TECHIN

String

Jaroslav Grablevski

Strings

- Simbolių eilutei saugoti plačiai naudojama String klasė iš `java.lang` paketo:

```
public final class String
```

- turi apie 15 konstruktorių ir apie 55 metodų įvairioms operacijoms su simbolių eilute
- Dokumentacija:
  https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html

Strings

- A String is an object that contains a sequence of characters.
- Declaring and instantiating a String is much like any other object variable.
- However, there are differences:
  - They can be instantiated without using the new keyword.
  - They are immutable.
  - Once instantiated, they are final and cannot be chang**ed.**

Strings

- String nėra primityvus tipas, bet speciali klasė (objektas), su kuria Java kompiliatorius leidžia naudoti platesnį spektrą operacijų nei su kitomis klasėmis
- String skirtumai nuo kitų klasių:
  - objektą galimą sukurti nenaudojant žodžio new
  - jie yra nekintami (immutable)

```
String str1 = "Labas!";
String str2 = new String("Labas!");
```

Strings

```
String simpleString = "It is a simple string"; // a simple string
String s = "s";           // a string consisting of one character
String escape = "This is\na multiple\nstring"; // with escape sequences

String emptyString = ""; //empty string
String nullString = null; // it is null

// it creates an object and assigns it to the variable
String str = new String("my-string");
```

