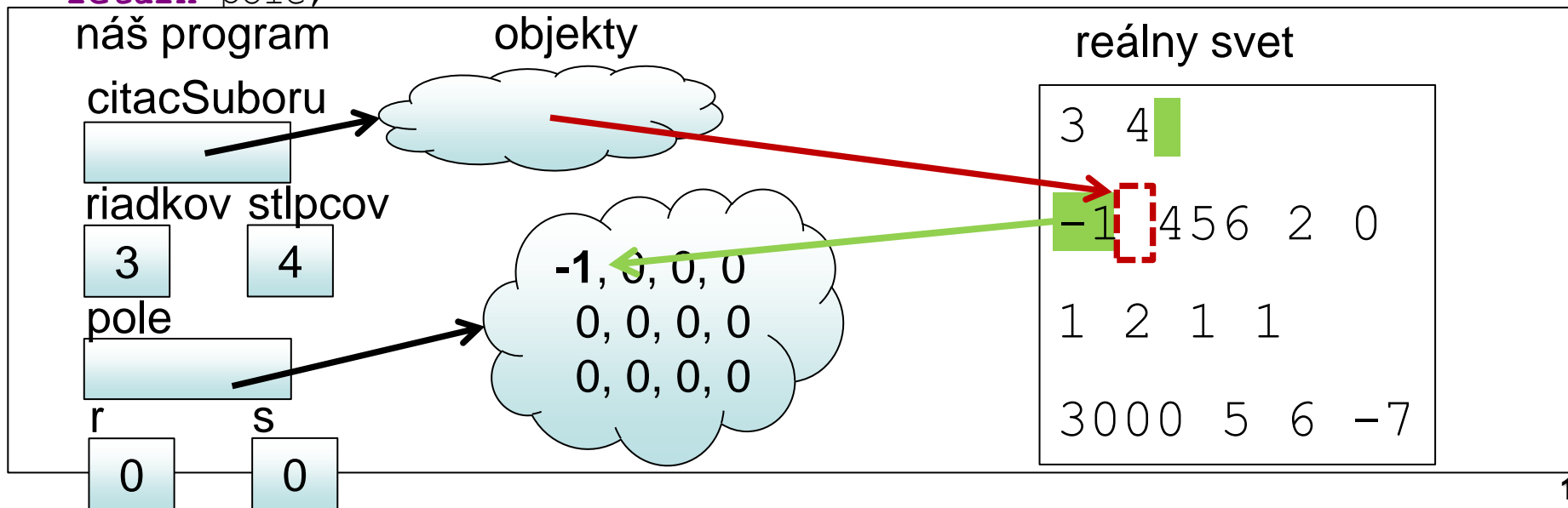






# Čítame maticu zo súboru

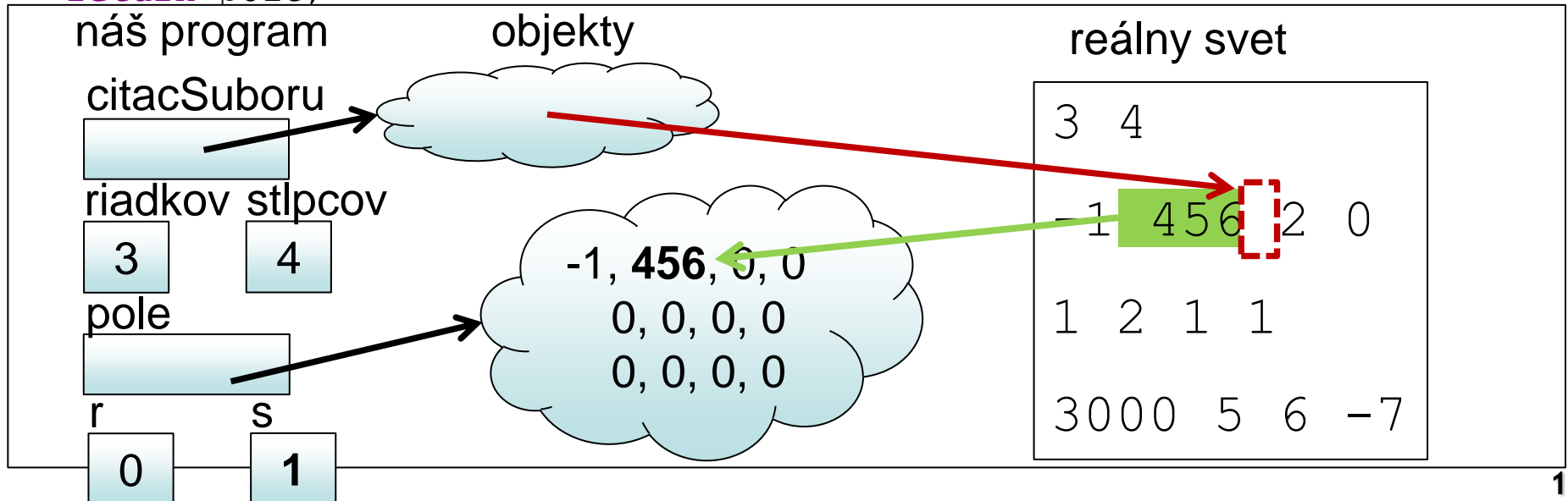
```
public int[][] nacistajMaticu(File subor) {
    Scanner citacSuboru = null;
    try {
        citacSuboru = new Scanner(subor);
        int riadkov = citacSuboru.nextInt();
        int stlpcov = citacSuboru.nextInt();
        int[][] pole = new int[riadkov][stlpcov];
        for (int r=0; r < riadkov; r++)
            for (int s=0; s < stlpcov; s++)
                pole[r][s] = citacSuboru.nextInt();
    }
    return pole;
}
```





# Čítame maticu zo súboru

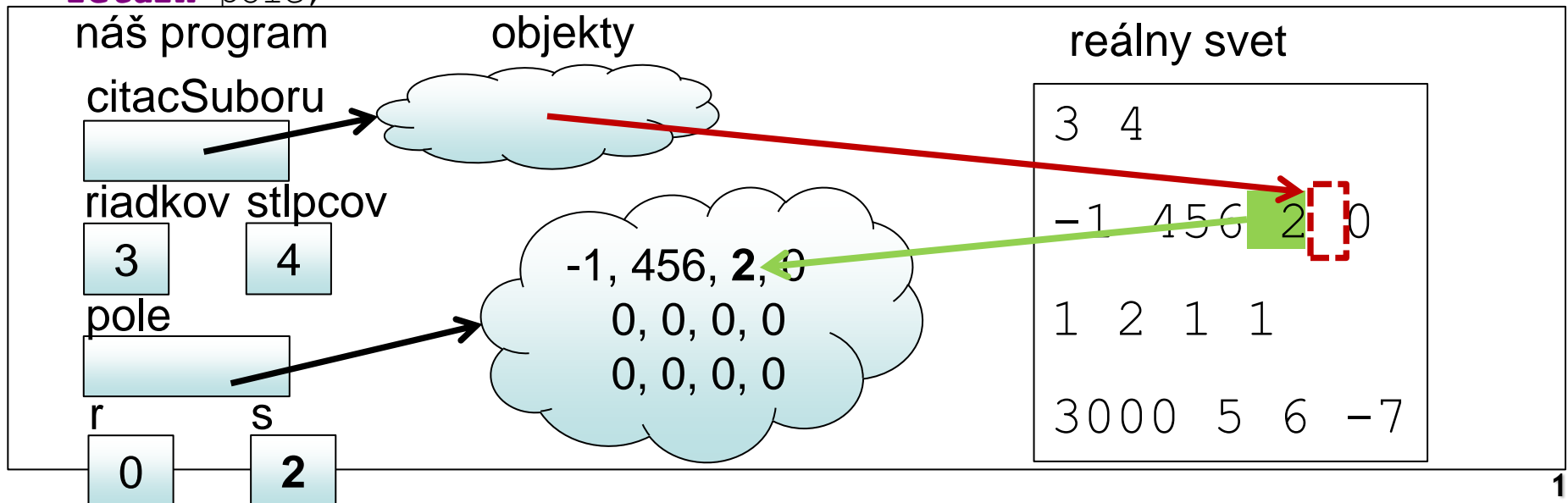
```
public int[][] nacistajMaticu(File subor) {
    Scanner citacSuboru = null;
    try {
        citacSuboru = new Scanner(subor);
        int riadkov = citacSuboru.nextInt();
        int stlpcov = citacSuboru.nextInt();
        int[][] pole = new int[riadkov][stlpcov];
        for (int r=0; r < riadkov; r++)
            for (int s=0; s < stlpcov; s++)
                pole[r][s] = citacSuboru.nextInt();
        return pole;
    }
}
```





