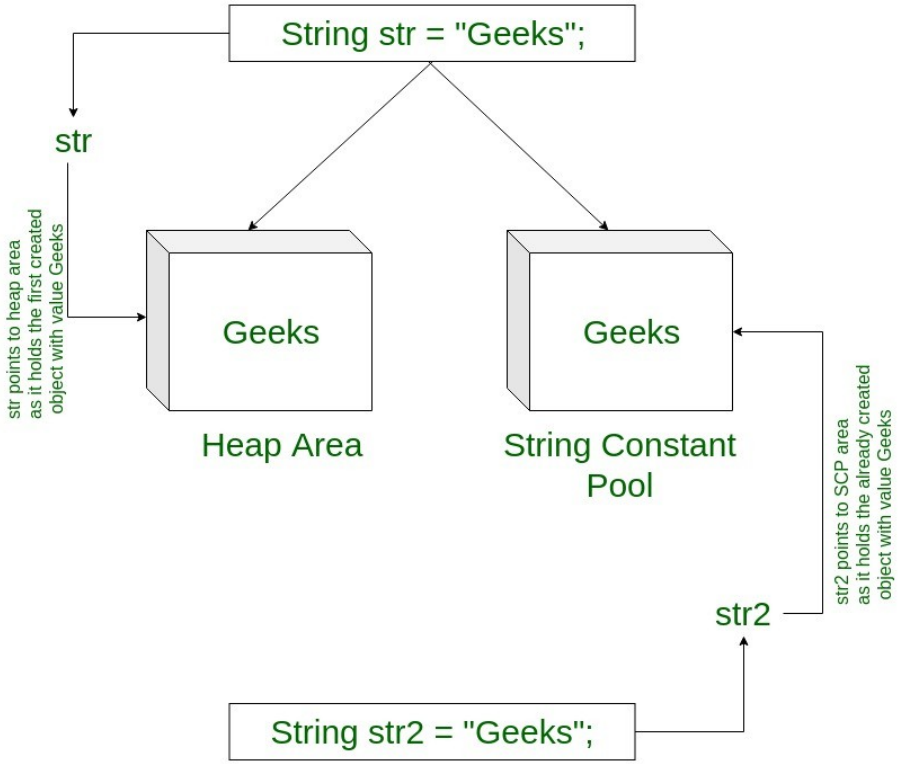
Strings In Java

Strings

- Strings in Java are Objects that are backed internally by a char array
- Strings are immutable
- Changing a String creates an entirely new String
- Uses String Constant Pool for storing strings
- Pointers to SCP is saved in Heap



String examples

// Single object with two references

String s1="Welcome";

String s2="Welcome";

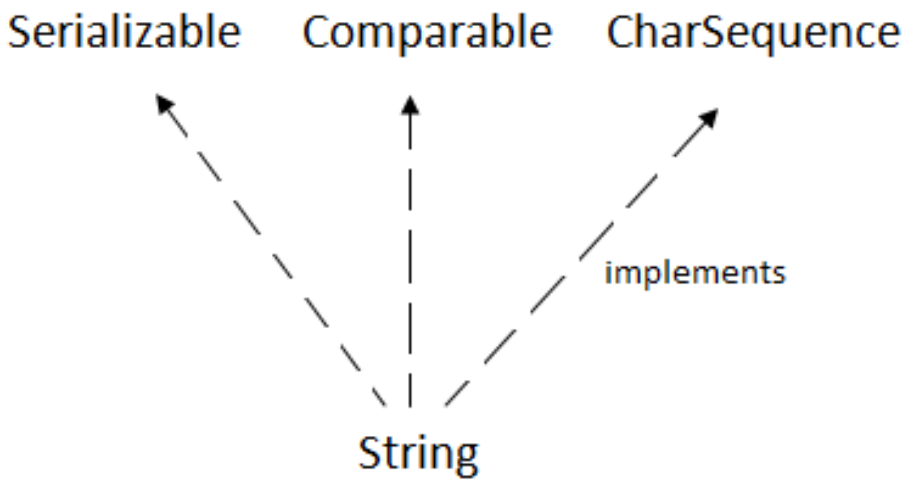
//creates two objects and one reference variable

String s=new String("Welcome");

//converting char array to string

char ch[]={'s','t','r','i','n','g','s'};

String s2=new String(ch);



Some String Methods

No.	Method	Description
1	char charAt(int index)	returns char value for the particular index
2	int length()	returns string length
3	static String format(String format, Object... args)	returns a formatted string.
4	String substring(int beginIndex, int endIndex)	returns substring for given begin index and end index.
5	boolean contains(CharSequence s)	returns true or false after matching the sequence of char value.
6	boolean equals(Object another)	checks the equality of string with the given object.
7	boolean isEmpty()	checks if string is empty.
8	String concat(String str)	concatenates the specified string.
9	String replace(char old, char new)	replaces all occurrences of the specified char value.
10	String replace(CharSequence old, CharSequence new)	replaces all occurrences of the specified CharSequence.
11	static String equalsIgnoreCase(String another)	compares another string. It doesn't check case.
12	String[] split(String regex)	returns a split string matching regex.
13	int indexOf(int ch)	returns the specified char value index.
14	String toLowerCase()	returns a string in lowercase.
15	String toUpperCase()	returns a string in uppercase.
16	String trim()	removes beginning and ending spaces of this string.

For-Loops

For	For-Each

for (int i = 0; i < 5; i++) {
  System.out.println(i);
}
	String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
	
	for (String car : cars) {
	  System.out.println(car);
	}

Break-Continue

Break	Continue

for (int i = 0; i < 10; i++) {
  if (i == 4) {
  break;
  }
  System.out.println(i);
}
	for (int i = 0; i < 10; i++) {
	  if (i == 4) {
	  continue;
	  }
	  System.out.println(i);
	}

int i = 0;
while (i < 10) {
  System.out.println(i);
  i++;
  if (i == 4) {
  break;
  }
}
	int i = 0;
	while (i < 10) {
	  if (i == 4) {
	  i++;
	  continue;
	  }
	  System.out.println(i);
	  i++;
	}

Swithcing

```
int day = 6;
switch (day) {
  case 6:
    System.out.println("Today is Saturday");
    break;
  case 7:
    System.out.println("Today is Sunday");
    break;
  default:
    System.out.println("Looking forward to the Weekend");
}
```

Management Phrases

{"Multi-Tier", "B-to-B" , "Win-win" , "Front-end" , "Web-based" , "Pervasive", "Smart", "Six-sigma", "Critical-path", "Dynamic"}

{"empowered", "sticky","value-added", "oriented", "centric", "distributed", "clustered", "branded", "outside-the-box", ''positioned'', "networked", "focused", "leveraged", "aligned", "targeted", "shared", "cooperative", "accelerated"}

{"process", "tipping-point", "solution", "architecture", "core competency", "strategy", "mindshare", "portal" , "space", "vision", 'paradigm", "mission"}

Swithcing

```
int day = 6;
switch (day) {
  case 6:
    System.out.println("Today is Saturday");
    break;
  case 7:
    System.out.println("Today is Sunday");
    break;
  default:
    System.out.println("Looking forward to the Weekend");
}
```