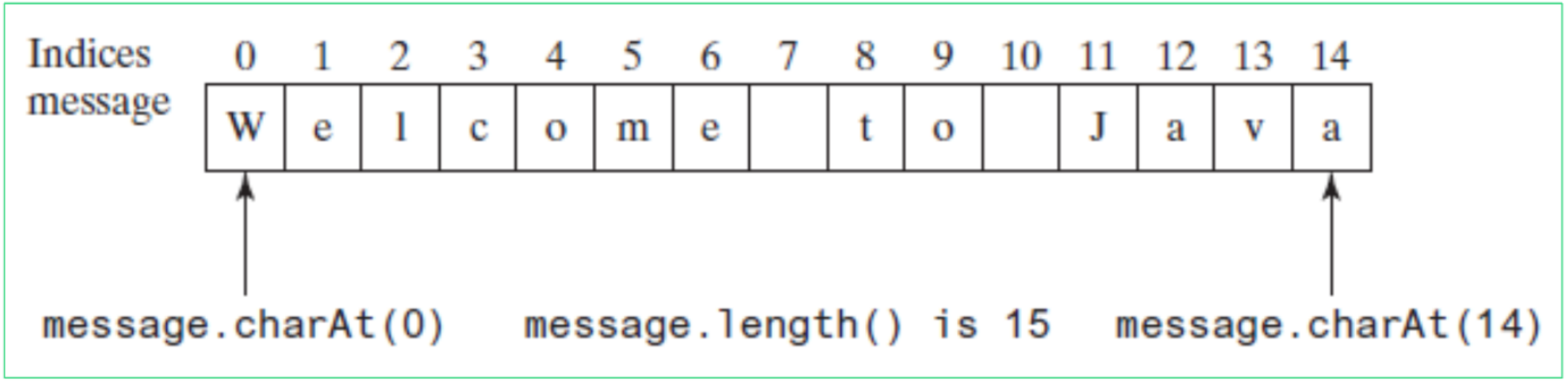
Strings

- String duomenų tipas sudarytas iš char masyvo

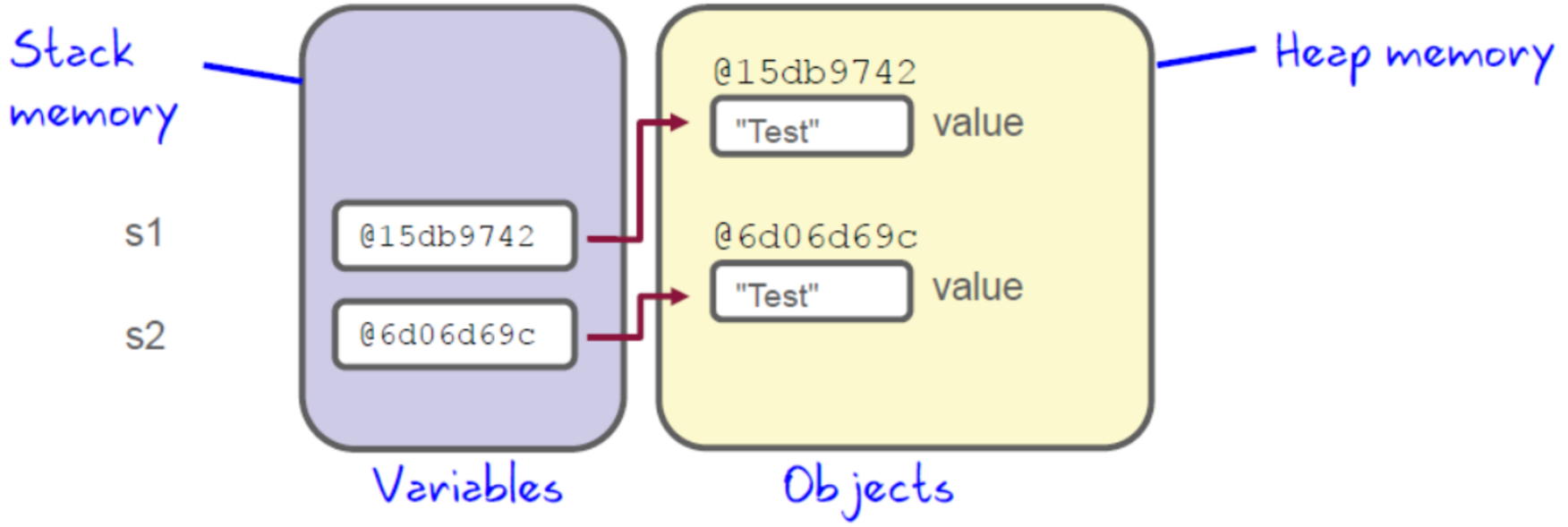
```
char[] charMasyvas = new char[] {'L', 'a', 'b', 'a', 's', '!'};
String str = new String(charMasyvas);
System.out.println(str);
```

```
char[] array = { 'A', 'B', 'C', 'D', 'E', 'F' };

String string = String.valueOf(array); // "ABCDEF"

char[] chars = string.toCharArray(); // { 'A', 'B', 'C', 'D', 'E', 'F' }
```