# Čítame maticu zo súboru

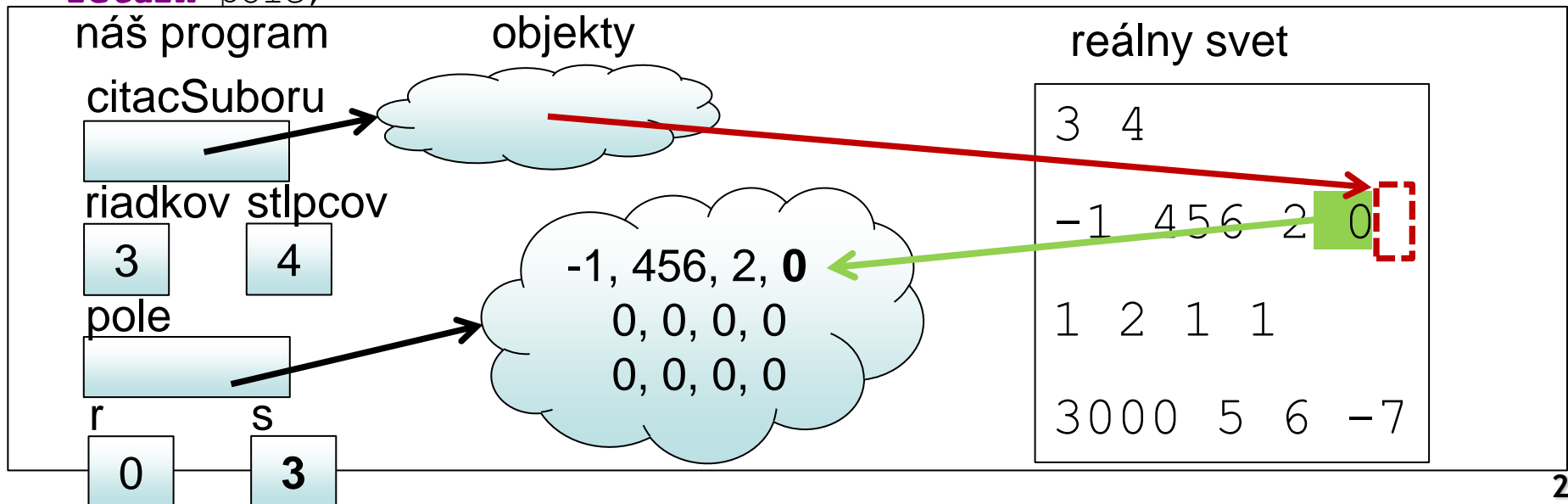
```
public int[][] nacistajMaticu(File subor) {
    Scanner citacSuboru = null;
    try {
        citacSuboru = new Scanner(subor);
        int riadkov = citacSuboru.nextInt();
        int stlpcov = citacSuboru.nextInt();
        int[][] pole = new int[riadkov][stlpcov];
        for (int r=0; r < riadkov; r++)
            for (int s=0; s < stlpcov; s++)
                pole[r][s] = citacSuboru.nextInt();
    }
    return pole;
}
```





# Čítame maticu zo súboru

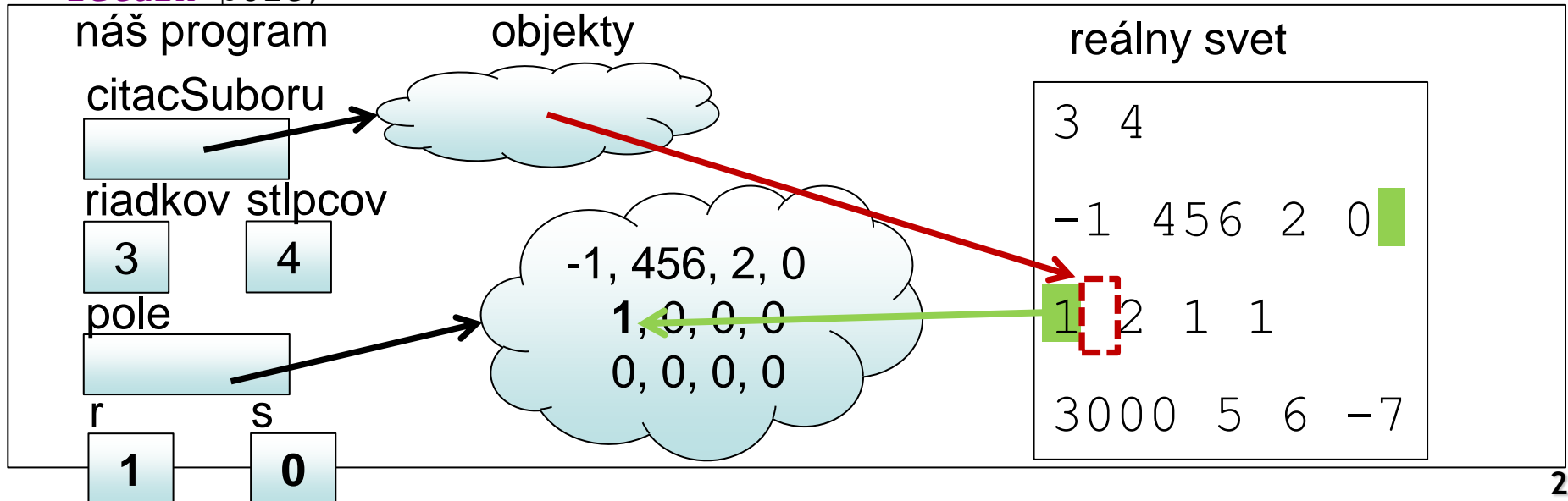
```
public int[][] nacistajMaticu(File subor) {
    Scanner citacSuboru = null;
    try {
        citacSuboru = new Scanner(subor);
        int riadkov = citacSuboru.nextInt();
        int stlpcov = citacSuboru.nextInt();
        int[][] pole = new int[riadkov][stlpcov];
        for (int r=0; r < riadkov; r++)
            for (int s=0; s < stlpcov; s++)
                pole[r][s] = citacSuboru.nextInt();
    }
    return pole;
}
```





# Čítame maticu zo súboru

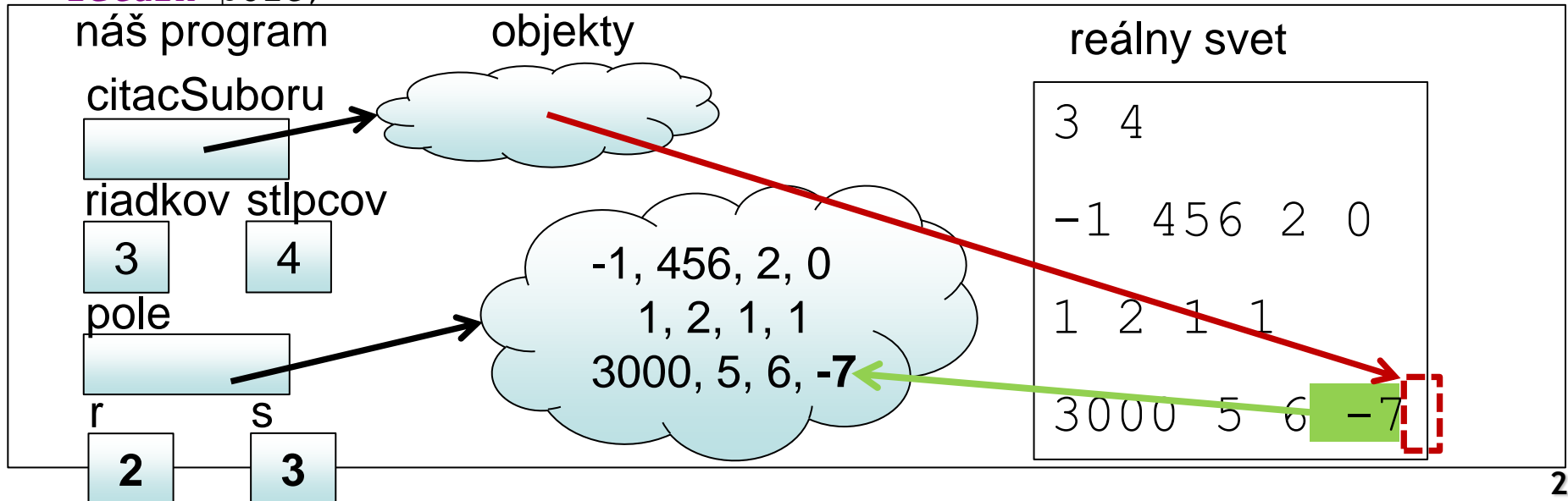
```
public int[][] nacistajMaticu(File subor) {
    Scanner citacSuboru = null;
    try {
        citacSuboru = new Scanner(subor);
        int riadkov = citacSuboru.nextInt();
        int stlpcov = citacSuboru.nextInt();
        int[][] pole = new int[riadkov][stlpcov];
        for (int r=0; r < riadkov; r++)
            for (int s=0; s < stlpcov; s++)
                pole[r][s] = citacSuboru.nextInt();
    }
    return pole;
}
```





# Čítame maticu zo súboru

```
public int[][] nacistajMaticu(File subor) {
    Scanner citacSuboru = null;
    try {
        citacSuboru = new Scanner(subor);
        int riadkov = citacSuboru.nextInt();
        int stlpcov = citacSuboru.nextInt();
        int[][] pole = new int[riadkov][stlpcov];
        for (int r=0; r < riadkov; r++)
            for (int s=0; s < stlpcov; s++)
                pole[r][s] = citacSuboru.nextInt();
    }
    return pole;
}
```





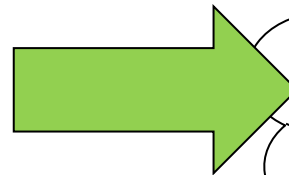
# Zasa čítame maticu zo súboru

- Vytvorme si metódu, ktorá načíta zo súboru maticu čísiel, ale:
  - V prvom riadku **nie je** uvedený počet riadkov a počet stĺpcov, ale rovno prvý riadok matice

```
-1 456 2 0
```

```
1 2 1 1
```

```
3000 5 6 -7
```



-1	456	2	0
1	2	1	1
3000	5	6	-7



# Zasa čítame maticu zo súboru

- Vytvorme si metódu, ktorá načíta zo súboru maticu čísiel, ale:
  - V prvom riadku **nie je** uvedený počet riadkov a počet stĺpcov, ale rovno prvý riadok matice
  - Aké veľké dvojrozmerné pole máme vytvoriť?





# Zasa čítame maticu zo súboru

- Vytvorme si metódu, ktorá načíta zo súboru maticu čísiel, ale:
  - V prvom riadku **nie je** uvedený počet riadkov a počet stĺpcov, ale rovno prvý riadok matice
  - Aké veľké dvojrozmerné pole máme vytvoriť?
    - Prvé riešenie : nejaké „určite väčšie“ ako dáta (napr. 20x20)
      - Pamätám si použitú časť dvojrozmerného poľa

riadkov stĺpcov

3

4

-1	456	2	0	0	0	...
1	2	1	1	0	0	...
3000	5	6	-7	0	0	...
0	0	0	0	0	0	...
...						...



