

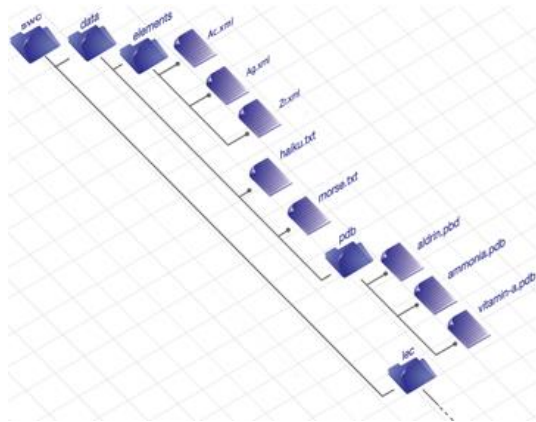


8. prednáška (4.11.2013)

Midterm a súbory prakticky

alebo

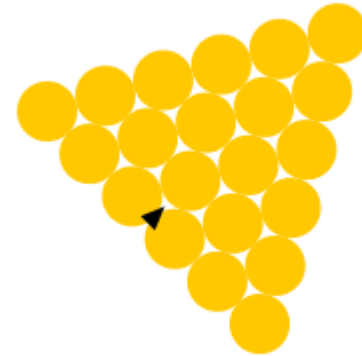
Vstupy zo súborov
sú super!





Midterm - Pyramída

- Do triedy `Midtermarka` pridajte metódu `pyramida`, ktorá nakreslí pyramídu z kruhov (vid' ilustračný obrázok). Metóda má dva parametre `pocetRadov` a `polomer`. Prvý parameter `pocetRadov` určuje počet radov, z ktorých sa pyramída skladá (je to zároveň počet kruhov v najširšom rade). Druhý parameter `polomer` určuje polomer kruhov, z ktorých sa pyramída skladá. Farba kruhov je aktuálne nastavená farba výplne. Je dôležité, aby sa všetky „susediace“ kruhy v pyramíde presne dotýkali. Korytnačka sa na začiatku nachádza v strede najširšieho radu a je natočená v smere výšky pyramídy. Po nakreslení pyramídy nech je korytnačka na pozícii a je natočená v smere, ako bola pred volaním metódy.



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



Midterm – Inverzný faktoriál

- Do triedy Midtermarka pridajte metódu inverznyFaktorial, ktorá pre zadané nezáporné celé číslo n vráti najmenšie také celé číslo k , že $n \leq k!$

```
public boolean inverznyFaktorial(long n)
```



Midterm – Pomoc pre hendikepovaných

- Naučte korytnačky triedy `ZadanaKorytnacka` metódu `odstranDuplicitu`, ktorá vráti referenciu na novovytvorený objekt triedy `String`, ktorého obsah vznikne odstránením duplicitných výskytov rovnakého znaku v rade za sebou.

```
public String odstranDuplicitu(String r)
```

- Príklady:
 - `odstranDuplicitu("Jaaavvaaa") = "Java"`
 - `odstranDuplicitu("IIIssiel") = "Isiel"`
 - `odstranDuplicitu("IIissiel") = "Isiel"`
 - `odstranDuplicitu("Odddstran duuuupllicitu") = "Odstran duplicitu"`



Midterm - Bombastické korytnačky

- Vytvorte triedu `MidtermPane`, ktorá rozširuje triedu `WinPane`. Po vytvorení kresliacej plochy triedy `MidtermPane` nech sa v nej vytvorí 13 korytnačiek triedy `Turtle` na náhodných pozíciách vo viditeľnej časti kresliacej plochy.
- Do kresliacej plochy umiestnili nálož so zadanou silou (parameter `silou`) na pozíciu `[x, y]`. Pri explózii nálož vznikne tlaková vlna, ktorá odhodí korytnačky v smere od miesta explózie. Ak uvážime zjednodušený fyzikálny model, korytnačka vzdialená d od miesta explózie so silou S je odhodená na vzdialenosť S^2/d^4 . Budeme predpokladať, že žiadna korytnačka sa nenachádza v mieste explózie. Do triedy `MidtermPane` pridajte metódu `explozia`, ktorá odsimuluje účinky explózie nálož so zadanou silou a umiestnenej na zadanej pozícii. T.j. všetky korytnačky v ploche sa presunú (bez kreslenia trajektórie pohybu) na miesto, kam ich odhodí tlaková vlna, a budú nasmerované v smere od miesta explózie. Metóda vráti najväčšiu vzdialenosť, na akú bola odhodená nejaká z korytnačiek v ploche.

```
public double explozia(double x, double y, double sila)
```



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

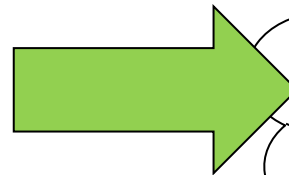
- Vytvorme si metódu, ktorá načíta zo súboru maticu čísiel.
 - V prvom riadku je uvedený počet riadkov a počet stĺpcov

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

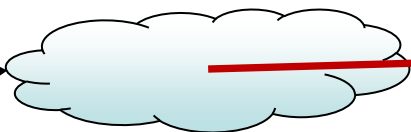
objekty

reálny svet

subor



citacSuboru



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-
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
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

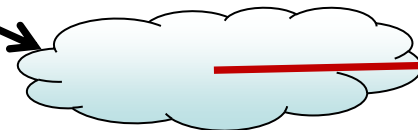
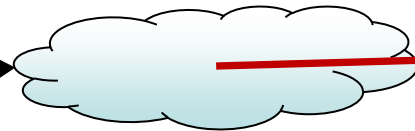
objekty

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;
    }
}
```

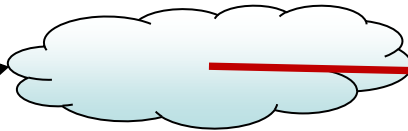


náš program

citacSuboru



objekty



reálny svet

3	4		
-1	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

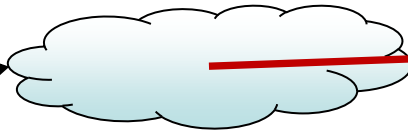
citacSuboru



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

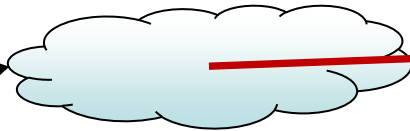
citacSuboru



riadkov stlpcov



objekty



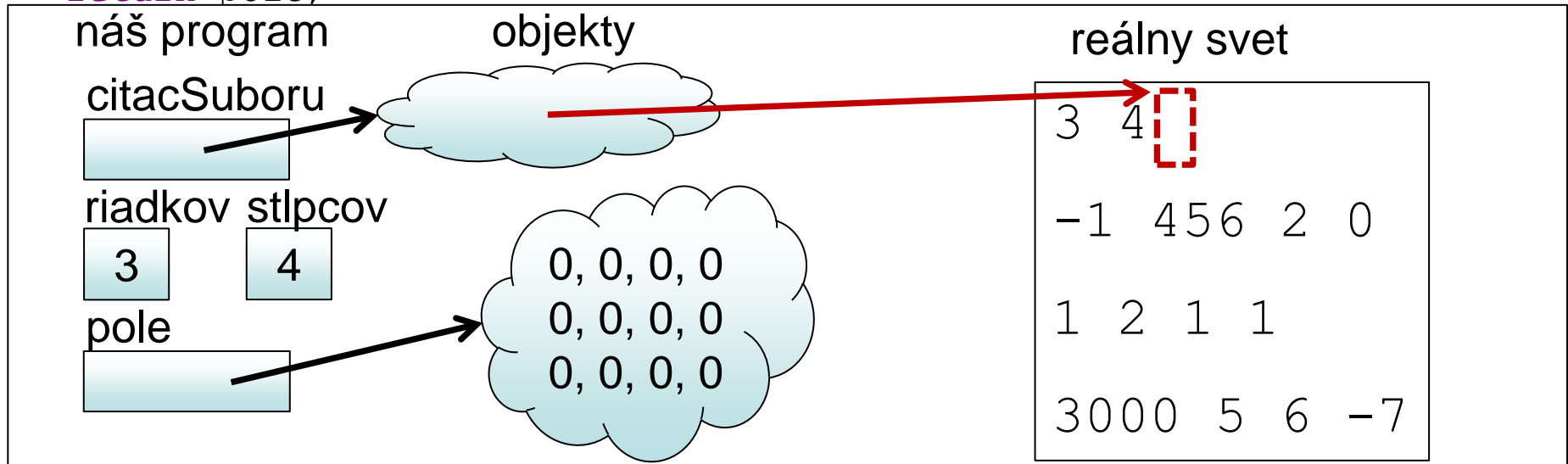
reálny svet

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



Čítame maticu zo súboru

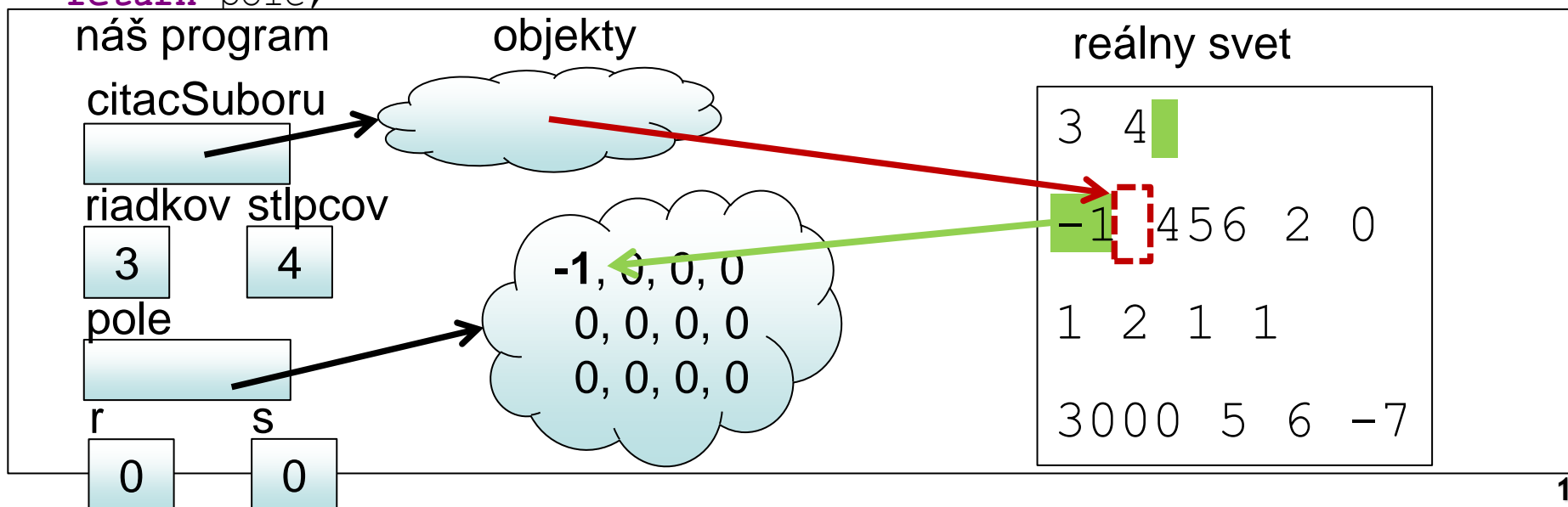
```
public int[][] nacistajMaticu(File subor) {
    Scanner citacSuboru = null;
    try {
        citacSuboru = new Scanner(subor);
        int riadkov = citacSuboru.nextInt();
        int stlpcov = citacSuboru.nextInt();
        int[][] pole = new int[riadkov][stlpcov]; ←
        for (int r=0; r < riadkov;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