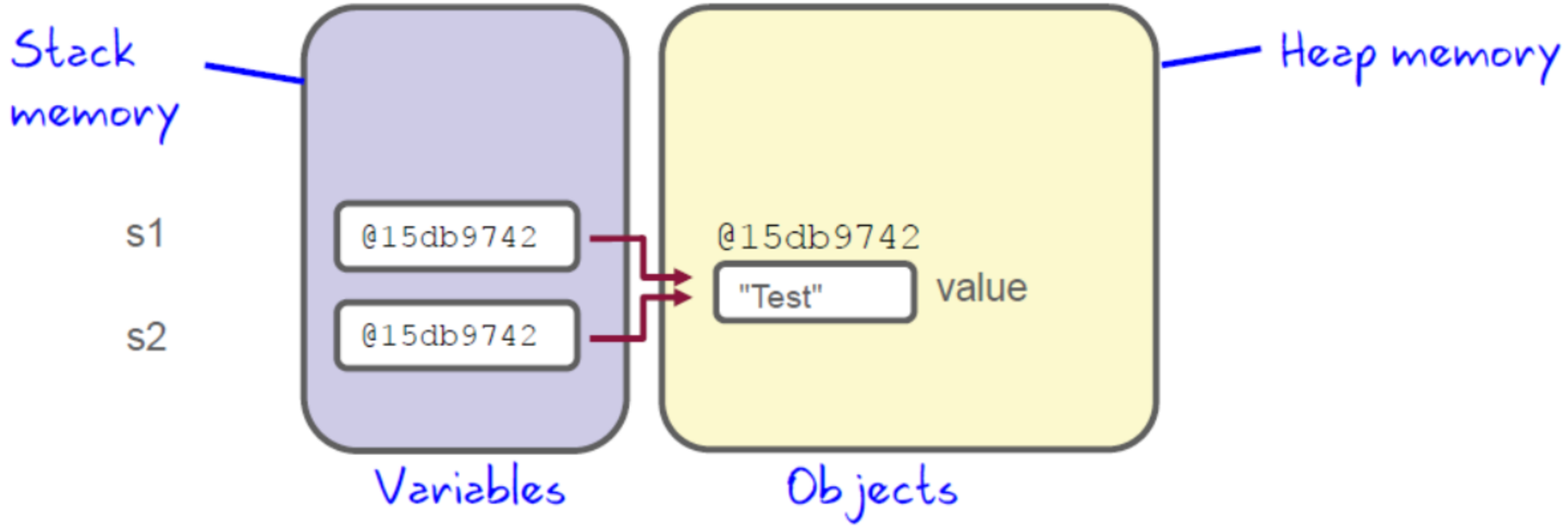
Strings


Indices
message
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
W e l c o m e t o J a v a
message.charAt(0)
message.length() is 15
message.charAt(14)

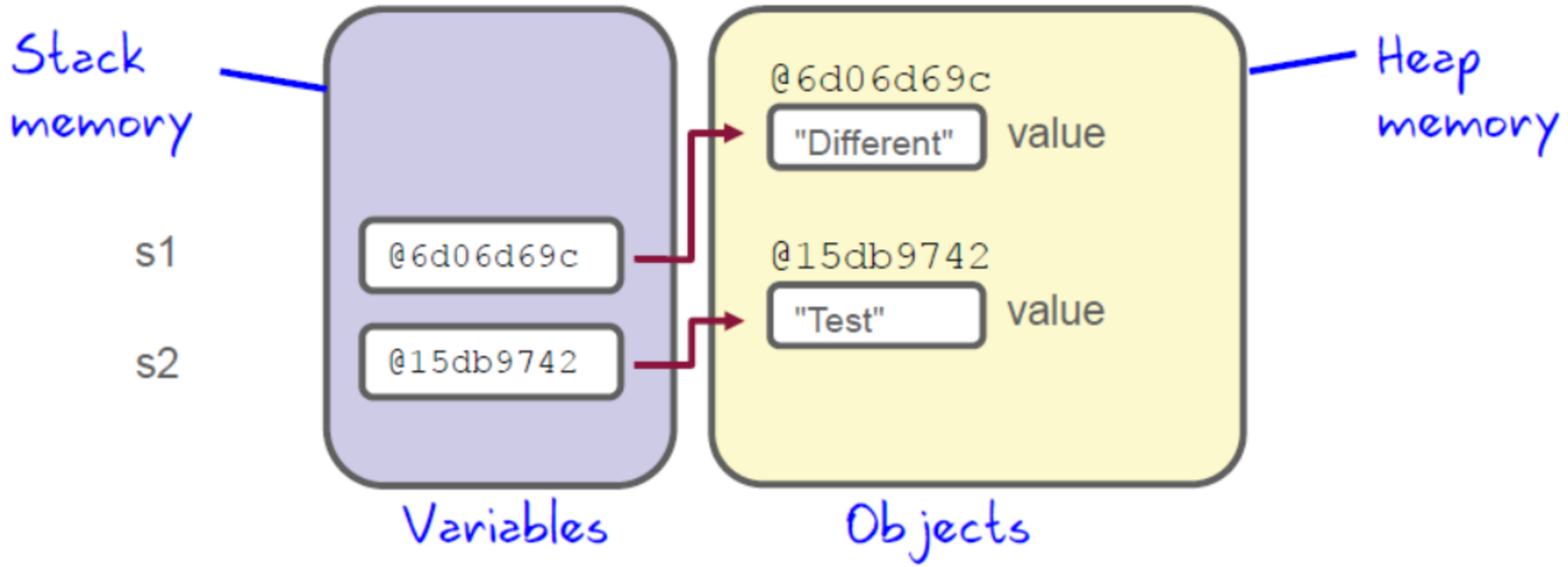
```
String s1 = new String("Test");
String s2 = new String("Test");
```



```
String s1 = "Test";
String s2 = "Test";
```



```
String s1 = "Test";
String s2 = "Test";
s1 = "Different";
```