# Zasa čítame maticu zo súboru

- Vytvorme si metódu, ktorá načíta zo súboru maticu čísiel, ale:
  - V prvom riadku **nie je** uvedený počet riadkov a počet stĺpcov, ale rovno prvý riadok matice
  - Aké veľké dvojrozmerné pole máme vytvoriť?
    - Prvé riešenie : nejaké „určite väčšie“ ako dáta (napr. 20x20)
      - Pamätám si použitú časť dvojrozmerného poľa
      - Použijem zbytočne väčšiu RAMku, alebo podcením veľkosť
      - Časté riešenie: skúsite si ho doma alebo na cvičení 😊
      - Po ňom môže nasledovať kopírovanie do presného poľa



# Zasa čítame maticu zo súboru

- Vytvorme si metódu, ktorá načíta zo súboru maticu čísiel, ale:
  - V prvom riadku **nie je** uvedený počet riadkov a počet stĺpcov, ale rovno prvý riadok matice
  - Aké veľké dvojrozmerné pole máme vytvoriť?
    - Prvé riešenie : nejaké „určite väčšie“ ako dáta (napr. 20x20)
      - Pamätám si použitú časť dvojrozmerného poľa
      - Použijem zbytočne väčšiu RAMku, alebo podcením veľkosť
      - Časté riešenie: skúsite si ho doma alebo na cvičení
      - Po ňom môže nasledovať kopírovanie do presného poľa
    - Druhé riešenie : Najprv si zistím počet riadkov a stĺpcov, potom vytvorím pole presnej veľkosti a nakoniec ho naplním



# Zisťujeme veľkosť matice v súbore

```
public int[][] nacistajLubovolnuMaticu(File subor) {
    Scanner citacSuboru = null;
    try {
        citacSuboru = new Scanner(subor);
        int stlpcov = 0;
        int riadkov = 0;
        while (citacSuboru.hasNextLine()) {
            String riadok = citacSuboru.nextLine();
            riadkov++;
            Scanner citacRiadka = new Scanner(riadok);
            stlpcov = 0;

```

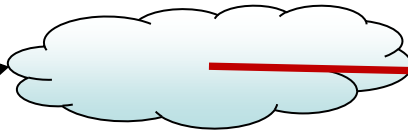


náš program

citacSuboru



objekty



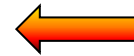
reálny svet

-1	456	2	0
1	2	1	1
3000	5	6	-7



# Zisťujeme veľkosť matice v súbore

```
public int[][] nacistajLubovolnuMaticu(File subor) {
    Scanner citacSuboru = null;
    try {
        citacSuboru = new Scanner(subor);
        int stlpcov = 0;
        int riadkov = 0;
        while (citacSuboru.hasNextLine()) {
            String riadok = citacSuboru.nextLine();
            riadkov++;
            Scanner citacRiadka = new Scanner(riadok);
            stlpcov = 0;
```



náš program

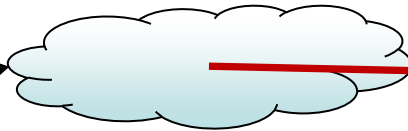
citacSuboru

riadkov stlpcov

0

0

objekty



reálny svet

-1	4	5	6	2	0
1	2	1	1		
3000	5	6	-7		

true



# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine(); ←
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```

náš program

citacSuboru

riadkov stlpcov

0 0

riadok

objekty

reálny svet

-1 456 2 0

1 2 1 1

3000 5 6 -7



# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```



náš program

citacSuboru



riadkov stlpcov

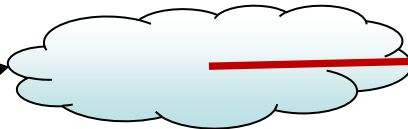
1

0

riadok



objekty



reálny svet

-1	456	2	0
1	2	1	1
3000	5	6	-7



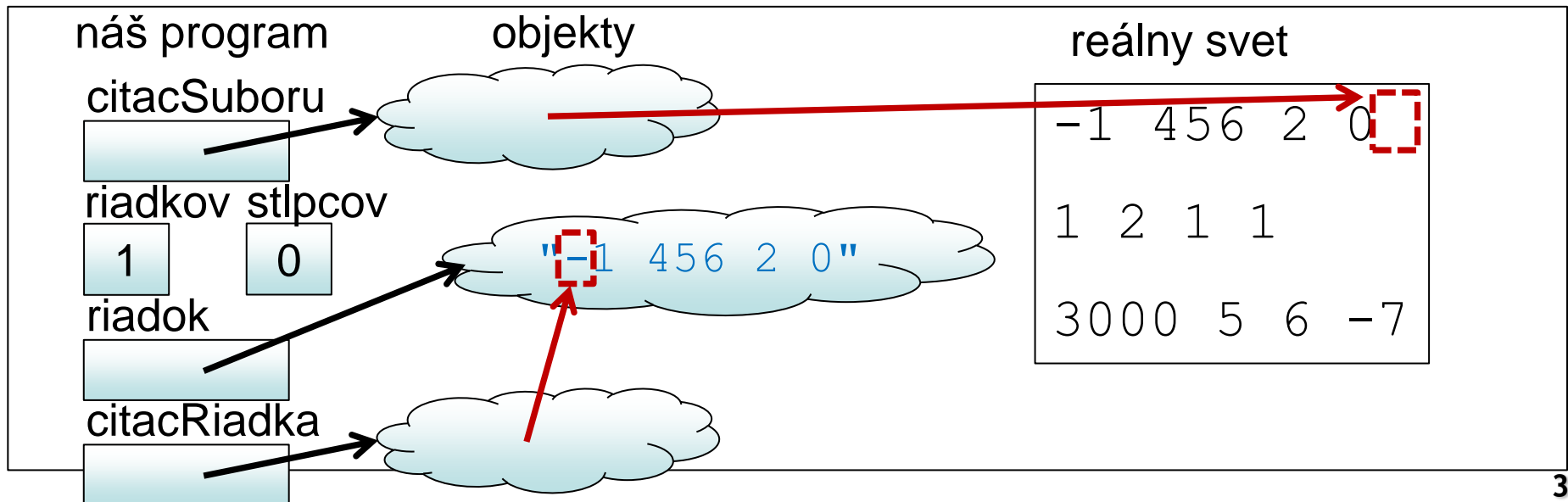


# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok); ←
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```







# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```



náš program

objekty

reálny svet

citacSuboru

riadkov stlpcov

riadok

citacRiadka

1

0

"-1 456 2 0"