Čítame maticu zo súboru

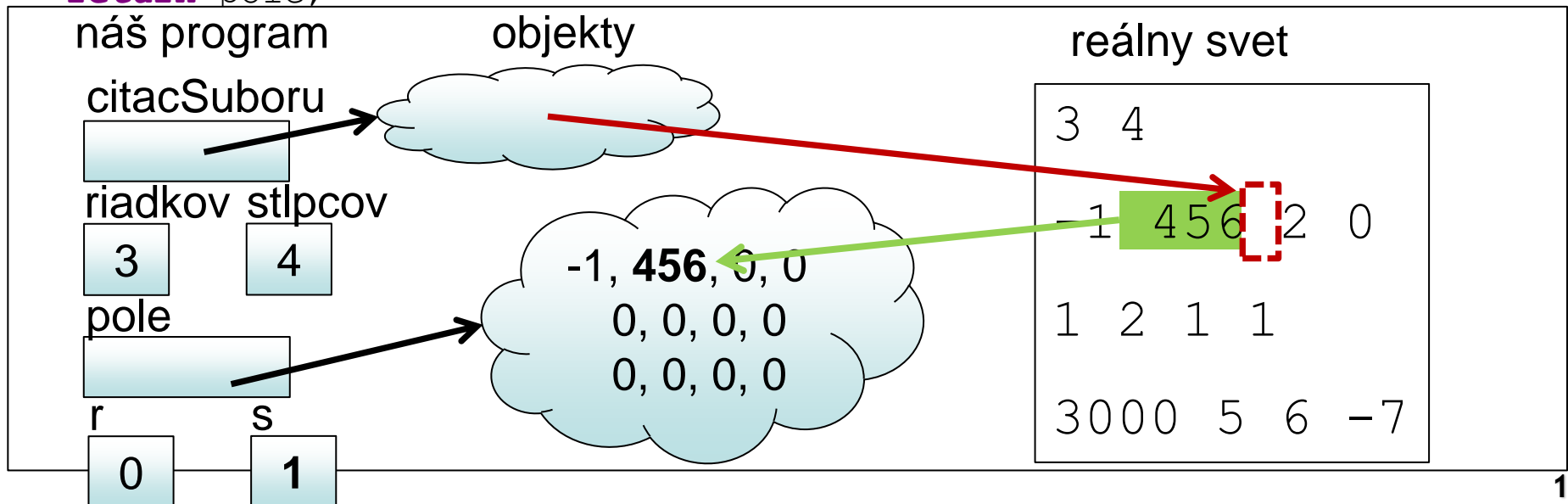
```
public int[][] nacistajMaticu(File subor) {
    Scanner citacSuboru = null;
    try {
        citacSuboru = new Scanner(subor);
        int riadkov = citacSuboru.nextInt();
        int stlpcov = citacSuboru.nextInt();
        int[][] pole = new int[riadkov][stlpcov];
        for (int r=0; r < riadkov;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





Čítame maticu zo súboru

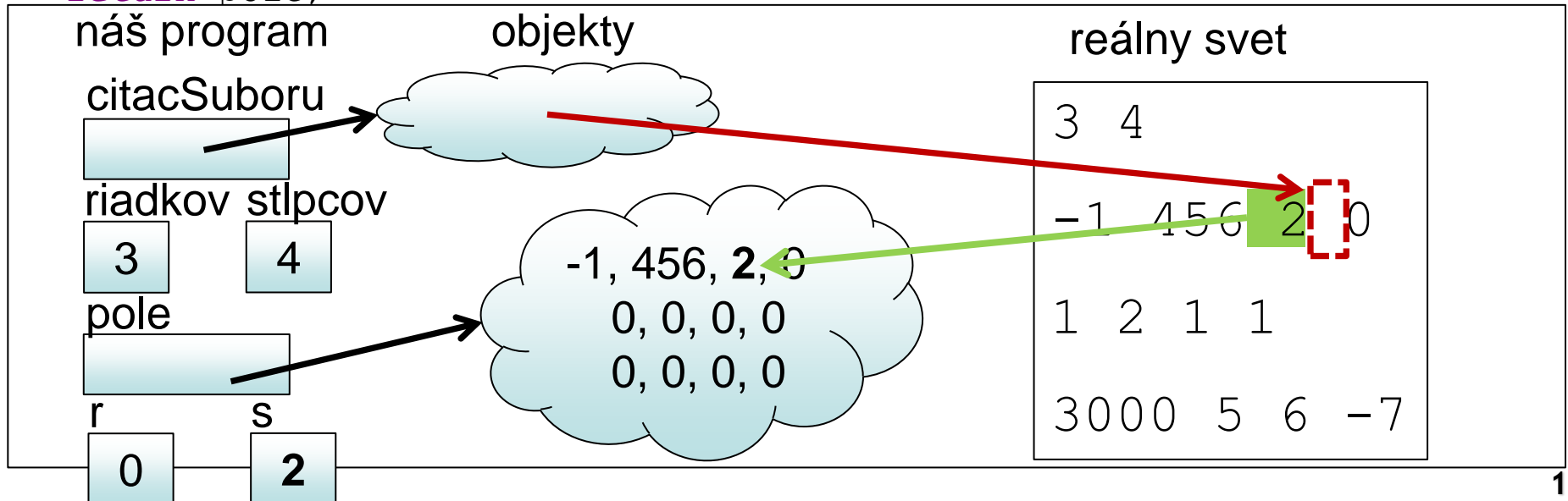
```
public int[][] nacistajMaticu(File subor) {
    Scanner citacSuboru = null;
    try {
        citacSuboru = new Scanner(subor);
        int riadkov = citacSuboru.nextInt();
        int stlpcov = citacSuboru.nextInt();
        int[][] pole = new int[riadkov][stlpcov];
        for (int r=0; r < riadkov;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





Čítame maticu zo súboru

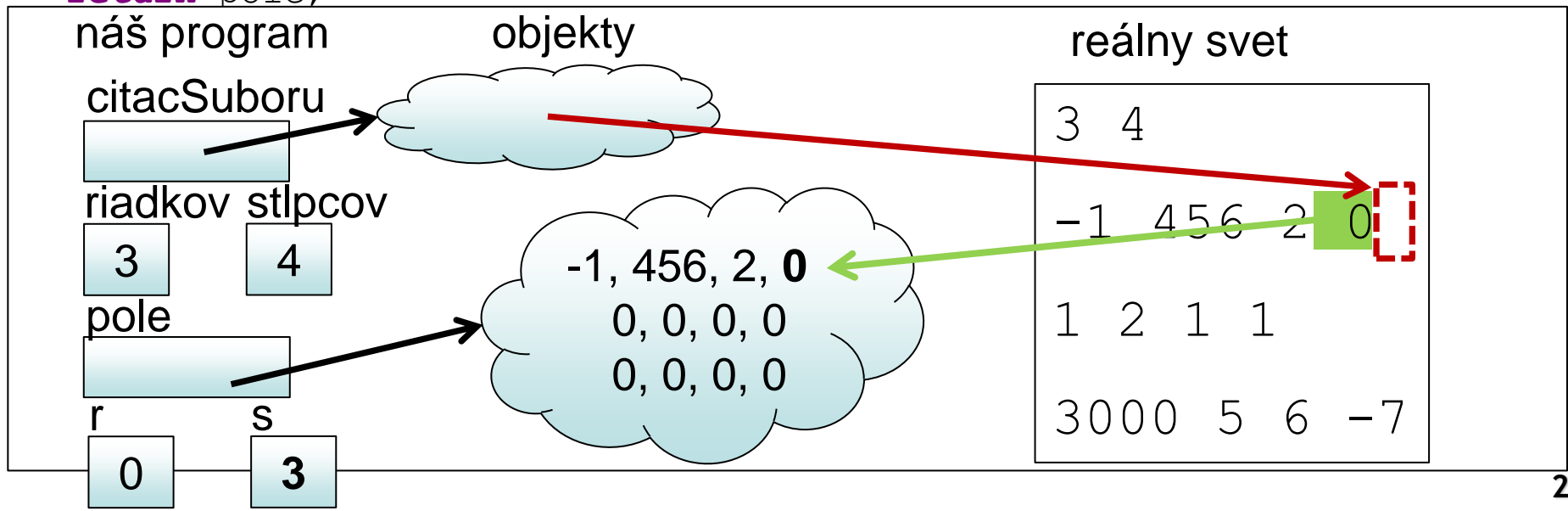
```
public int[][] nacistajMaticu(File subor) {
    Scanner citacSuboru = null;
    try {
        citacSuboru = new Scanner(subor);
        int riadkov = citacSuboru.nextInt();
        int stlpcov = citacSuboru.nextInt();
        int[][] pole = new int[riadkov][stlpcov];
        for (int r=0; r < riadkov;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





Čítame maticu zo súboru

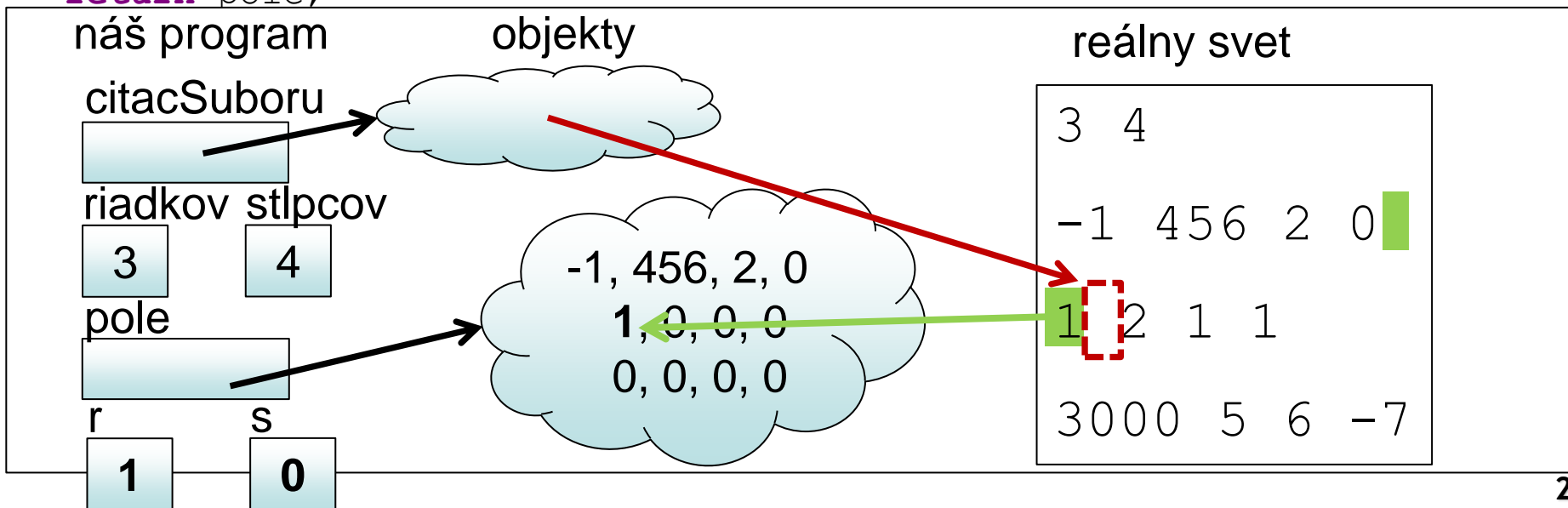
```
public int[][] nacistajMaticu(File subor) {
    Scanner citacSuboru = null;
    try {
        citacSuboru = new Scanner(subor);
        int riadkov = citacSuboru.nextInt();
        int stlpcov = citacSuboru.nextInt();
        int[][] pole = new int[riadkov][stlpcov];
        for (int r=0; r < riadkov;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





