

Conversions:-

String to int :- $\text{String s} = \text{Scanner.nextLine();}$ $i = \text{Integer.parseInt}(s);$

int to String :- $\text{String s} = \text{String.valueOf(i);}$

int to String :-

$\text{int i} = 10;$
 $\text{String s} = \text{String.valueOf(i);}$

String to long :- $\text{long l} = \text{long.parseLong}(s);$

long to String :- $\text{long i} = 999999999999999L;$
 $\text{String s} = \text{long.toString(i);}$

String to float :- $\text{float f} = \text{Float.parseFloat}(s);$

float to string :- $\text{float f} = 12.3F;$
 $\text{String s} = \text{String.valueOf(f);}$

String to double :- $\text{double d} = \text{Double.parseDouble}(s);$

double to String :- $\text{double d} = 12.3;$
 $\text{String s} = \text{String.valueOf(d);}$

String to Date:

String sDate = "31/12/1998";
Date date1 = new SimpleDateFormat("dd/MM/yyyy").parse(sDate);

Date to String:

Date date = Calendar.getInstance().getTime();
DateFormat dateFormat = new SimpleDateFormat("yyyy-mm-dd hh:mm:ss");
String strDate = dateFormat.format(date);

String to Char:

String s = "hello";
char c = s.charAt(0);

Char to String:

String s = String.valueOf(c);

String to Object:

String s = "hello";
Object obj = s;

Object to String:

```
Emp e=new Emp();
String s=e.toString();
String sa=String.valueOf(e);
System.out.println(s);
System.out.println(sa);
```

Int to long:

```
int i=200;
```

```
long j=i;
```

long to Int:

```
long l=500;
```

```
int i=(int)l;
```

Int to double:

```
int i=200;
```

```
double d=i;
```

double to Int:

```
double d=19.5;
```

```
int i=(int)d;
```

Char to Int:

```
char c='l';
```

```
int a=Character.getNumericValue(c);
```

Int to Char:

```
int a=65;
```

```
char c=(char)a;
```

String to boolean:

String s1 = "true";

String s2 = "ok";

boolean b1 = Boolean.parseBoolean(s1);

boolean b2 = Boolean.parseBoolean(s2);

boolean to String:

boolean b1 = true;

String s1 = String.valueOf(b1);

Date to Timestamp:

Date date = new Date();

TimeStamp ts = new Timestamp(date.getTime());

Timestamp to Date:

ts = new Timestamp(System.currentTimeMillis());

Timestamp ts = new Date(ts.getTime());

Date date = new Date(ts.getTime());

Binary to Decimal:

System.out.println(Integer.parseInt("1010", 2));

(Integer.toHexString(10));

Decimal to Binary:

(Integer.parseInt("a", 16));

Hex to Decimal:

(Integer.toHexString(289));

Decimal to Hex:

(Integer.parseInt("c21", 8));

Octal to Decimal:

(Integer.parseInt("123", 8));

* Methods in character class:

Methods

boolean isLetter(char ch)

return type
returns true if ch is
a alphabet, otherwise
return false

boolean isDigit (char ch)

returns true if ch is
a digit

boolean isWhitespace (char ch)

returns true if ch is
a white space

boolean isUpperCase (char ch)

specifies whether char is
uppercase or not

boolean isLowerCase (char ch)

specifies whether char is
lowercase or not

char toUpperCase (char ch)

returns uppercase from
specified char value

char toLowerCase (char ch)

returns lowercase from .. "

toString (char ch)

returns string object .. "

static int charCount (int code
Point)

determines no. of char values
needed (unicode code point)

char charValue ()

returns value of character
object ..

* Methods in character class:

Methods

boolean isLetter(char ch)

Return type

Returns true if ch is a alphabet, otherwise return false

boolean isDigit (char ch)

Returns true if ch is a digit

boolean isWhitespace (char ch)

Returns true if ch is a white space

boolean isUpperCase (char ch)

Specifies whether char is uppercase or not

boolean isLowerCase (char ch)

Specifies whether char is lowercase or not

char toUpperCase (char ch)

Returns uppercase from specified char value

char toLowerCase (char ch)

Returns lowercase from

toString (char ch)

Returns string object

static int charCount (int code Point)

Determines no. of char values needed (unicode code point)

char charValue ()

Returns value of character object.

Character class

Methods

Return type

charCount (int codePoint)

determines no. of char values

charValue ()

Returns value of character object

codePointAt (char[] a, int index)

Returns codePoint for specified index of given array

codePointBefore (char[] a, int index)

Returns codePoint for given array in preceding index

codePointCount (char[] a, int offset, int count)

Returns the total number of unicodes points in given subarray

codePointOf (String name)

Returns value of codePoint for given unicode character.

compare (char x, char y)

compares two characters type values numerically

compare (Character another character)

" " objects numerically

digit (char ch, int radix)

Returns numeric value of given character in specified index

digit (int codepoint, int radix)

" " " in specified radix.

String Class.

Method	Description	Return type
1. charAt (int index)	returns character at specified index	char
2. int length ()	returns length of string	int
3. String format (String format)	formats string	String
4. substring	gives a part of string with specified start and end index	String
5. boolean contains	returns true or false if string has subsequence	boolean
6. join	checks the equality of two strings	String
7. boolean equals (Object obj)	returns true or false if string is equal to object	boolean
8. is empty ()	checks if string is empty	boolean
9. concat (String str)	concatenates the specified string	String
10. String replace (char old, char new)	replaces all occurrences of the specified char values	String
11. String replace (CharSequence old, CharSequence new)	replaces all occurrences of the specified char sequences	String

12. `equalIgnoreCase (String other)` Replaces all occurrences of the specified char value

13. `split (String regex)` returns a split string matching regex

14. `String [] split (String regex, int limit)` returns a string matching regex and limit.

15. `intern ()` returns the specified char value index with given index.

16. `index of (String substring, int fromIndex)` returns the specified substring index starting with given index.

17. `toLowerCase ()` returns a string in lowercase.

18. `toUpperCase ()` returns a string in uppercase.

20. `trim()` removes beginning and end Space of this string.

21. `valueOf (int value)` converts given type into String. It is an overloaded method.

String Buffer class Methods

Method

Modifier and type

1) `append (String s)` public synchronized String
Buffer

2) `insert (int offset,
String s)` public synchronized String
Buffer

3) `replace (int start
index ,int end index,
String str)` public synchronized String
Buffer

4) `reverse ()` public synchronized String
Buffer

5) `capacity ()` public int

6) `ensureCapacity ()` public void

7) `charAt ()` public char

8. public string
public substring (int begin index)
will go to same bag
private
public string

9. substring (int begin index, int end index)
public string

substring ends return value

→ 10. two solutions

begin

part1 beginning sibling

(2 part) begin (1

sibling

part1 beginning sibling

return (1) + (2)

return

part1 beginning sibling

return (1) + (2)

return

part1 beginning sibling

return (1) + (2)