-1	456	2	0
1	2	1	1
3000	5	6	-7

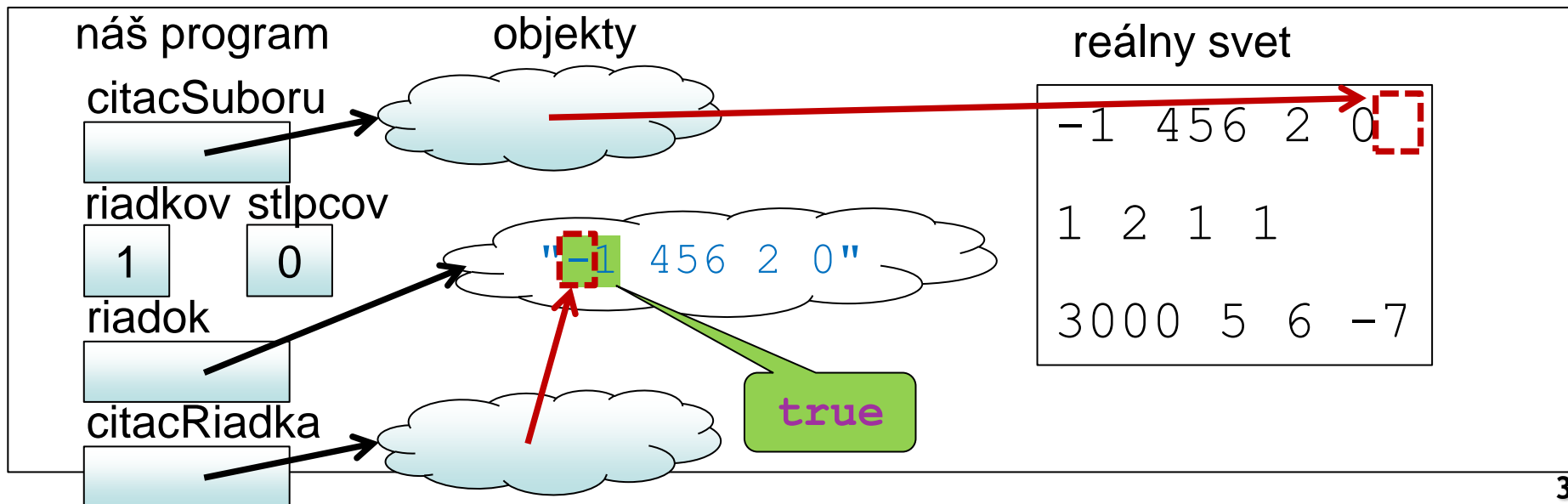


# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```

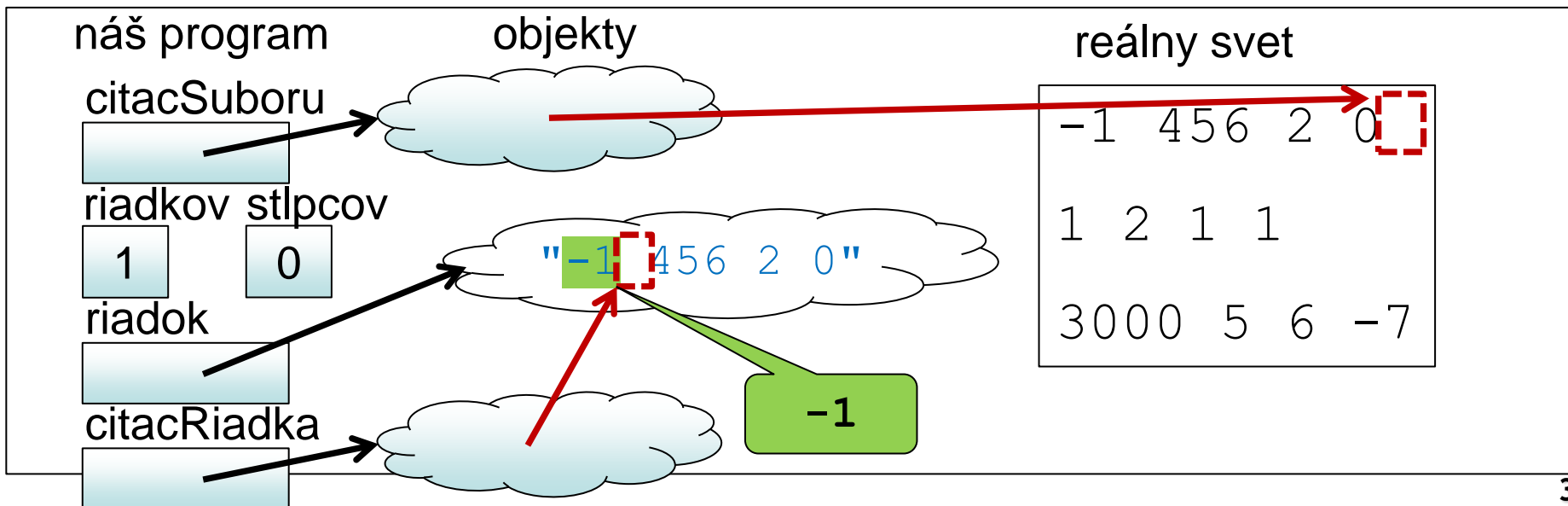




# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}
    
```

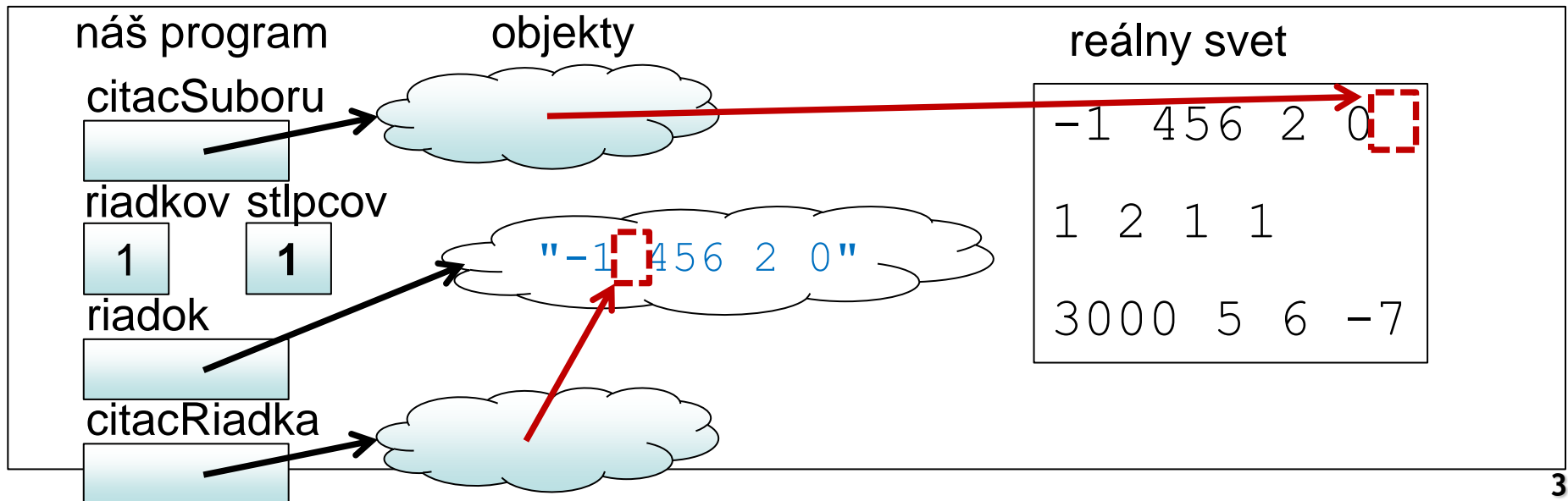




# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}
    
```