Čítame maticu zo súboru

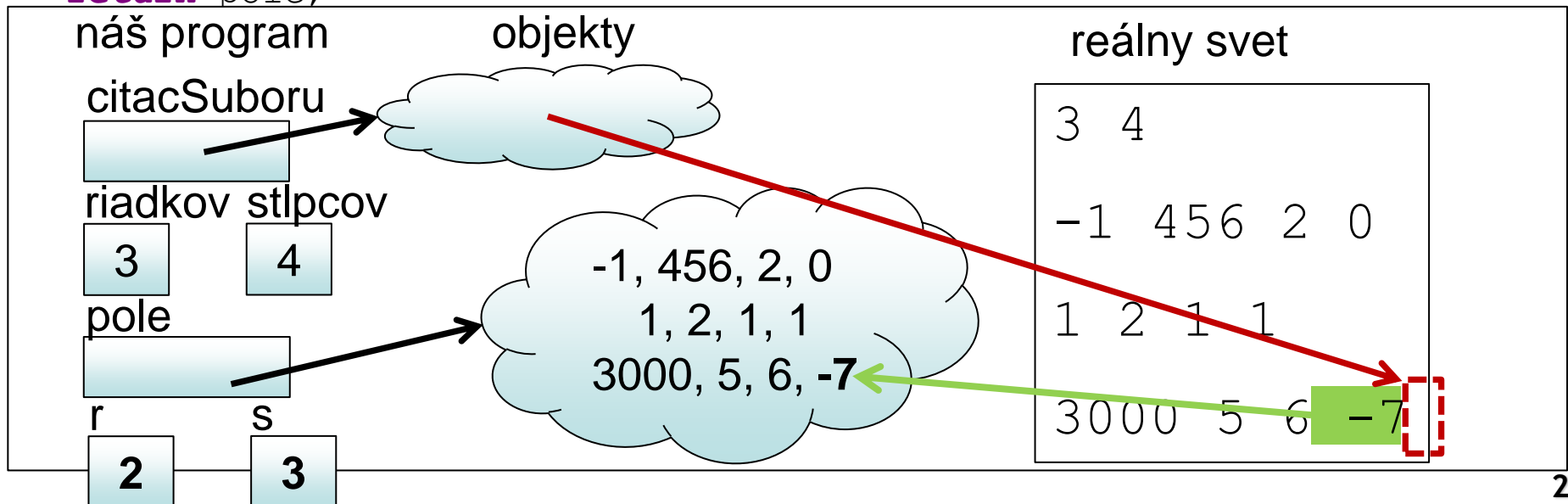
```
public int[][] nacistajMaticu(File subor) {
    Scanner citacSuboru = null;
    try {
        citacSuboru = new Scanner(subor);
        int riadkov = citacSuboru.nextInt();
        int stlpcov = citacSuboru.nextInt();
        int[][] pole = new int[riadkov][stlpcov];
        for (int r=0; r < riadkov;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





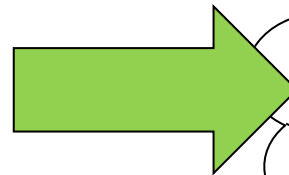
Zasa čítame maticu zo súboru

- 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;

```

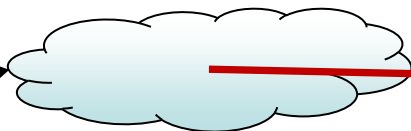


náš program

citacSuboru



objekty



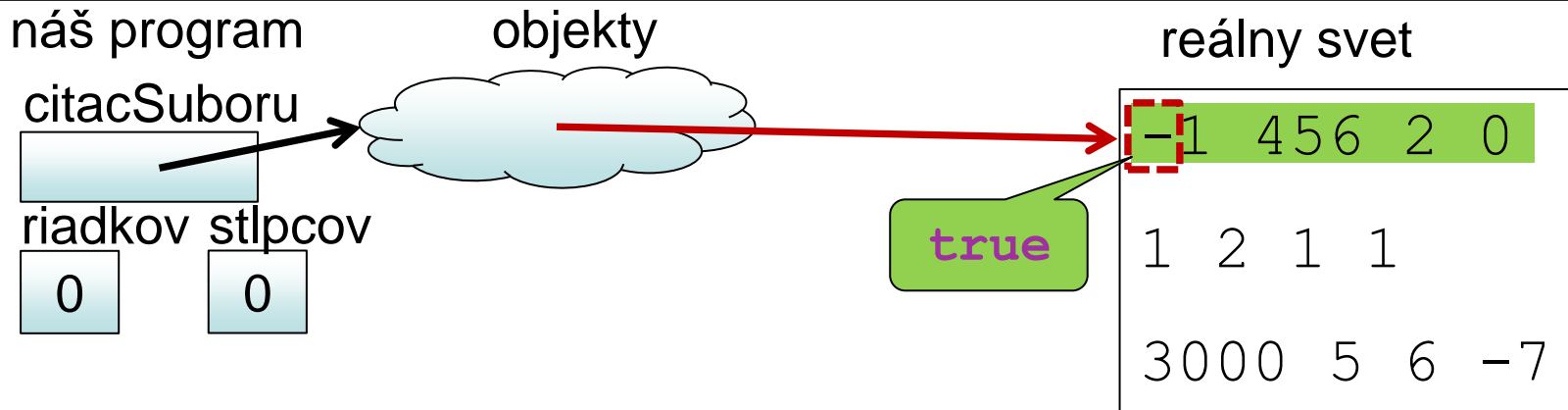
reálny svet

-1	4	5	6	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;
```





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

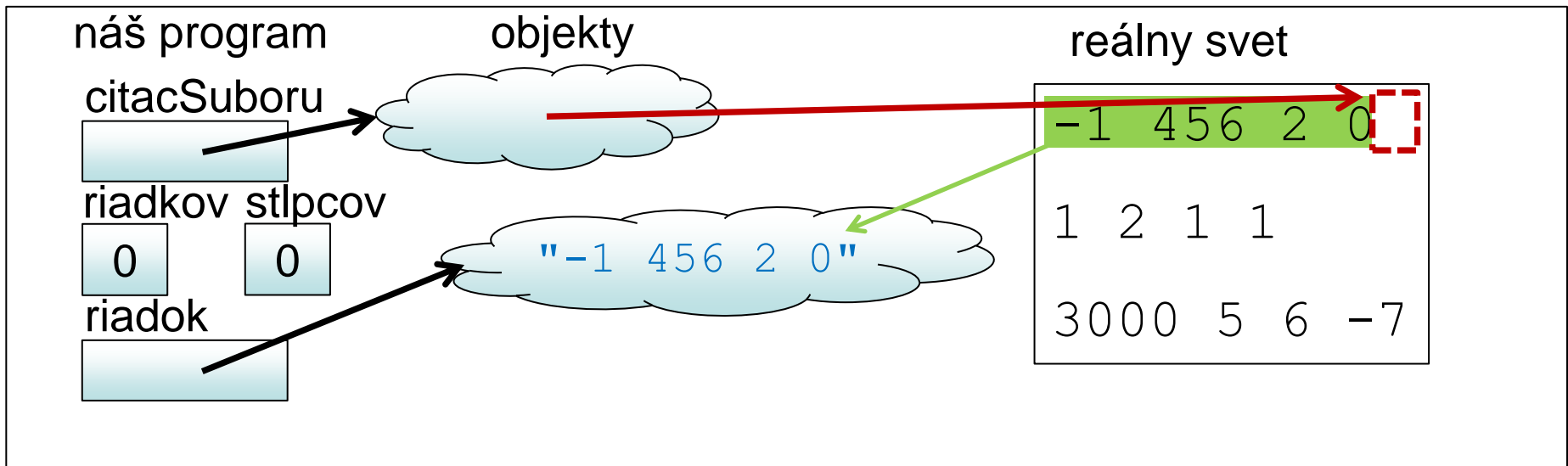
objekty

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

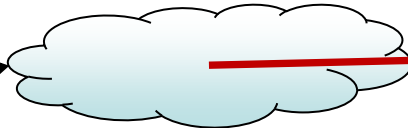
1

0

riadok



objekty



reálny svet

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



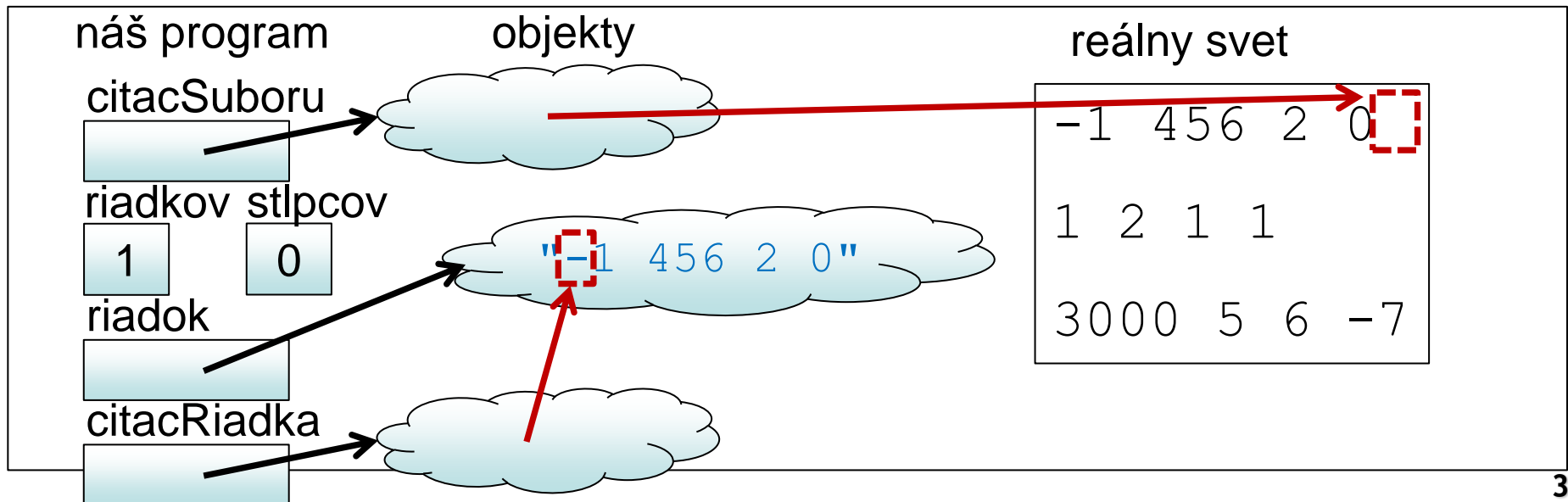


Zisťujeme veľkosť matice v súbore