String metodai (equals)

```
String s1 = "Java programavimas";
String s2 = "Java programavimas";
String p = "programavimas";
String s3 = "Java " + p;


System.out.println("s1==s2: " + (s1 == s2));
System.out.println("s2==s3: " + (s2 == s3));
System.out.println("s2.equals(s3): " + (s2.equals(s3)));
```

```
s1==s2: true
s2==s3: false
s2.equals(s3): true
```

Concatenating strings

```
String s1="Java", s2="Programavimas";
//Using concat method to concatenate two strings
String s3 = s1.concat(s2);//s3 becomes JavaProgramavimas
// Three strings are concatenated
String message = "Welcome " + "to " + "Java";
// String Chapter is concatenated with number 2
String s = "Chapter" + 2; // s becomes Chapter2
// String Supplement is concatenated with character B
String s4 = "Supplement" + 'B'; // s4 becomes SupplementB
```

Concatenating strings

```
String shortString = "str";
int number = 100;

String result1 = shortString + number + 50;
String result2 = number + 50 + shortString;
```

```
str10050
150str
```

String metodai

Metodas	Aprašymas
charAt(int index)	Grąžina simbolį kurį turi index
equals(Object anObject)	Lygina string su objektu
equalsIgnoreCase(String anotherString)	Lygina nepaisant raidžių dydžio
indexOf(int ch)	Grąžina numerį pirmojo nurodytosios eilutės simbolio
indexOf(String str)	Grąžina vietos numerį, nuo kurios prasideda ieškomas žodis
length()	Grąžina eilutė ilgį
substring(int beginIndex, int endIndex)	Grąžina naują eilutę, kuri yra eilutės pogrupis

String metodai

```
String word = "Java programavimas";
char character = word.charAt(3);
```

```
a
```

```
int wordLength = word.length();
```

```
char character2 = word.charAt(word.length());
```

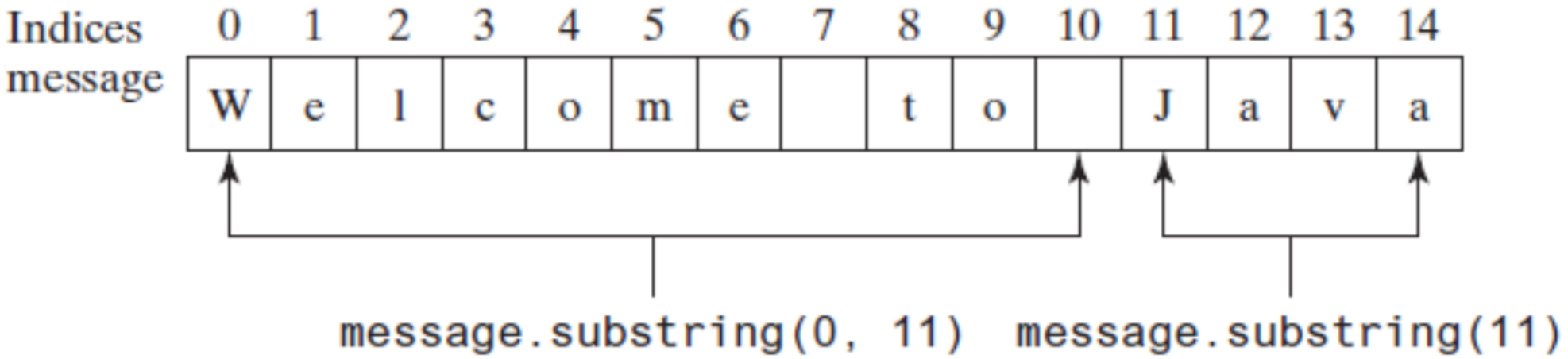
```
Exception in thread "main" java.lang.StringIndexOutOfBoundsException: String index out of range: 18
        at java.lang.String.charAt(Unknown Source)
```

```
char character2 = word.charAt(word.length()-1);
```

String metodai (substring)

```
String word = "Java programavimas";

System.out.println(word.substring(5));
System.out.println(word.substring(0, 4));
System.out.println(word.substring(5, 13));
```

```
programavimas
Java
programa
```



String metodai (indexOf)