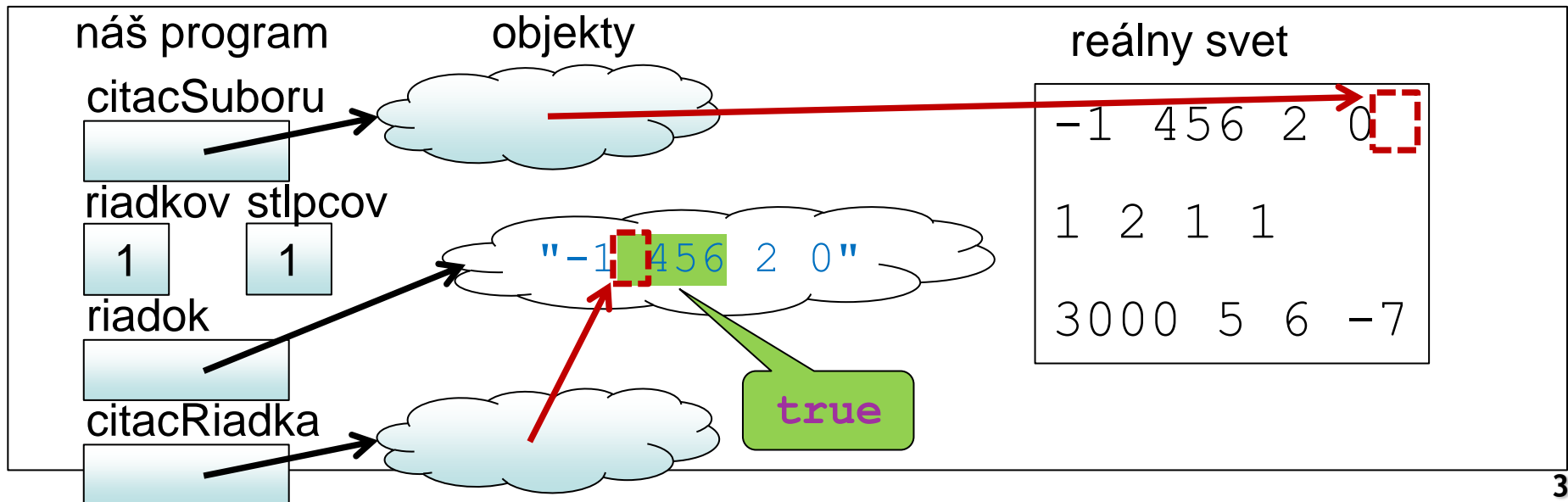


# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```

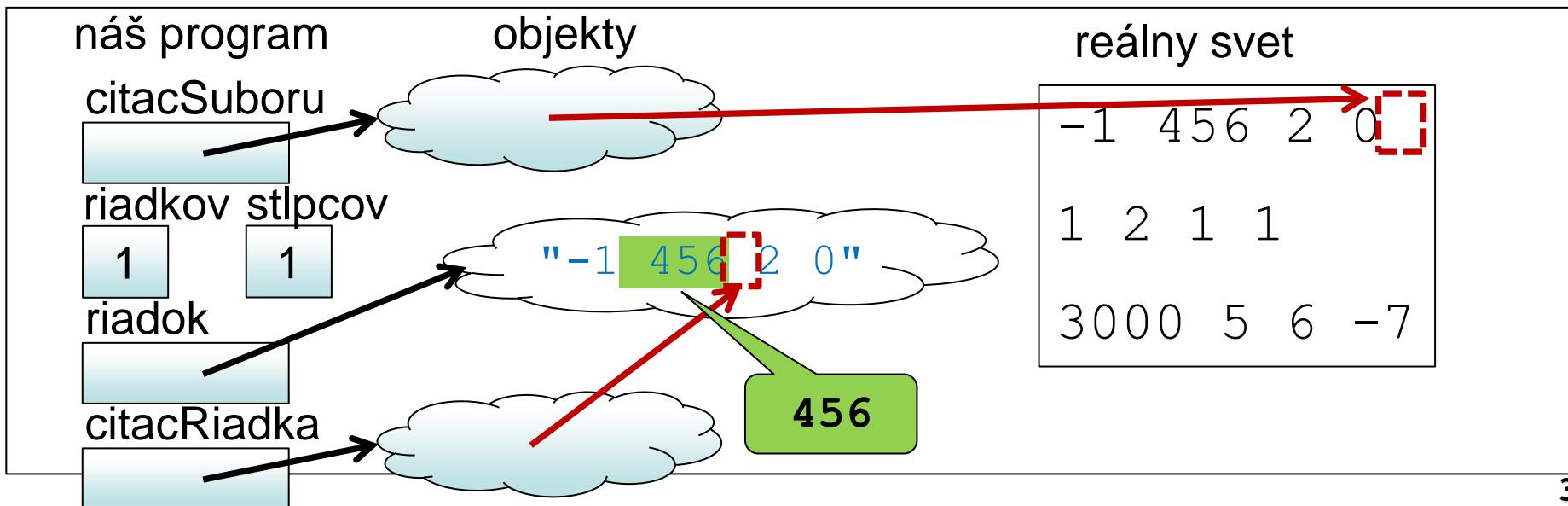




# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}
    
```



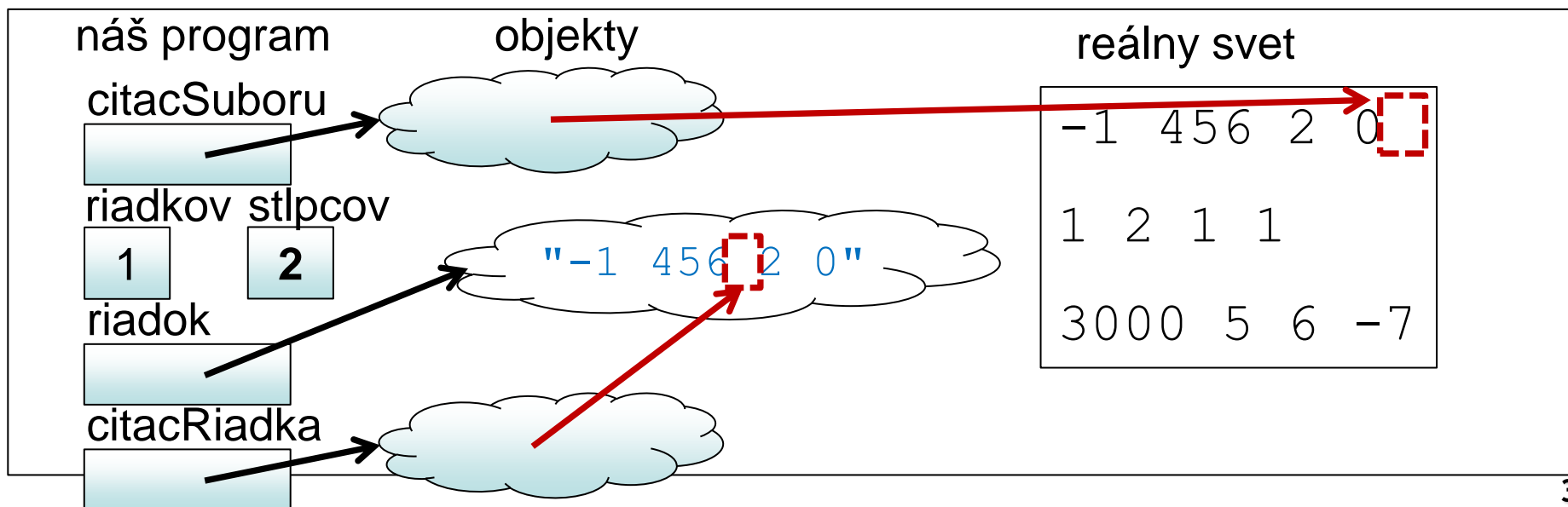


# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```



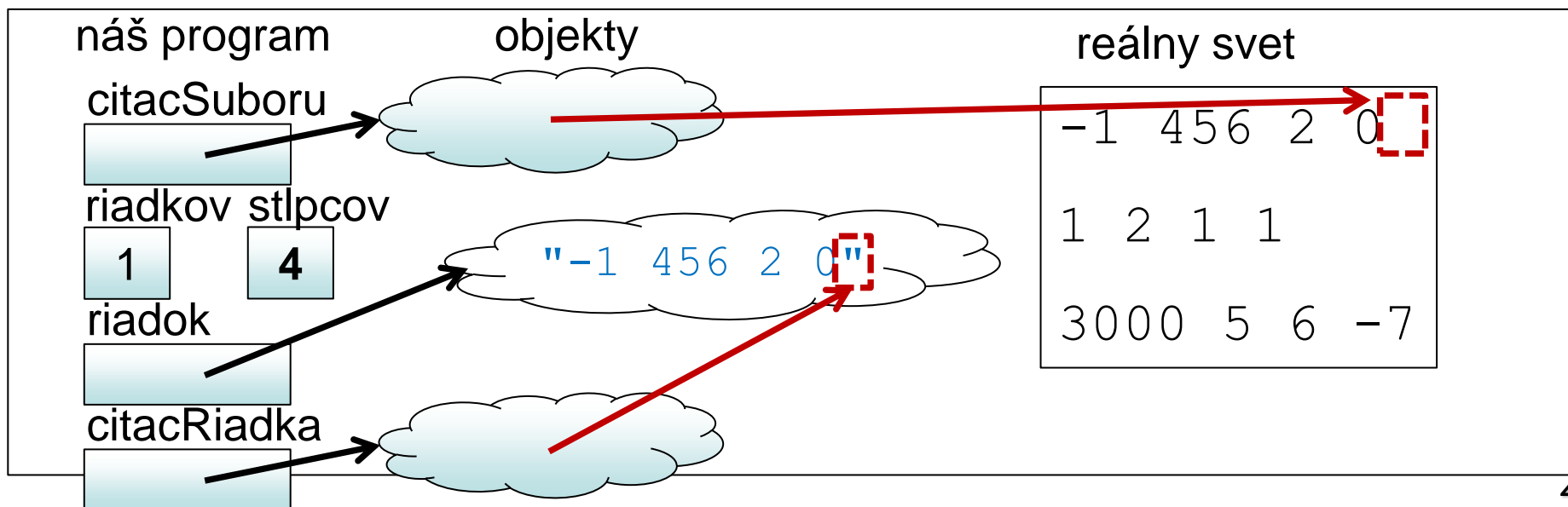


# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```





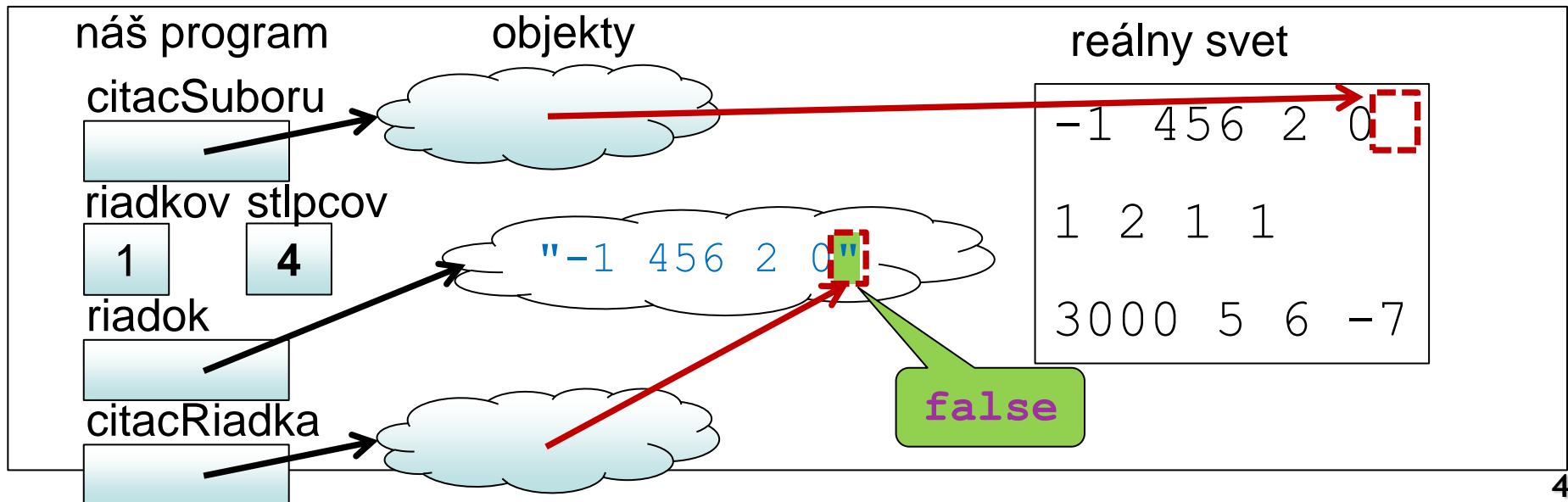


# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```





# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```



náš program

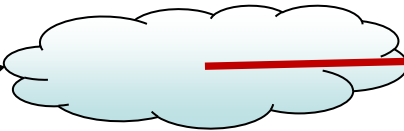
citacSuboru

riadkov stlpcov

1

4

objekty



reálny svet

-1	4	5	6	2	0
1	2	1	1		
3	0	0	5	6	-7

true



# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine(); ←
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```

náš program

citacSuboru

riadkov stlpcov

1

4

riadok

objekty

reálny svet

-1 456 2 0

1 2 1 1

3000 5 6 -7



# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```



náš program

citacSuboru



riadkov stlpcov

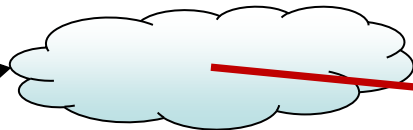
2

4

riadok



objekty



reálny svet

-1 456 2 0

1 2 1 1

3000 5 6 -7



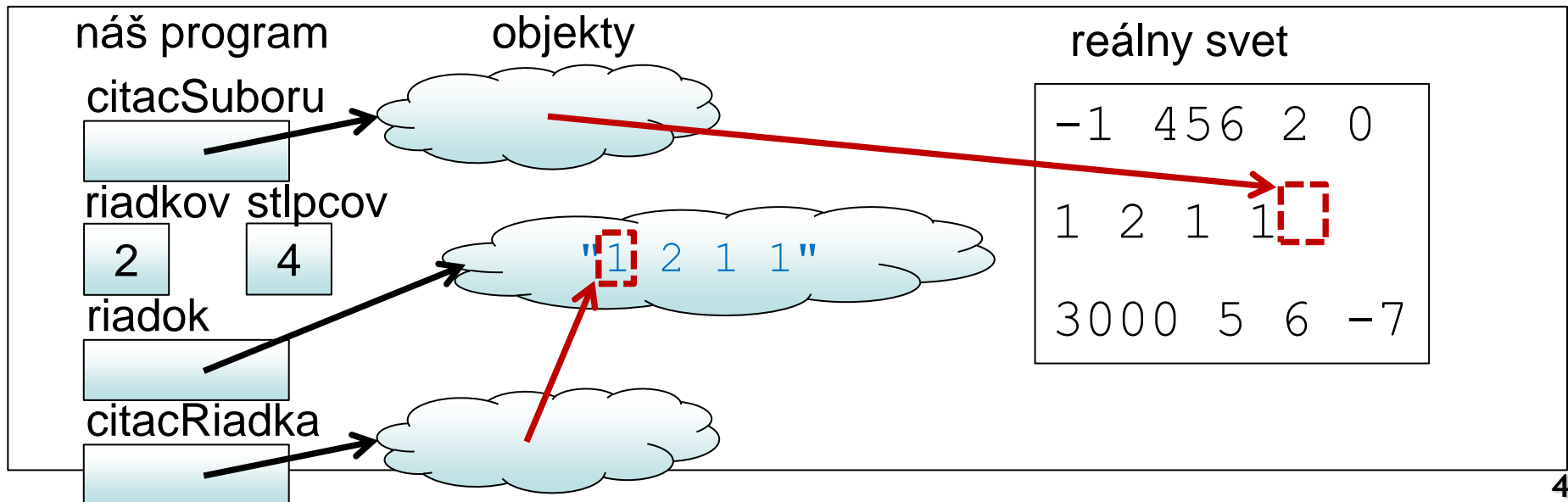


# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok); ←
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```





# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```



náš program

citacSuboru



riadkov stlpcov

2

0

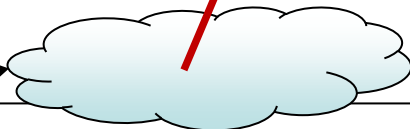
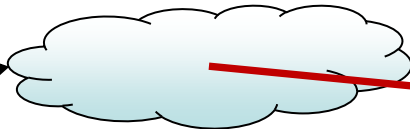
riadok



citacRiadka



objekty

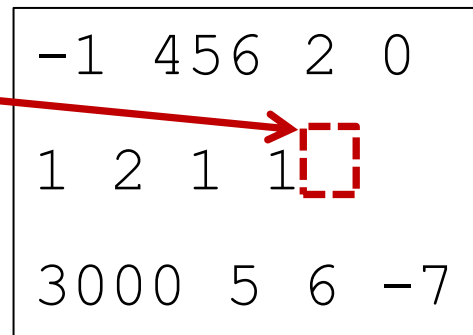


reálny svet

-1 456 2 0

1 2 1 1

3000 5 6 -7



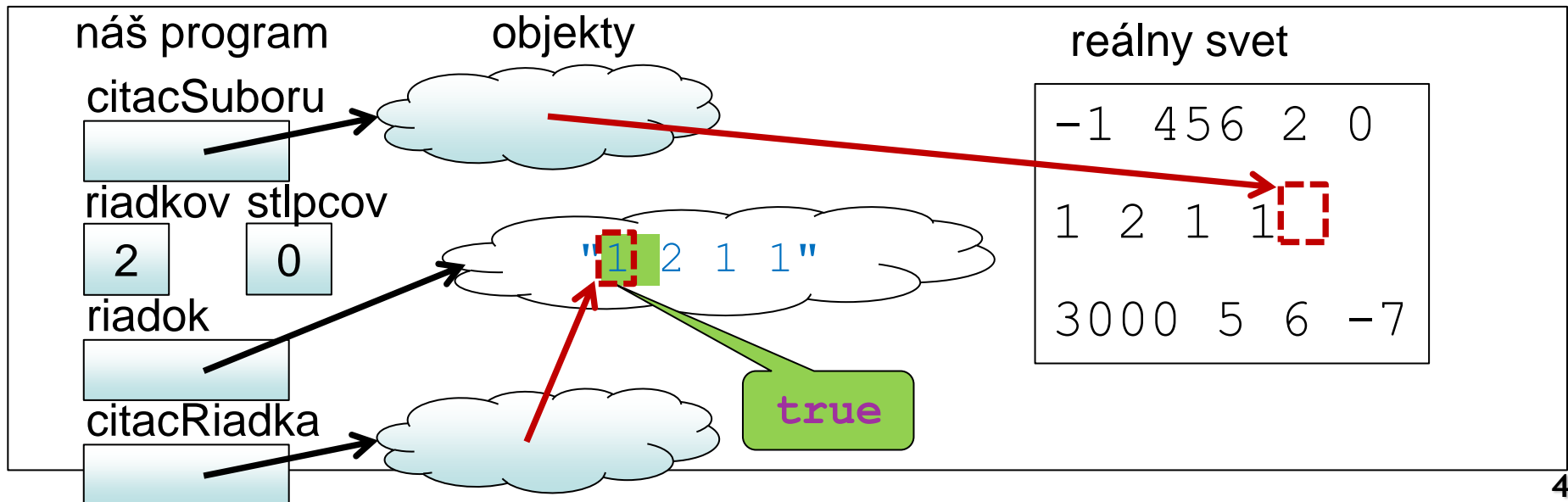


# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```



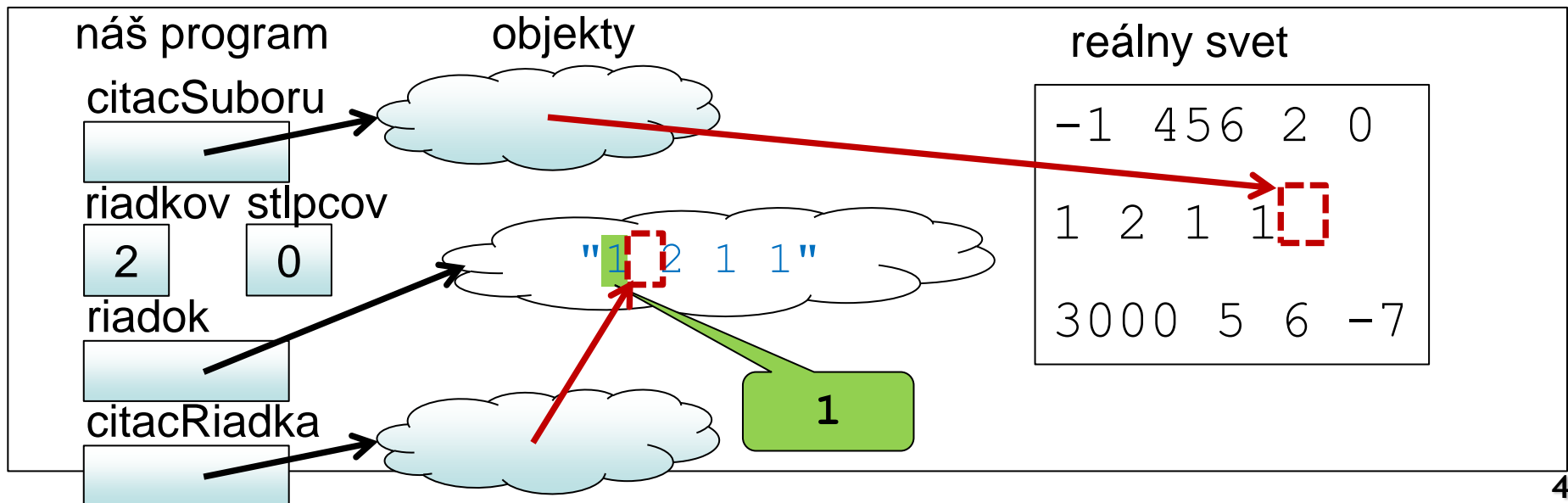


# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```





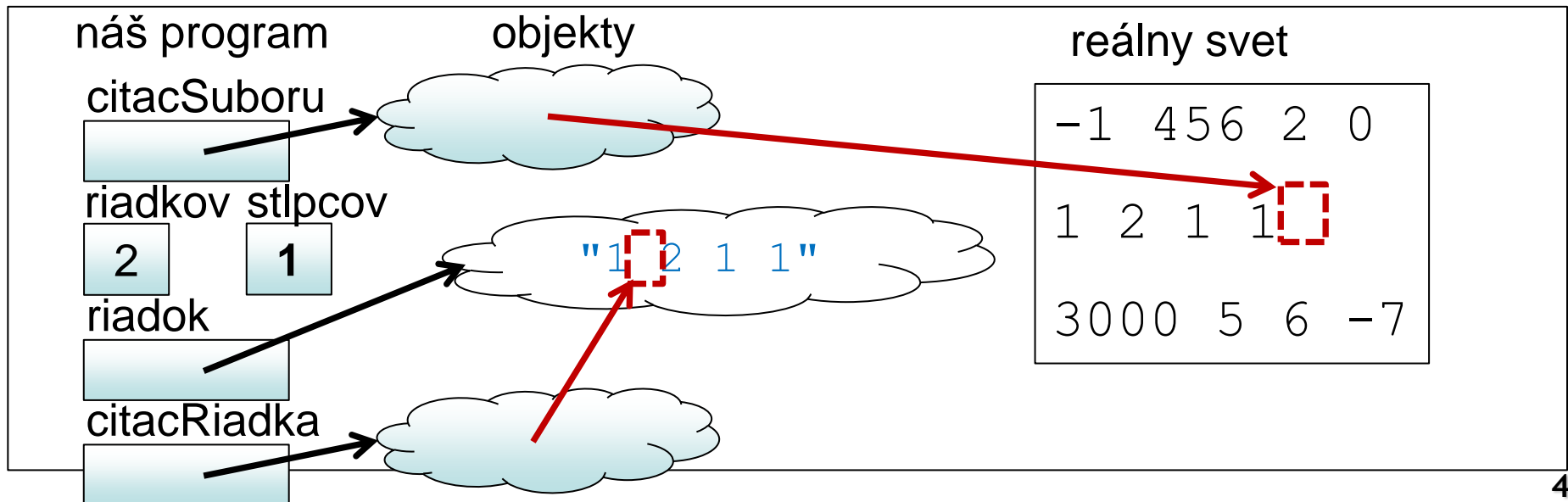


# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```



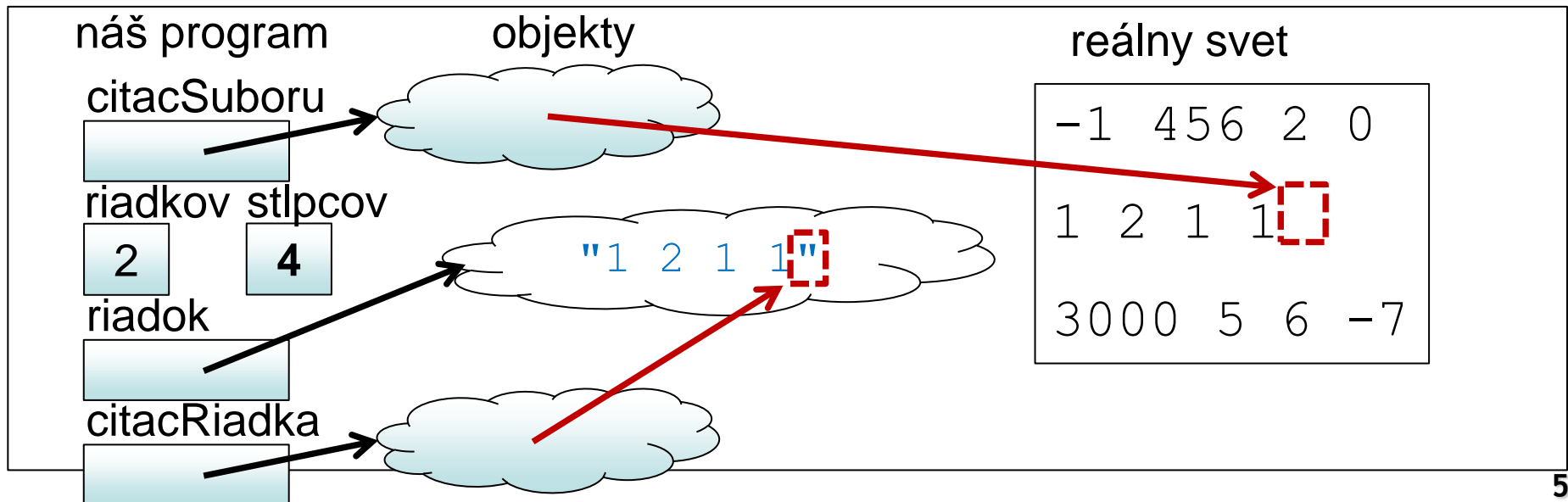


# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```



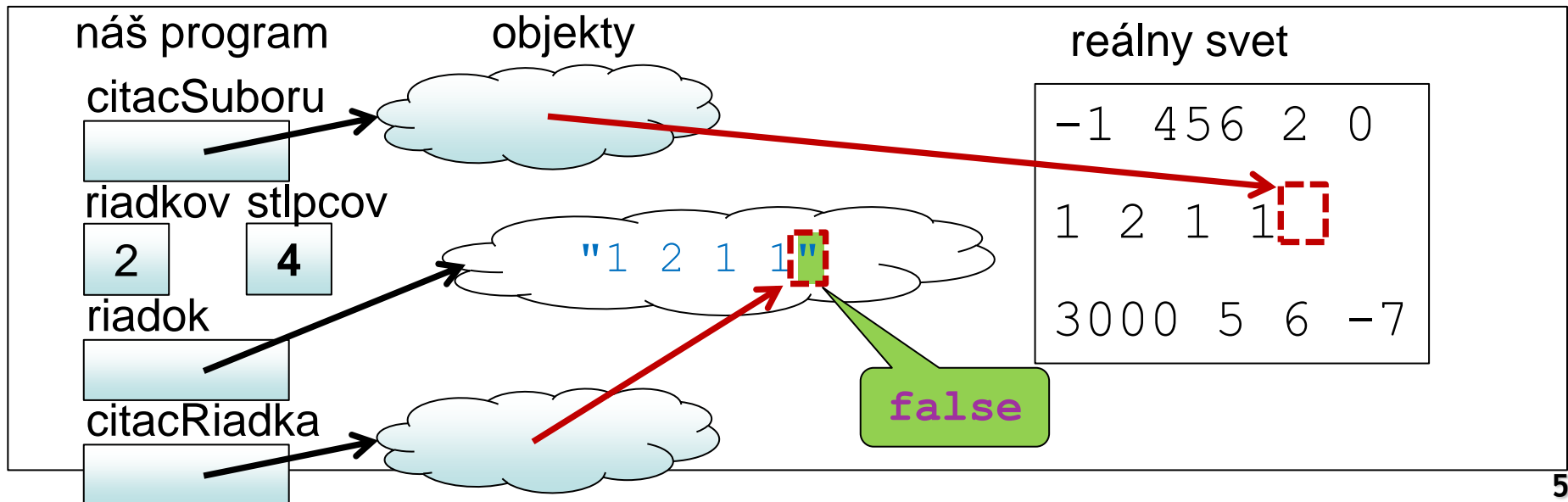


# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```





# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```



náš program

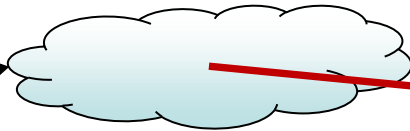
citacSuboru

riadkov stlpcov

2

4

objekty



reálny svet

-1 456 2 0

1 2 1 1

3000 5 6 -7

true



# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```



náš program