```

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

```



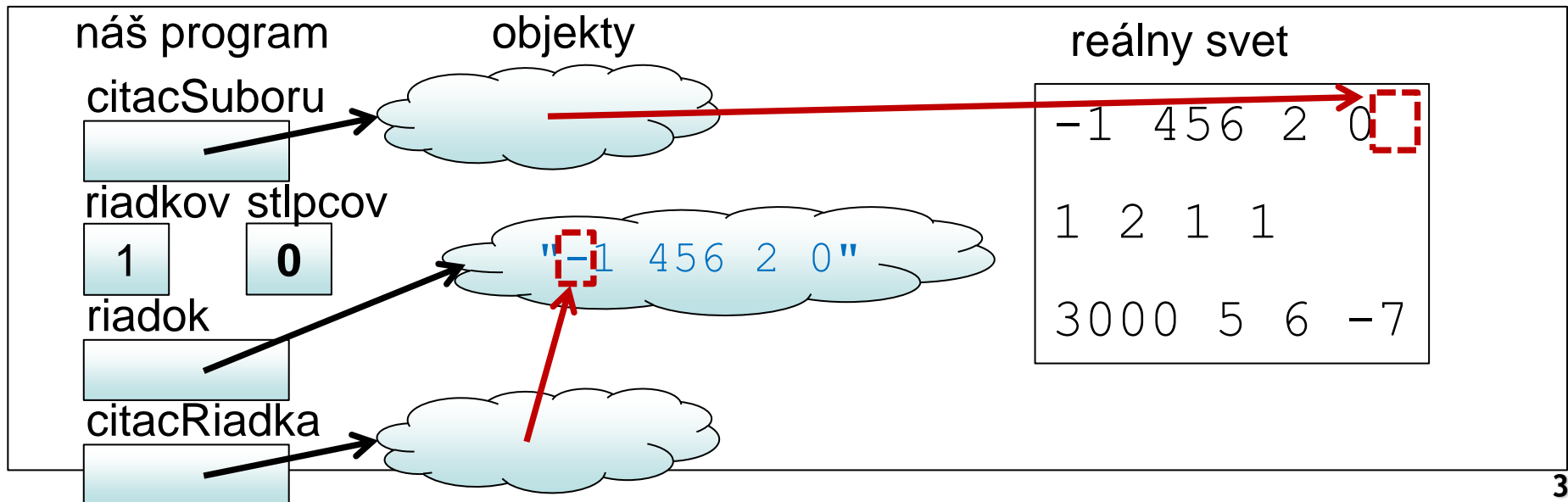


Zisťujeme veľkosť matice v súbore

```

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

```



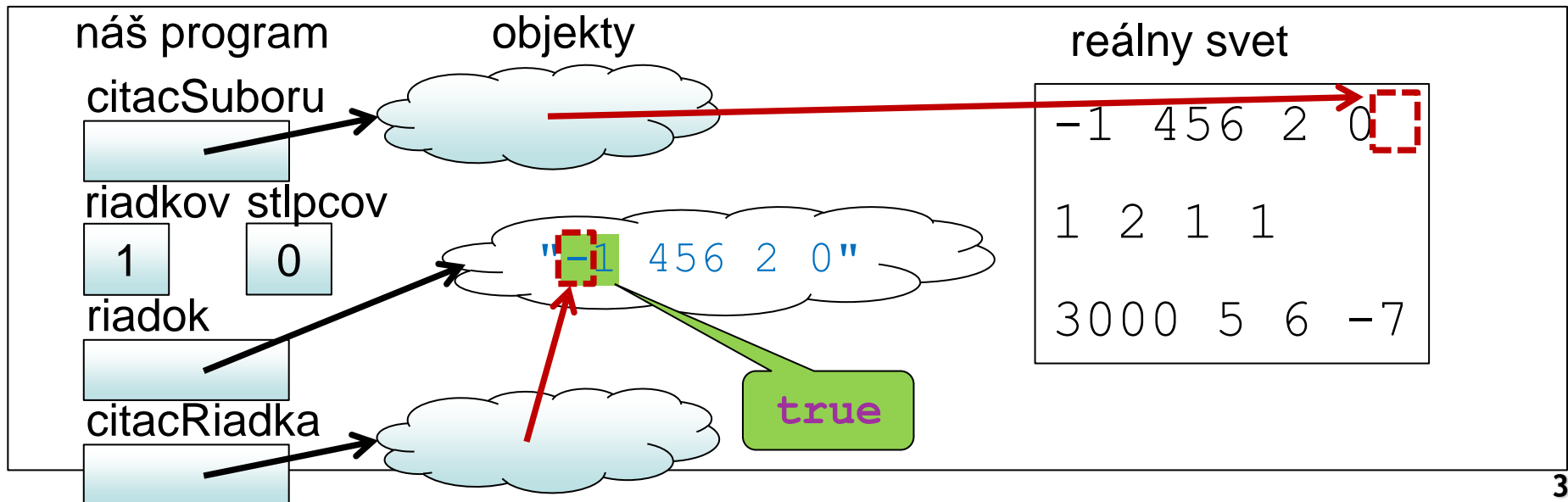


Zisťujeme veľkosť matice v súbore

```

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

```

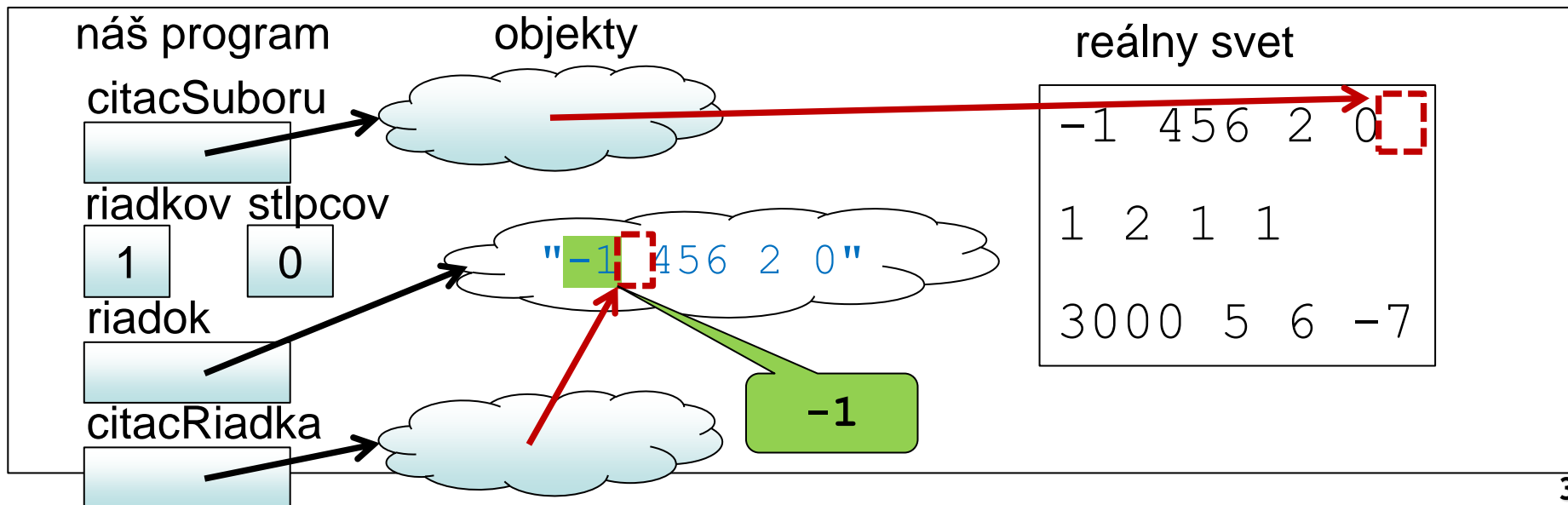




Zisťujeme veľkosť matice v súbore

```

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



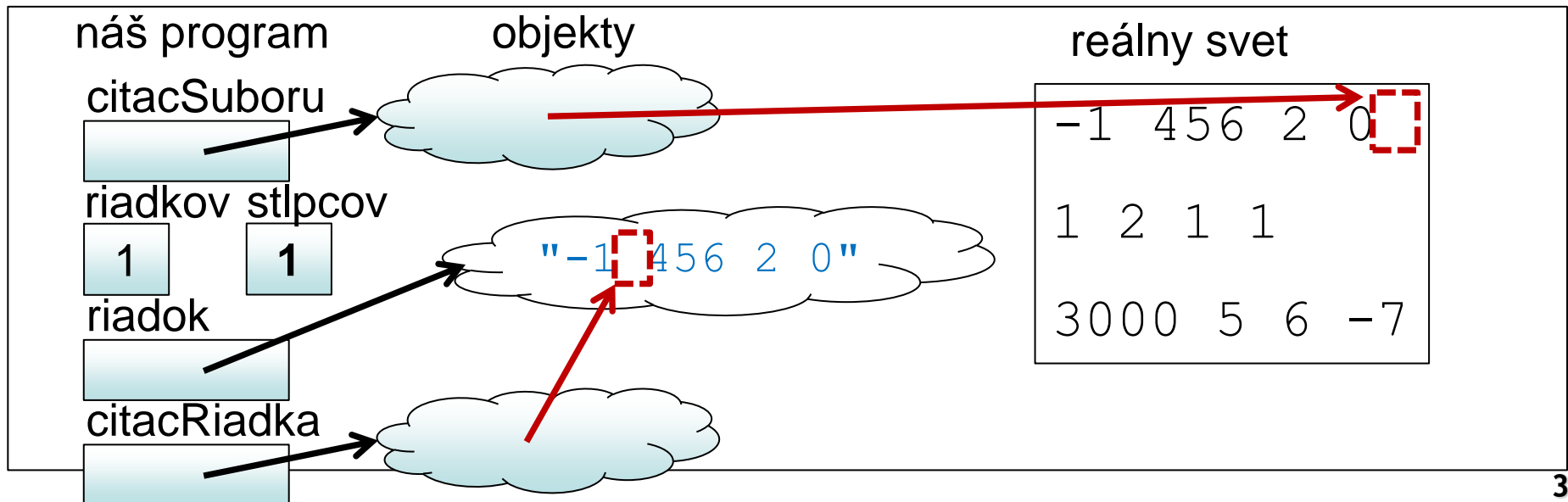


Zisťujeme veľkosť matice v súbore

```

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