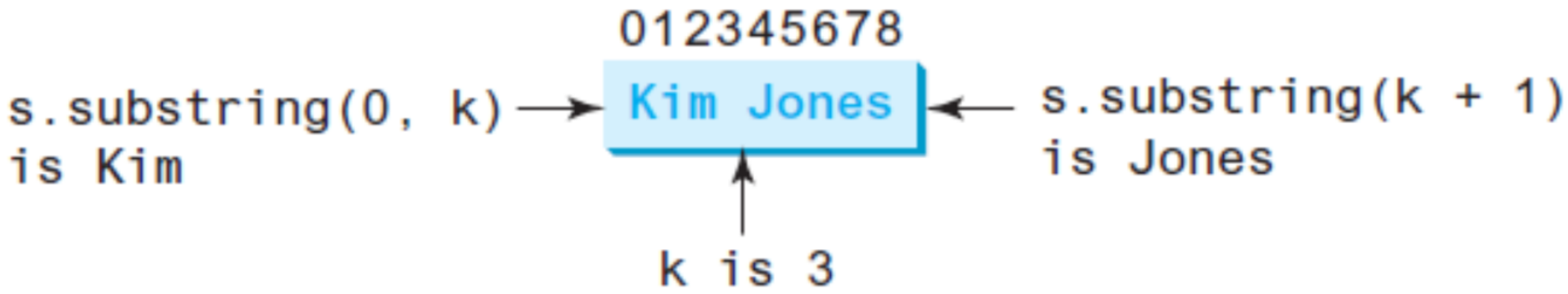
```
String word = "Java programavimas";
```

```
int index = word.indexOf("pro");
System.out.println(index);
System.out.println(word.substring(index));
System.out.println(word.indexOf("css"));
```

```
5
programavimas
-1
```

String metodai (indexOf)

```
String s = "Kim Jones";
int k = s.indexOf(' ');
String firstName = s.substring(0, k);
String lastName = s.substring(k + 1);
```



String metodai (split)

```
String text = "first second third fourth";
String[] pieces = text.split(" ");
System.out.println(pieces[0]);
System.out.println(pieces[1]);
System.out.println(pieces[2]);
System.out.println(pieces[3]);

System.out.println();

for (int i = 0; i < pieces.length; i++) {
    System.out.println(pieces[i]);
}
```

```
first
second
third
fourth

first
second
third
fourth
```