citacSuboru

riadkov stlpcov

2

4

riadok

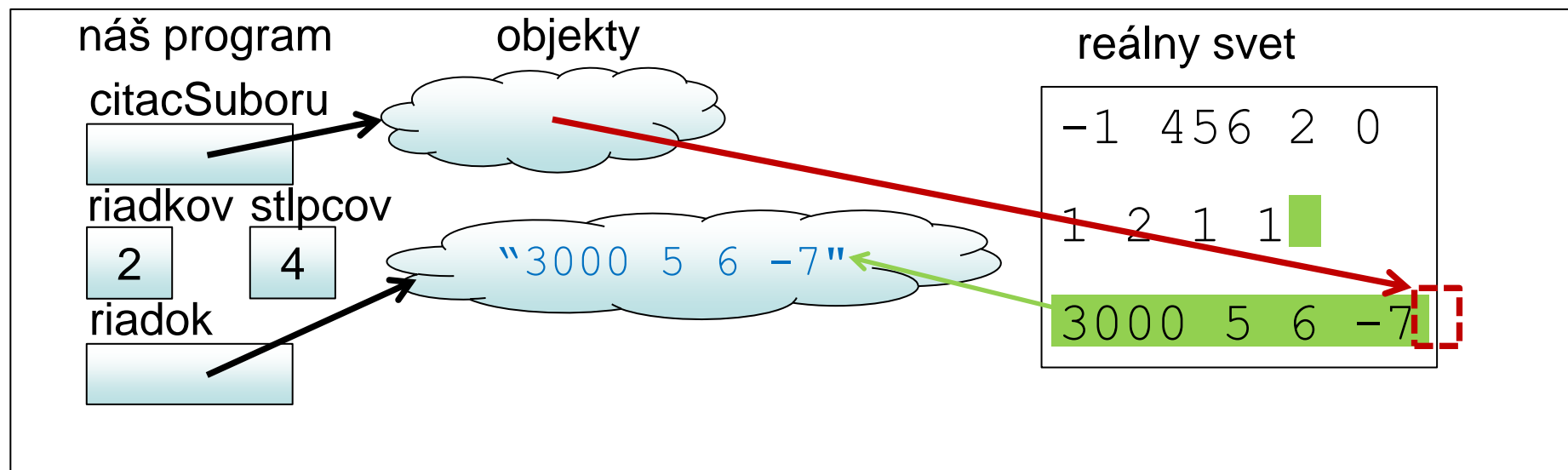
objekty

reálny svet

-1 456 2 0

1 2 1 1

3000 5 6 -7





# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```



náš program

citacSuboru



riadkov stlpcov

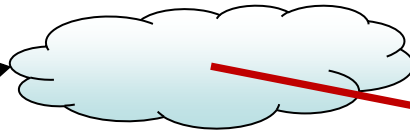
3

4

riadok



objekty



reálny svet

-1	456	2	0
1	2	1	1
3000	5	6	-7



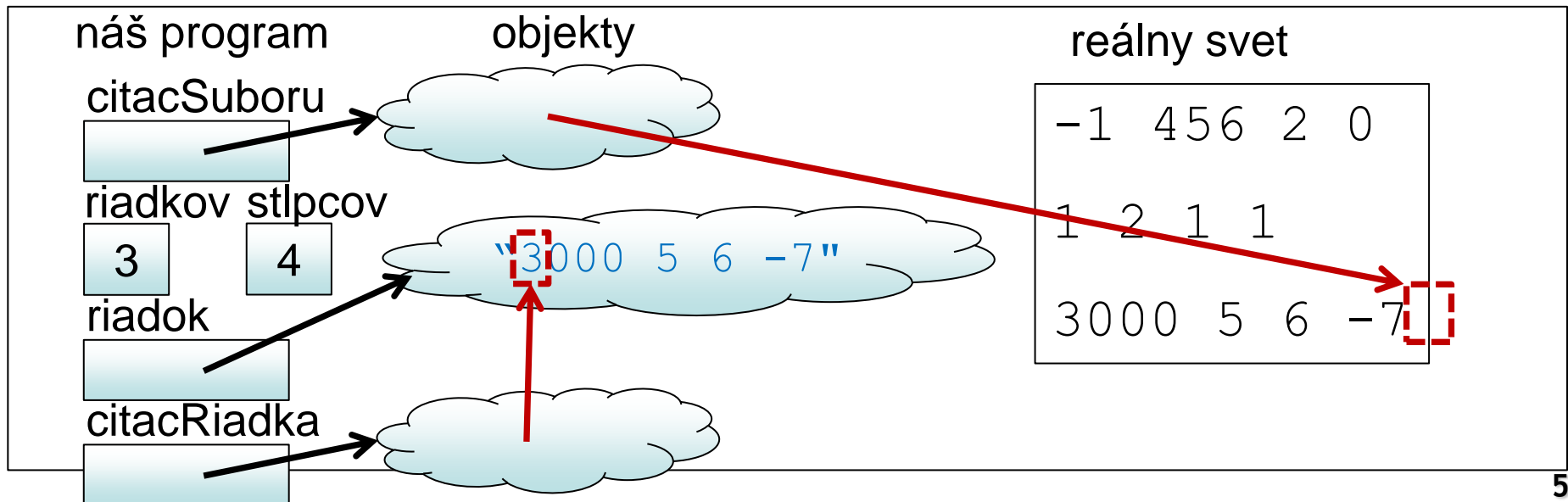


# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok); ←
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```





# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```



náš program

citacSuboru

riadkov stlpcov

3

0

riadok

citacRiadka

objekty

"3000 5 6 -7"

reálny svet

-1 456 2 0

1 2 1 1

3000 5 6 -7





# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```



náš program

citacSuboru



riadkov stlpcov

3

0

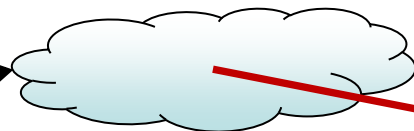
riadok



citacRiadka



objekty



true

reálny svet

-1 456 2 0

1 2 1 1

3000 5 6 -7

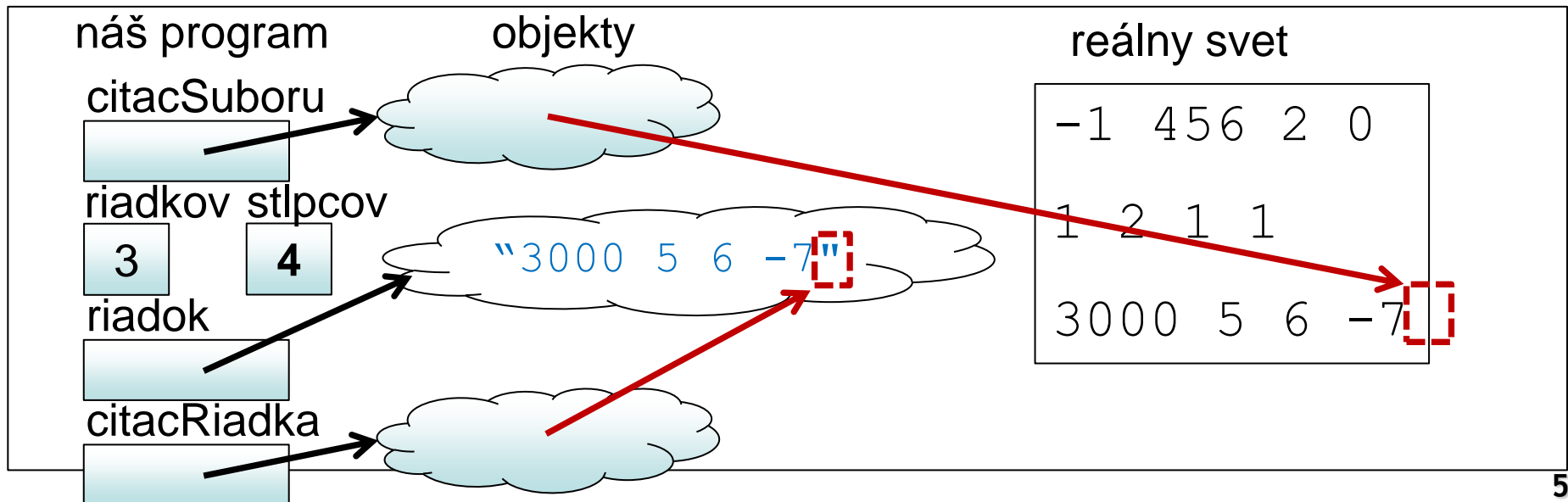


# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```



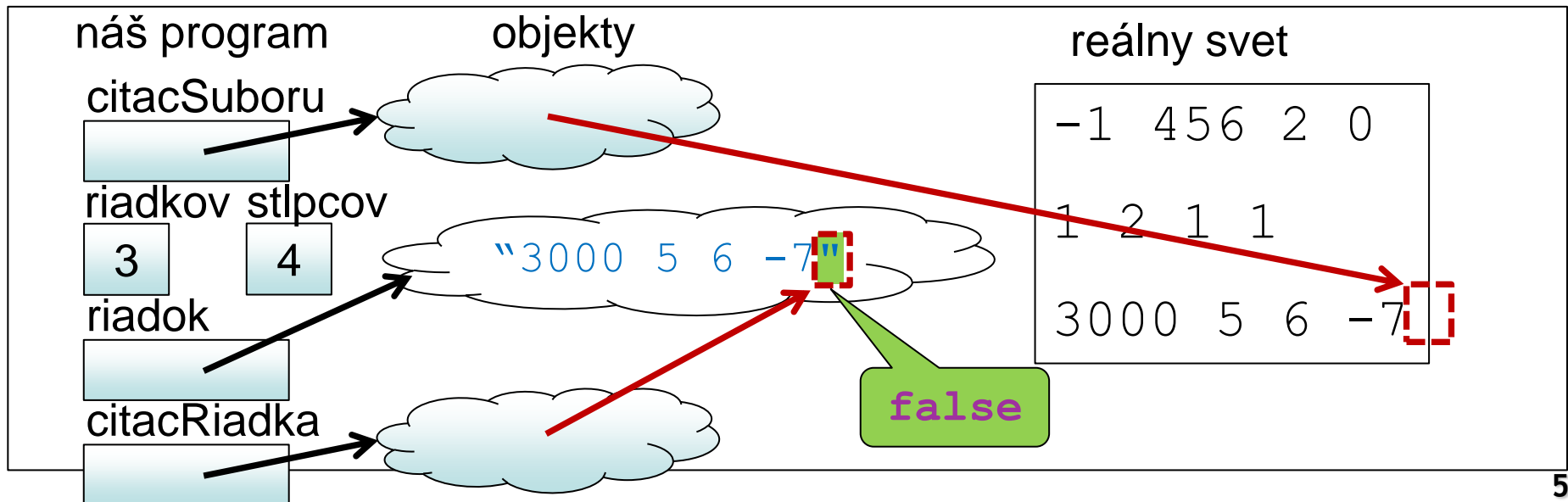


# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```





# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}