```

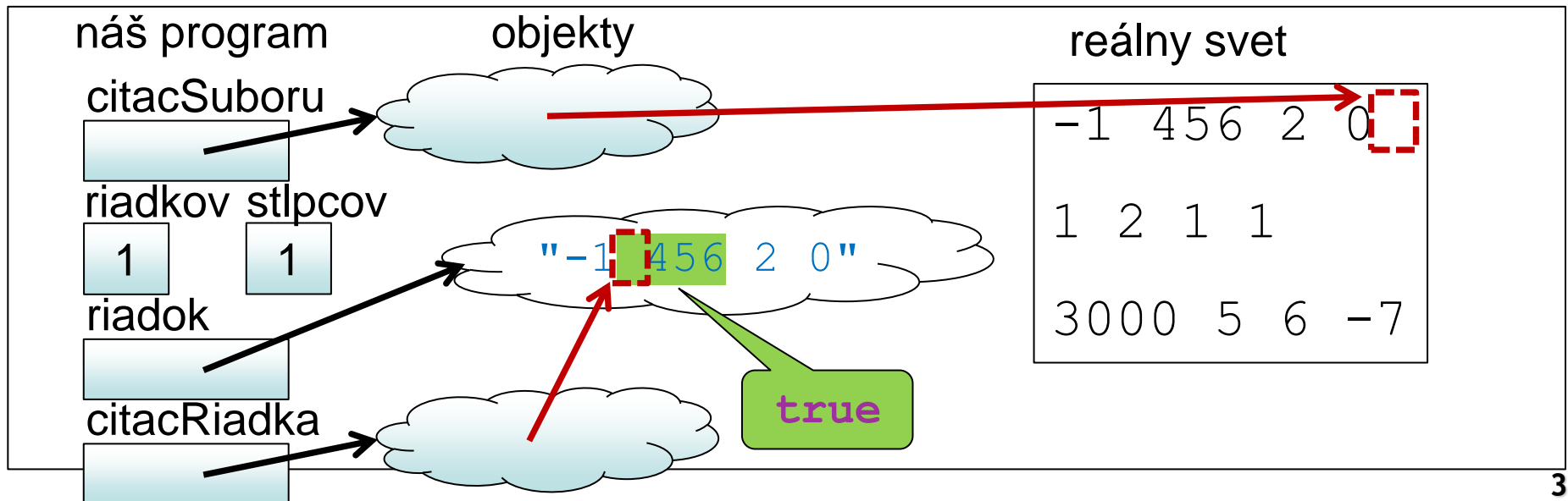




Zisťujeme veľkosť matice v súbore

```

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

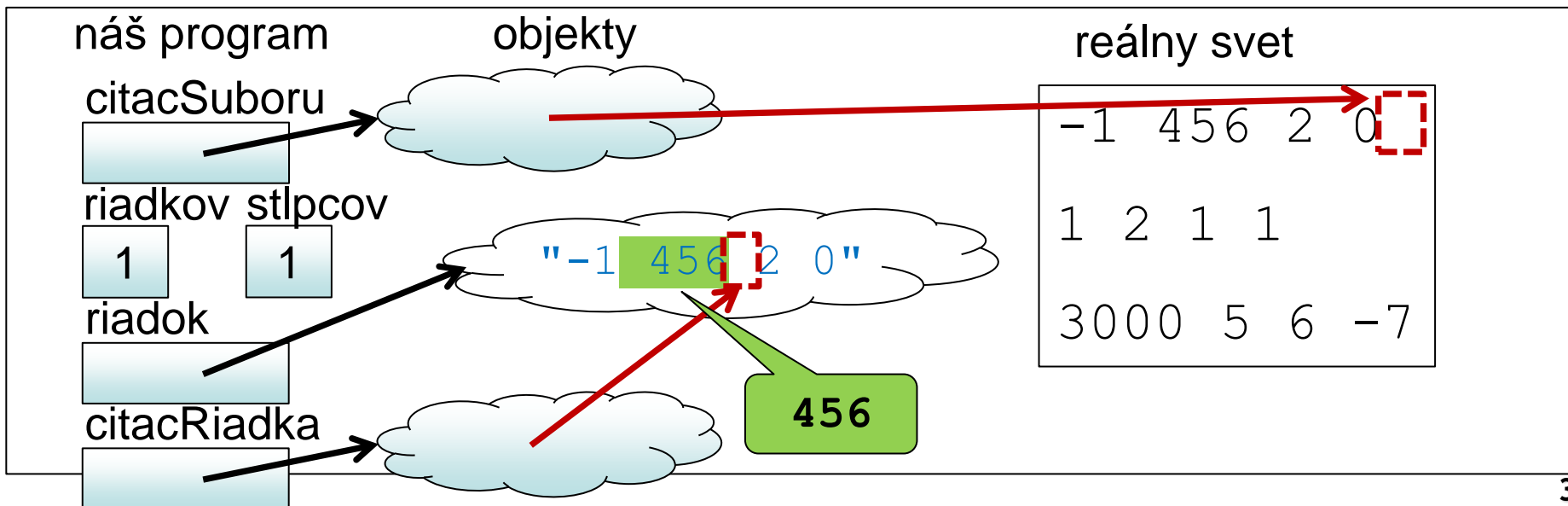




Zisťujeme veľkosť matice v súbore

```

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

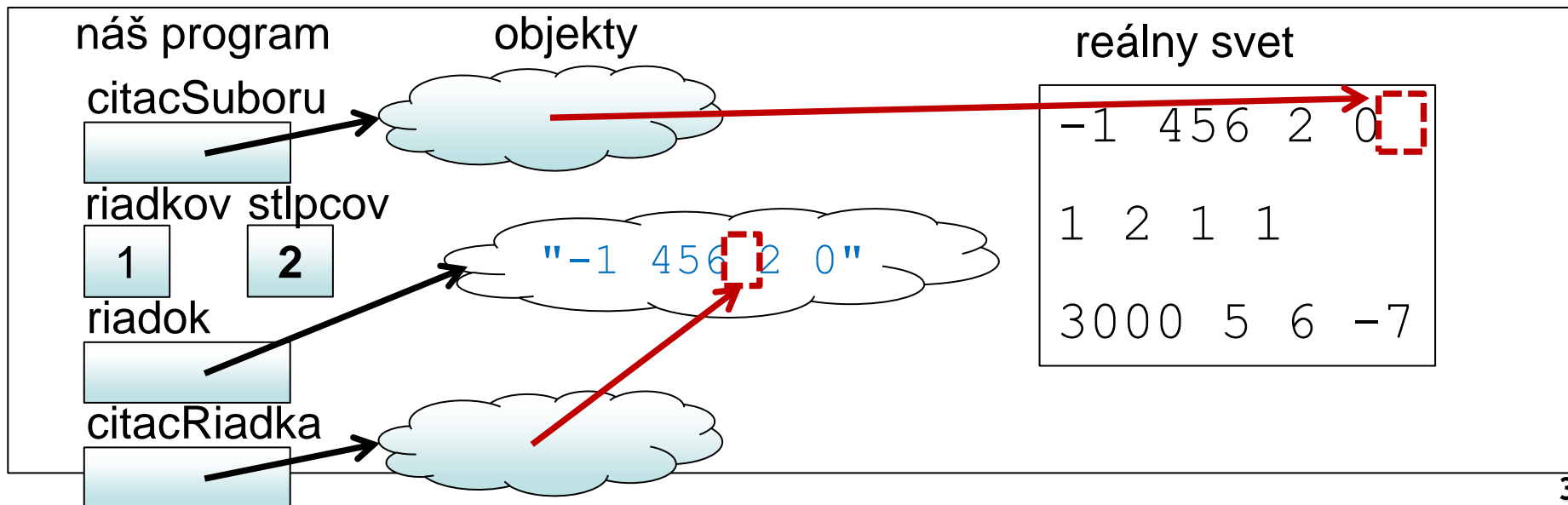




Zisťujeme veľkosť matice v súbore

```

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

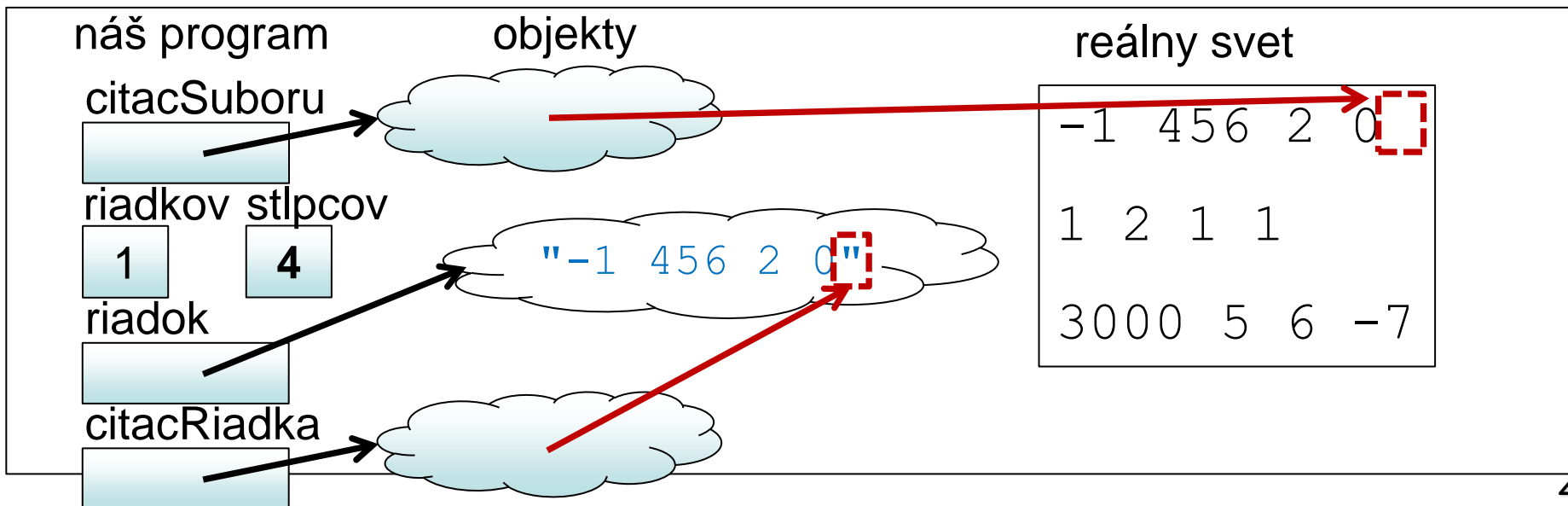




Zisťujeme veľkosť matice v súbore