Specialūs char simboliai

Java Escape Sequences

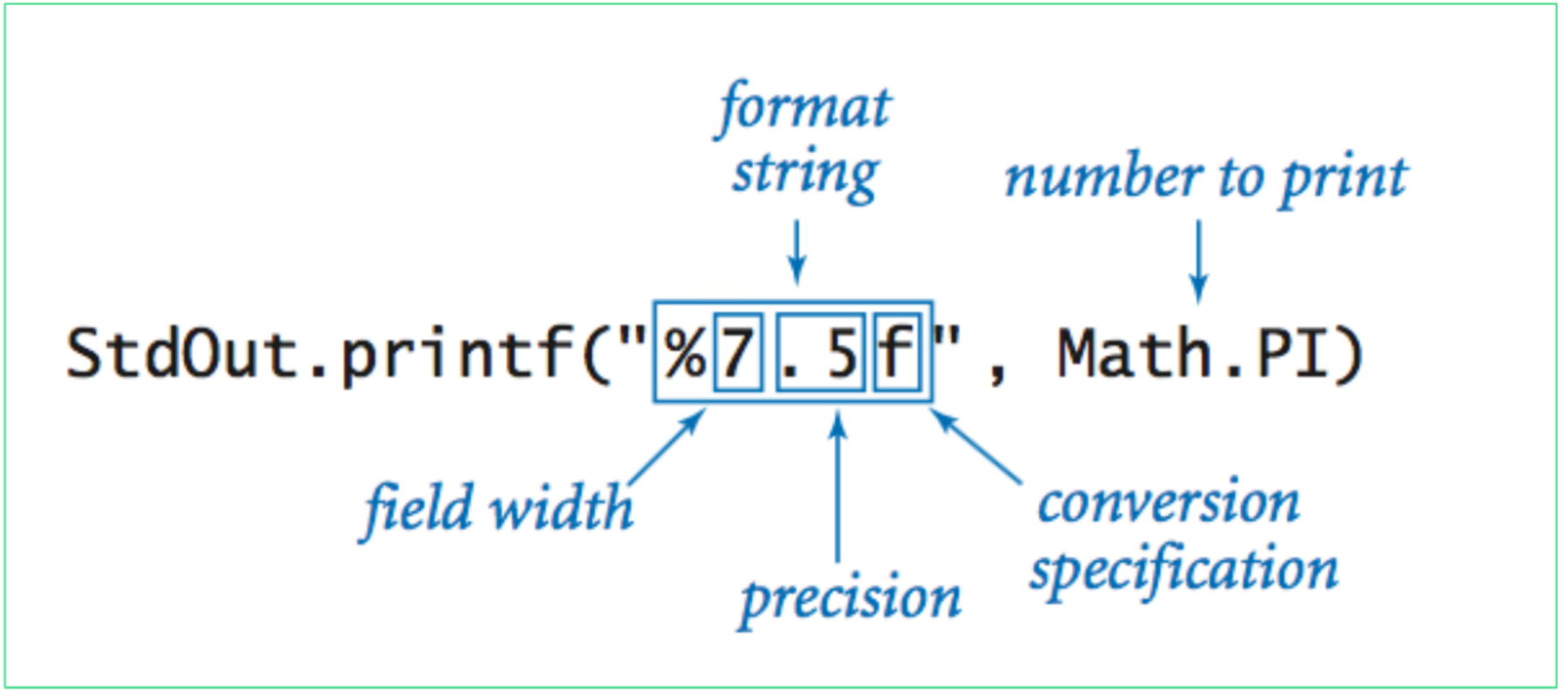
`\n`	newline	Advances the cursor to the next line for subsequent printing
`\t`	tab	Causes the cursor to skip over to the next tab stop
`\b`	backspace	Causes the cursor to back up, or move left, one position
`\r`	carriage return	Causes the cursor to go to the beginning of the current line, not the next line
`\\`	backslash	Causes a backslash to be printed
`\'`	single quote	Causes a single quotation mark to be printed
`\"`	double quote	Causes a double quotation mark to be printed

String metodai (format)

```
String output = String.format("Vardas: %s, amžius: %d", "Jonas", 35);
System.out.println(output);
System.out.printf("Vardas: %s, amžius: %d\n", "Petras", 25);

double pi = Math.PI;
System.out.format("%f%n", pi);
System.out.format("%.2f%n", pi);
System.out.format("|%10.3f|%n", pi);
System.out.format("|%-10s|%n", "kaire");
```

```
Vardas: Jonas, amžius: 35
Vardas: Petras, amžius: 25
3.141593
3.14
|     3.142|
|kaire     |
```

String metodai (format)



String metodai (format)

type	*code*	*typical literal*	*sample format strings*	*converted string values for output*
int	d	512	"%14d" "%-14d"	" 512" "512 "
double	f e	1595.1680010754388	"%14.2f" "%.7f" "%14.4e"	" 1595.17" "1595.1680011" " 1.5952e+03"
String	s	"Hello, World"	"%14s" "%-14s" "%-14.5s"	" Hello, World" "Hello, World " "Hello "
boolean	b	true	"%b"	"true"

StringBuilder

```
StringBuilder empty = new StringBuilder();
System.out.println(empty); // ""

StringBuilder sb = new StringBuilder("abc");
System.out.println(sb); // "abc"
sb.append("123");
System.out.println(sb); // "abc123"
```

StringBuilder

```
StringBuilder messageBuilder = new StringBuilder(); // empty

messageBuilder.append("From: Kate@gmail.com\n")
              .append("To: Max@gmail.com\n")
              .append("Text: I lost my keys.\n")
              .append("Please, open the door!");

System.out.println(messageBuilder);
```

StringBuilder

```
StringBuilder sb = new StringBuilder("I use Java");
System.out.println(sb.length()); // 10
System.out.println(sb.charAt(0)); // 'I'

StringBuilder sbc = new StringBuilder("Let's use C#");
sbc.replace(10, 12, "Java");
System.out.println(sbc); // Let's use Java

StringBuilder sbr = new StringBuilder("2 * 3 + 8 * 4");
sbr.reverse();
System.out.println(sb); // "4 * 8 + 3 * 2"
```