```



Končí vonkajší  
while cyklus

riadkov == 3

stlpcov == 4

náš program

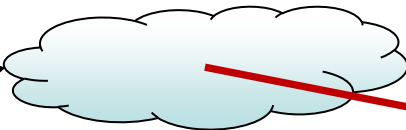
citacSuboru

riadkov stlpcov

3

4

objekty



reálny svet

-1 456 2 0

1 2 1 1

3000 5 6 -7

false



# Zisťujeme veľkosť matice v súbore

```

...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}
citacSuboru.close();
citacSuboru = new Scanner(subor);
int[][] pole = new int[riadkov][stlpcov];
for (int r=0; r < riadkov; r++) {
    for (int s=0; s < stlpcov; s++) {
        pole[r][s] = citacSuboru.nextInt();
    }
}
return pole;
...

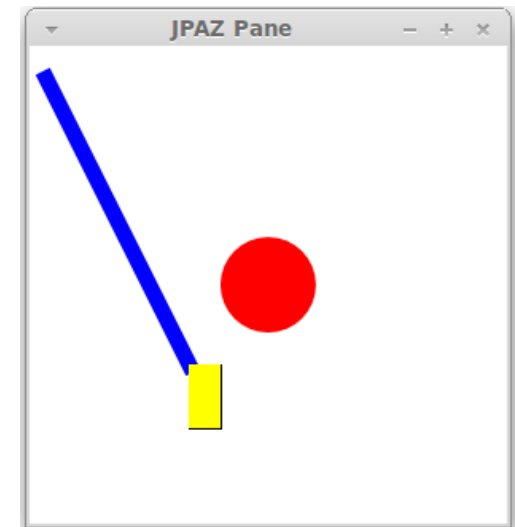
```

Stačí iba použiť  
známy prístup z  
predchádzajúcej  
metódy



# Čítame množinu tvarov

- Vytvoríme si metódu, ktorá načíta zo súboru množinu tvarov, ktoré treba vykresliť
- V súbore
  - ciara  $x_1$   $y_1$   $x_2$   $y_2$  farba hrúbka
    - Napr. ciara 10 20 100 200 blue 10
  - kruh  $x$   $y$  polomer farba
    - Napr. kruh 150 150 30 red
  - obdlzdnik  $x_1$   $y_1$   $x_2$   $y_2$  farba
    - Napr. obdlzdnik 100 200 120 240 yellow
- V programe:
  - `plocha.ciara(10, 20, 100, 200, Color.blue, 10);`
  - `plocha.kruh(150, 150, 30, Color.red);`
  - `plocha.obdlzdnik(100, 200, 120, 240, Color.yellow);`





# Ďakujem za pozornosť !