```

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

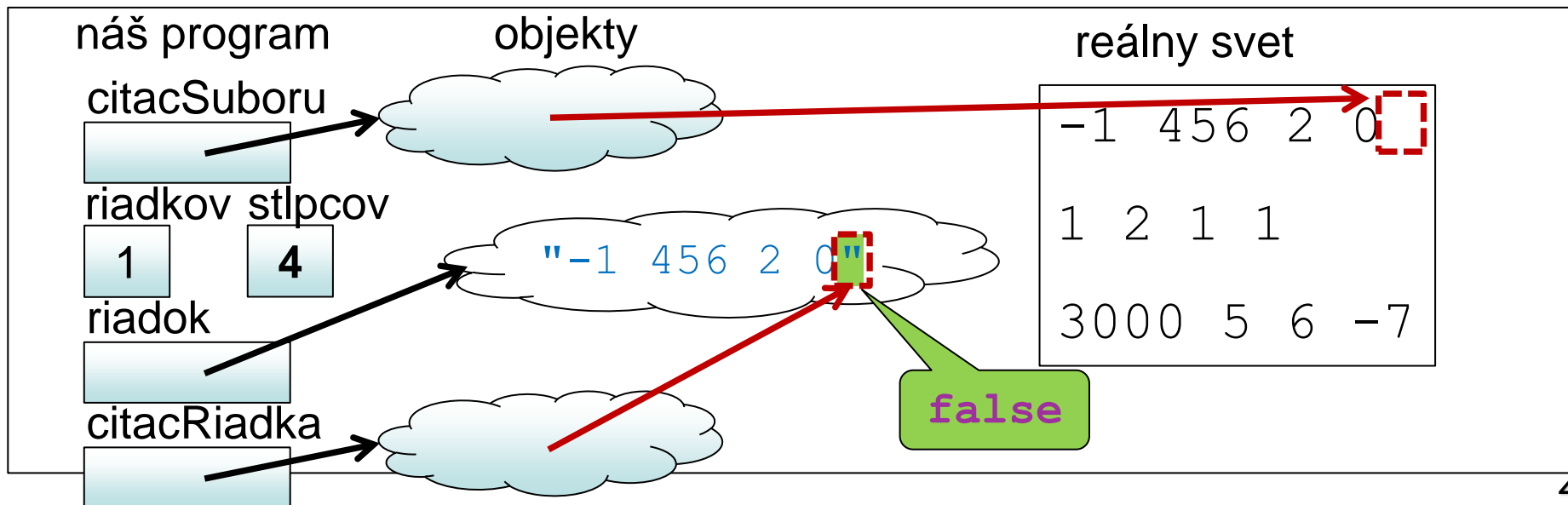




Zisťujeme veľkosť matice v súbore

```

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

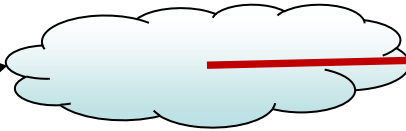
citacSuboru

riadkov stlpcov

1

4

objekty



reálny svet

-1 456 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

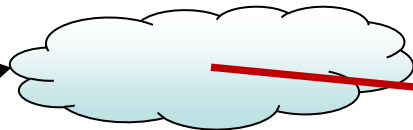
1

4

riadok



objekty



reálny svet

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

"1 2 1 1"

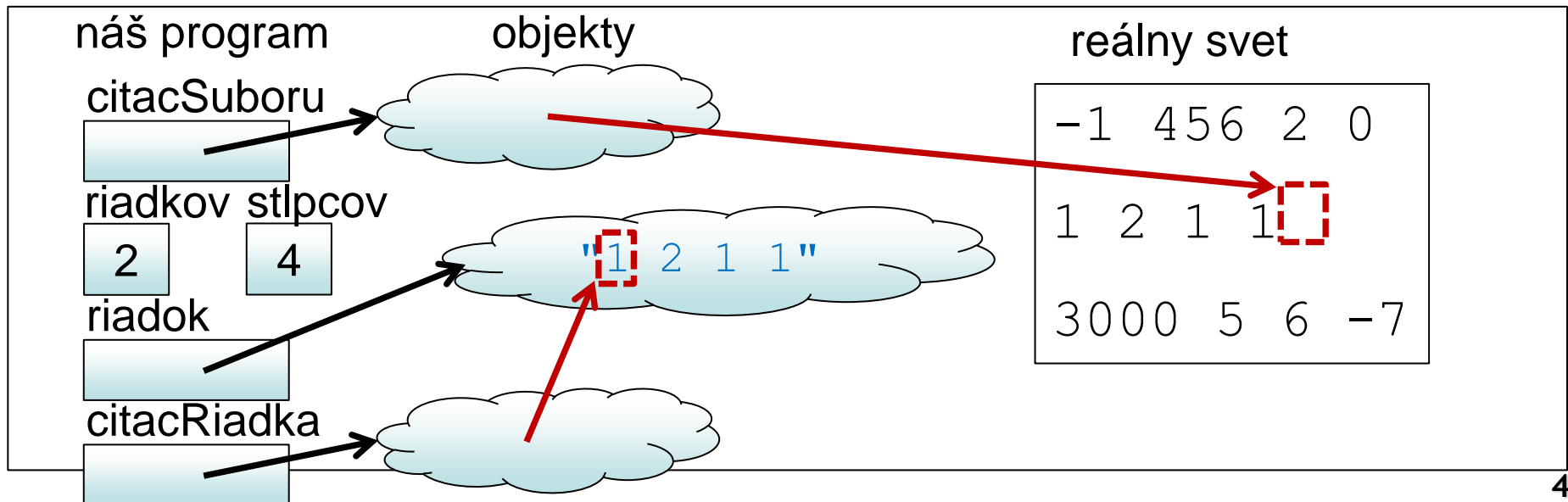


Zisťujeme veľkosť matice v súbore

```

...
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

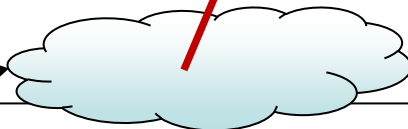
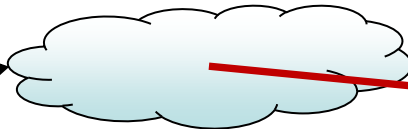
riadok



citacRiadka



objekty

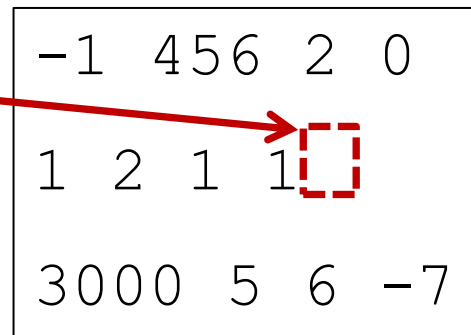


reálny svet

-1 456 2 0

1 2 1 1

3000 5 6 -7



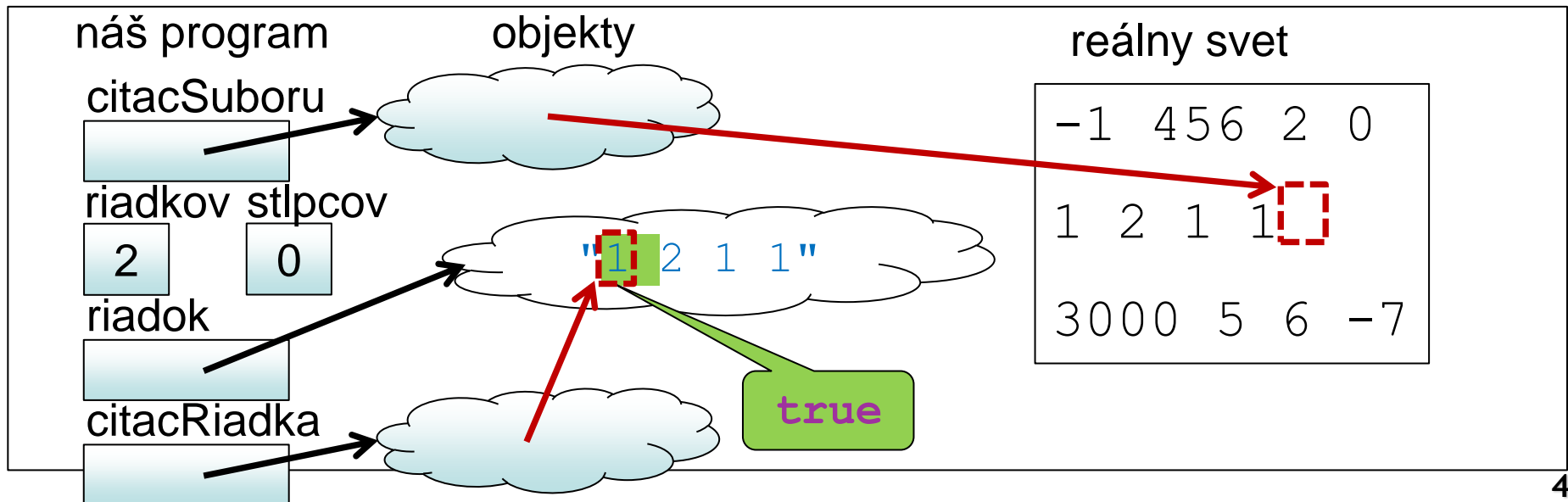


Zisťujeme veľkosť matice v súbore

```

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

```



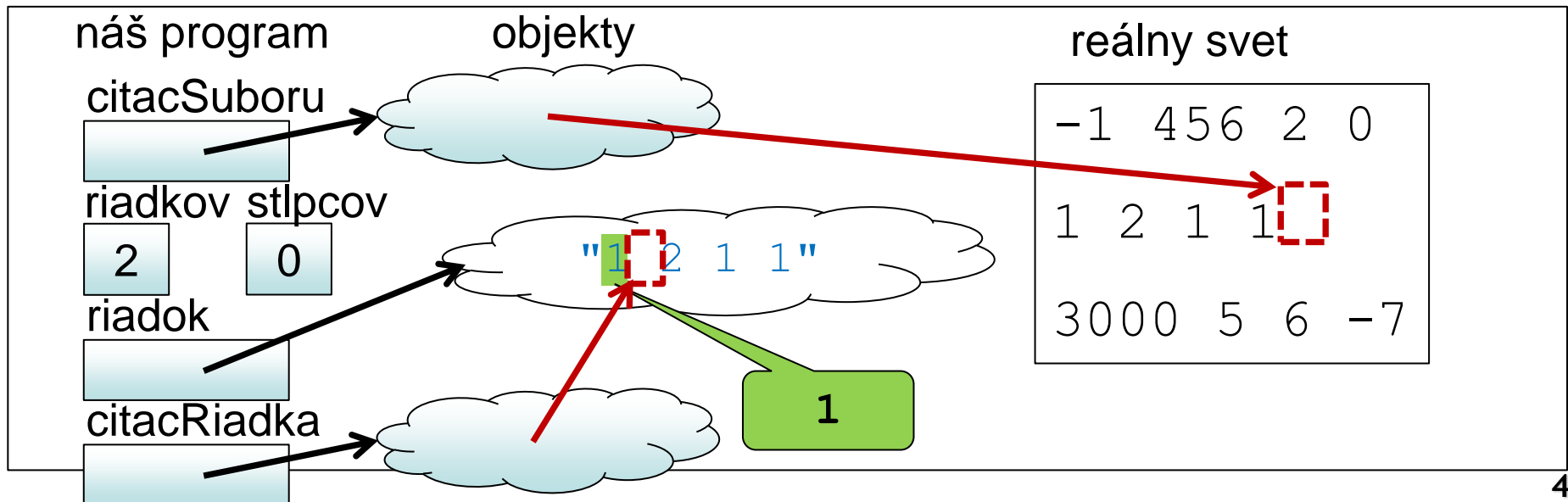


Zisťujeme veľkosť matice v súbore

```

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

```



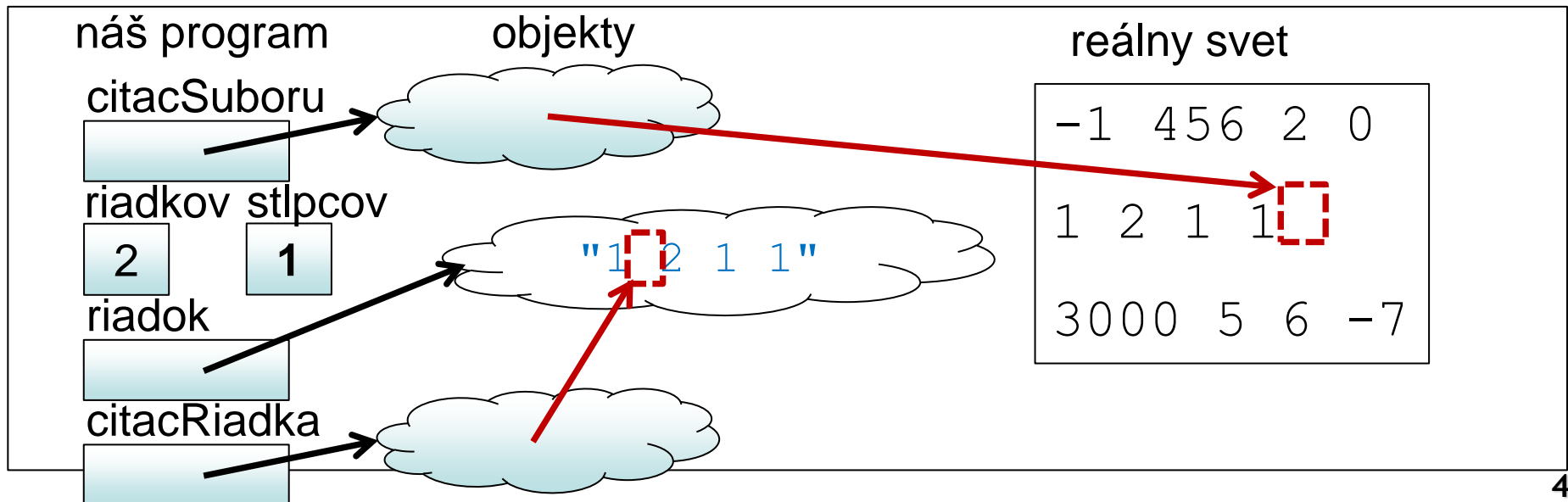


Zisťujeme veľkosť matice v súbore

```

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

```



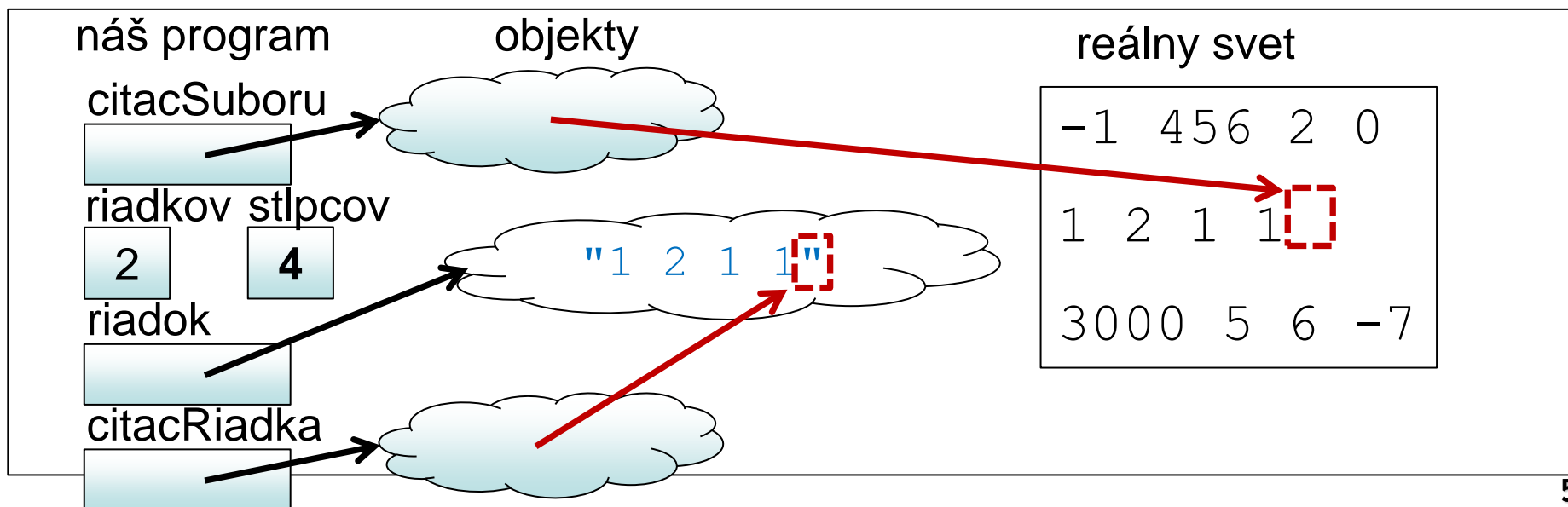


Zisťujeme veľkosť matice v súbore

```

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

```



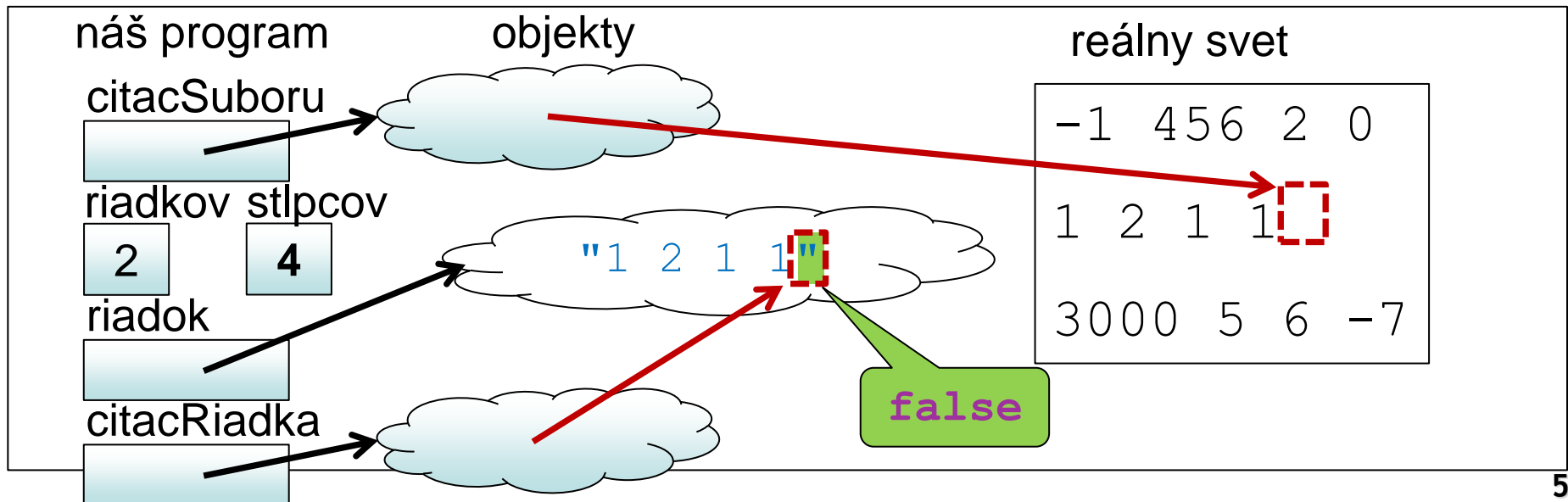


Zisťujeme veľkosť matice v súbore

```

...
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

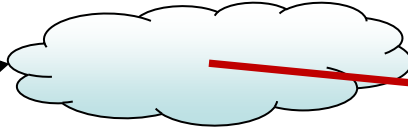
citacSuboru

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

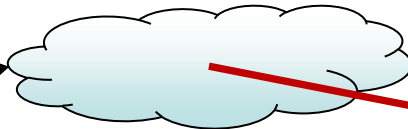
2

4

riadok

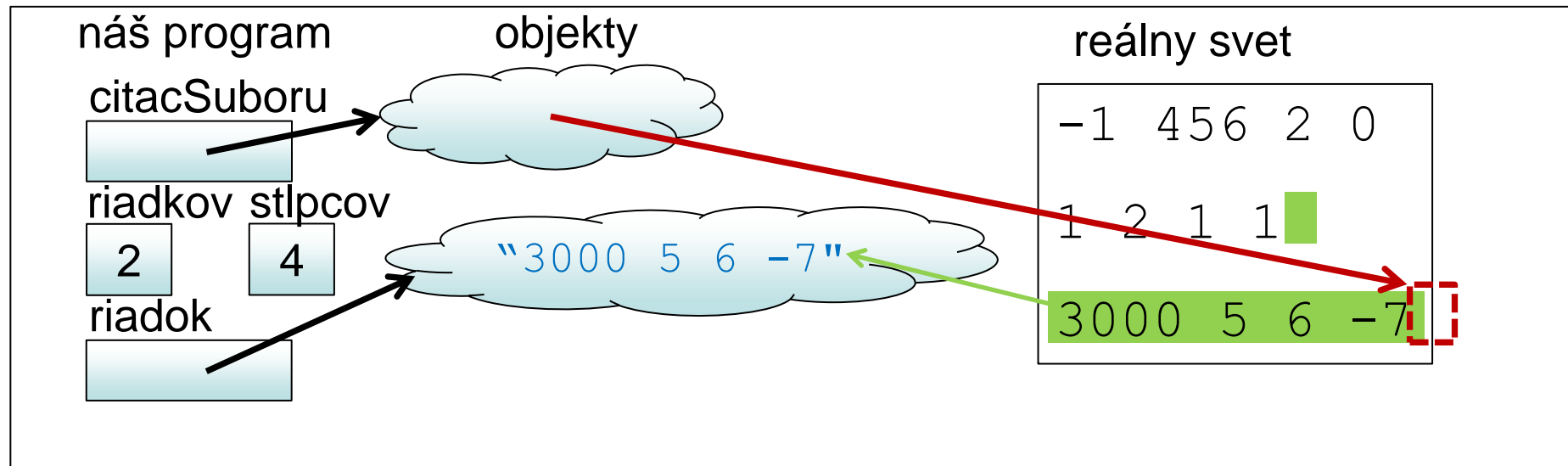


objekty



reálny svet

-1	4	5	6	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

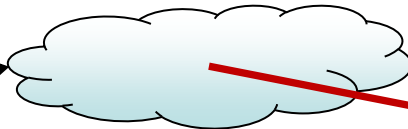
3

4

riadok



objekty



reálny svet

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



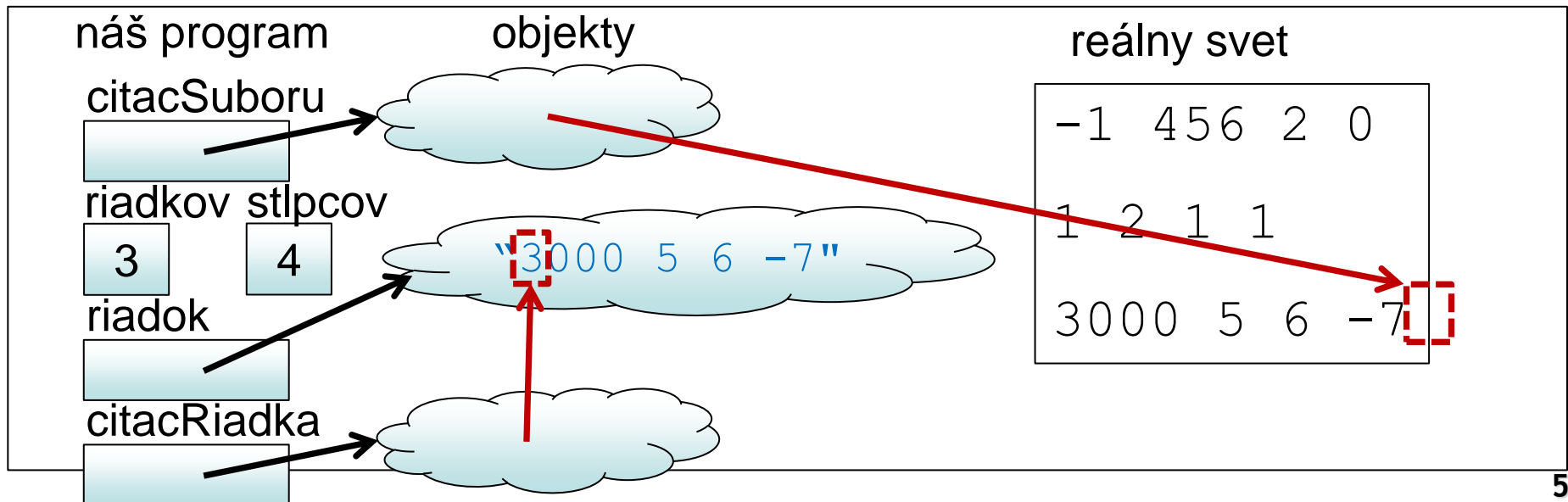


Zisťujeme veľkosť matice v súbore

```

...
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

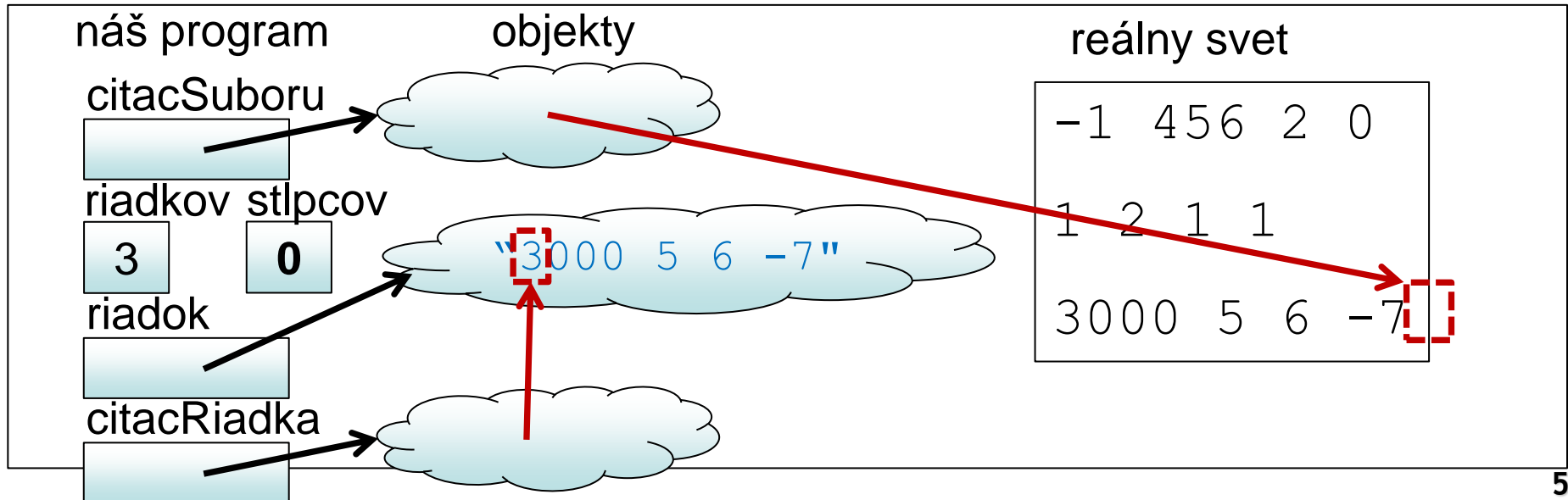
objekty

reálny svet

-1 456 2 0

1 2 1 1

3000 5 6 -7



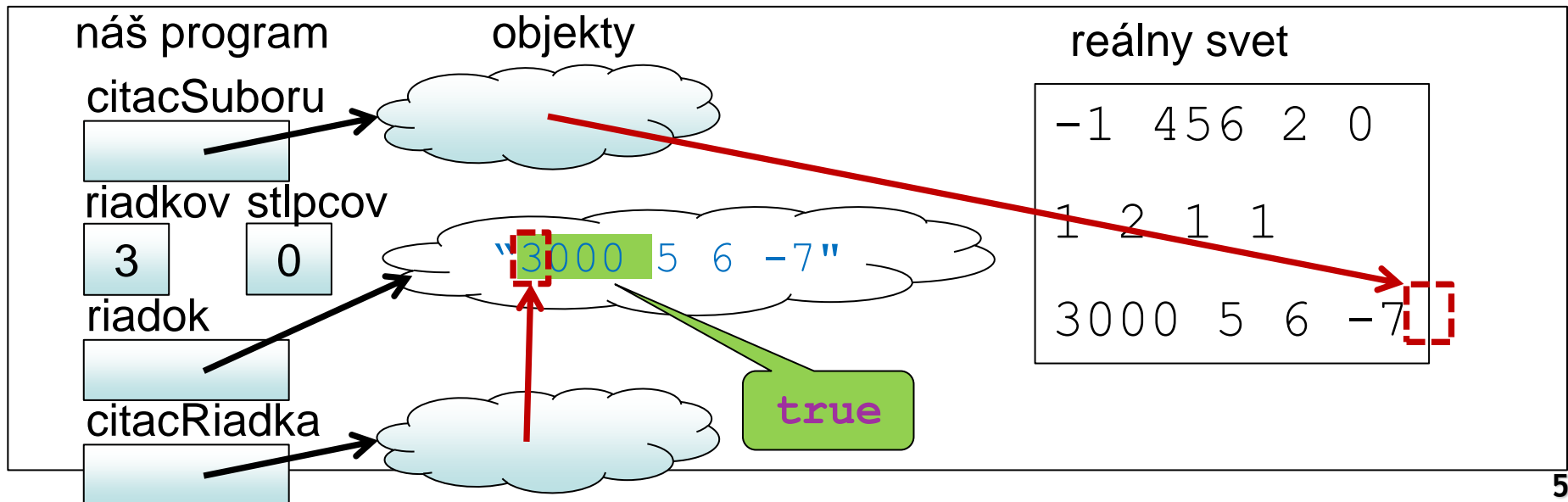


Zisťujeme veľkosť matice v súbore

```

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

```



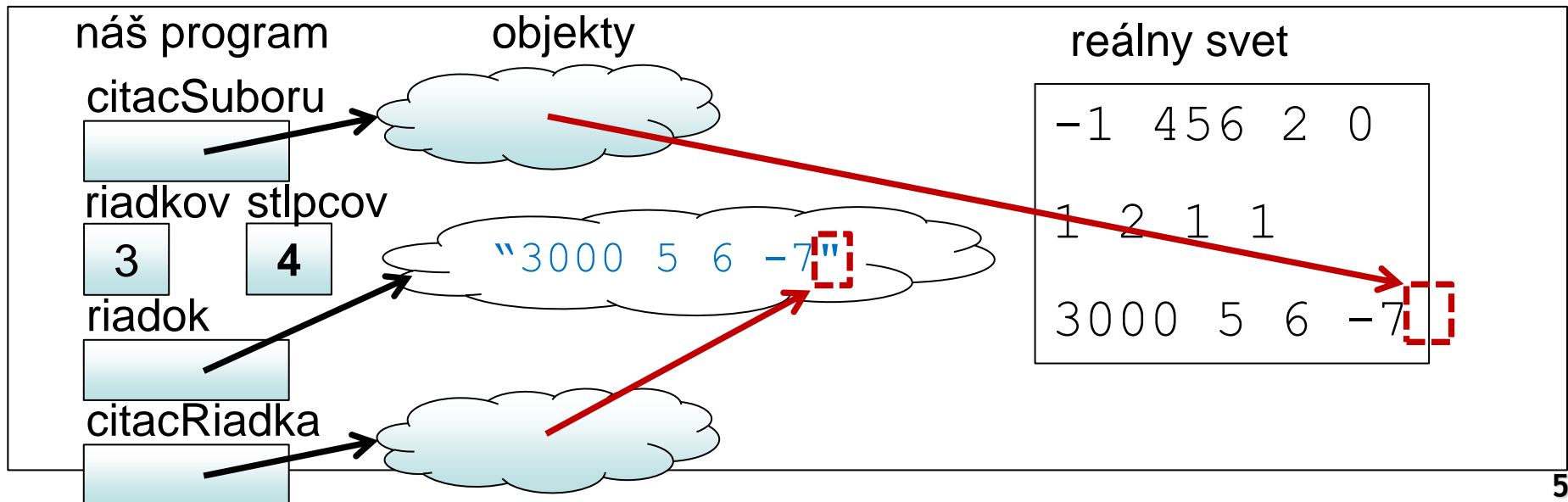


Zisťujeme veľkosť matice v súbore

```

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

```

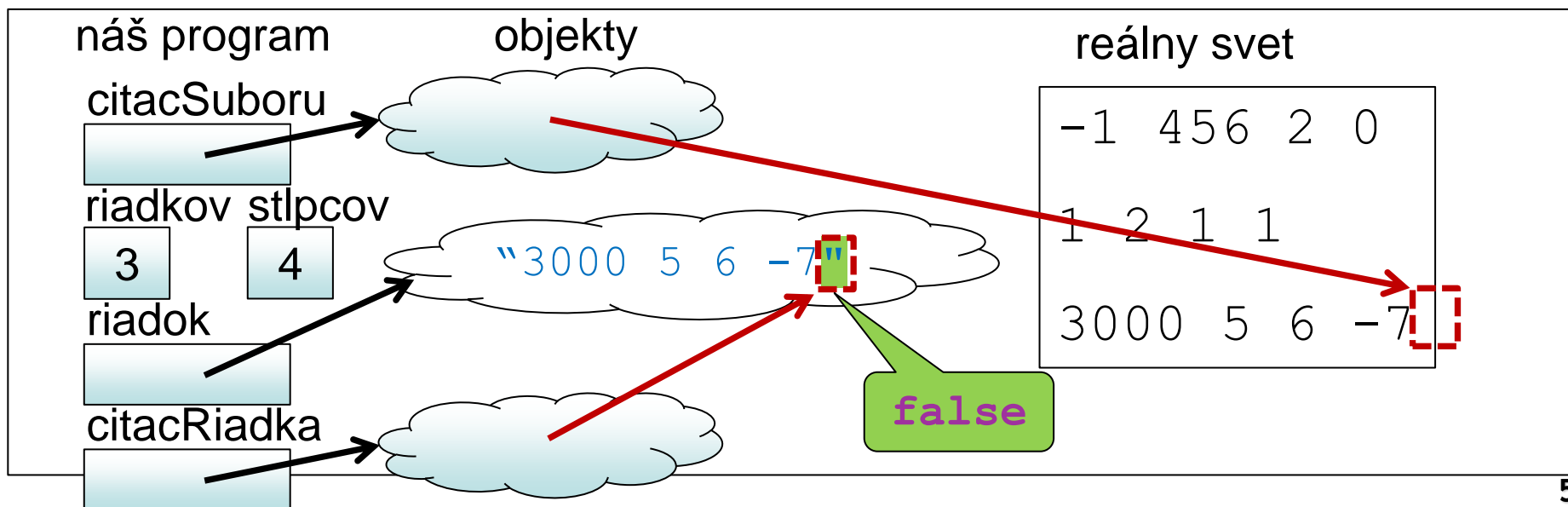




Zisťujeme veľkosť matice v súbore

```

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





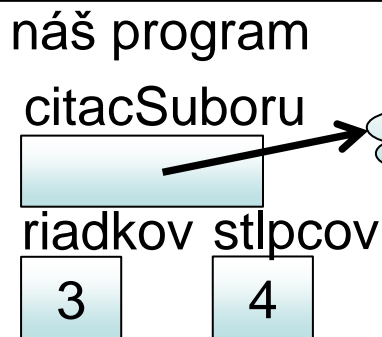
Zisťujeme veľkosť matice v súbore

```

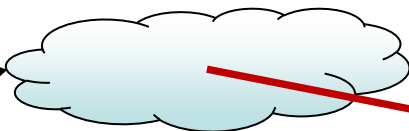
...
while (citacSuboru.hasNextLine()) {
    String riadok = citacSuboru.nextLine();
    riadkov++;
    Scanner citacRiadka = new Scanner(riadok);
    stlpcov = 0;
    while (citacRiadka.hasNextInt()) {
        citacRiadka.nextInt();
        stlpcov++;
    }
}
    
```



Končí vonkajší while cyklus
 riadkov == 3
 stlpcov == 4



objekty



reálny svet

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



Try so zdrojmi - od Javy 7

```
File subor = new File("C:\\adresare\\subor");
try (Scanner scanner = new Scanner(subor)) {
    // píšeme do pw
} catch (FileNotFoundException e) {
    System.out.println("Súbor " +
        subor.getName() + " som nenašiel");
}
```

Do okrúhlych zátvoriek za try inicializujeme premennú zdroja

Po skončení try-catch bloku sa zdroj vždy uzavrie, ak sa ho v try podarilo otvoriť



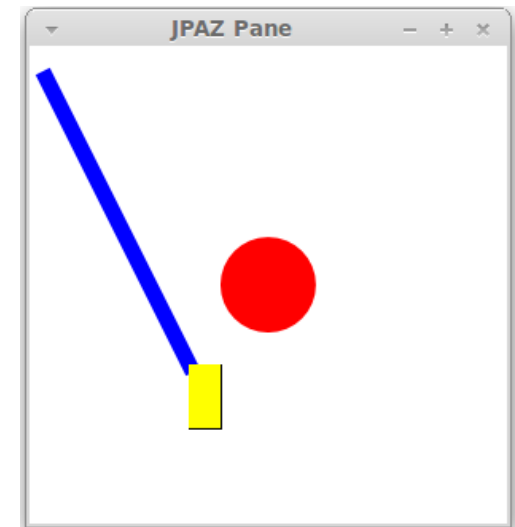
Try so zdrojmi - od Javy 7

```
File subor = new File("C:\\adresare\\subor");  
try (PrintWriter pw = new PrintWriter(subor)) {  
    // píšeme do pw  
  
} catch (FileNotFoundException e) {  
    System.out.println("Súbor " +  
        subor.getName() + " som nenašiel");  
}
```



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 !

