***String in Java***

String is the class that represents sequence of character.

Package is Java. Lang.

String class implements Serializable, comparable, char sequence interface.

String is the immutable, once we create a String object we can't perform any changes in the existing object. If we are trying to perform any changes with those changes a new object will be created. This behavior is called immutability of the String object.

**public** **class** StringDemo {

**public** **static** **void** main(String[] args) {

String s = "rahul";

s.concat("pawar");// concat() method appends the string at the end

System.***out***.println(s);// will print rahul because strings are immutable objects

}

}

For mutable class, you can use String buffer and String builder class.

Once we created a StringBuffer object we can perform any changes in the existing object. This behavior is called mutability of the StringBuffer object.

How to create the string object?

1. String literal
2. By new keyword
3. String literal-

It is created by using double quotes.

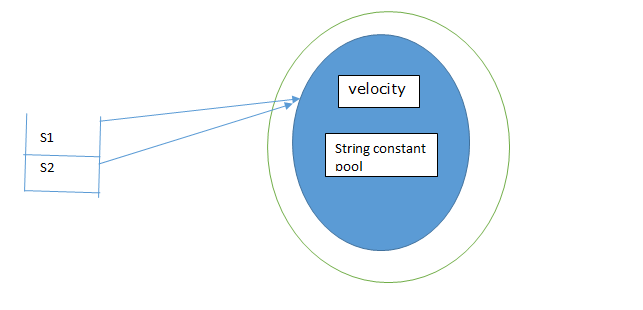
Example- String s=” velocity”;

Each time when you create string literal, the JVM check string constant pool first, if the string object is already present in the pool, reference to pooled instance is returned. If string does not present in the pool, new string instance is created and placed in pool.

Example- String s1=”velocity”;

String s2=” velocity”; //will not create the new instance.

In the above example, only one object will be created, firstly JVM will not find any string object with value “velocity” in the string constant pool, so it will create new object. After that it will find string with value= “velocity” in pool, it will not create the new object but will return reference to same instance.



Why Java uses the concept of string literal?

To make the java more memory efficient (because no new object is created if it exists already in string constant pool.)

1. By new Keyword-

Example- String s= new String(“pune”);

/\* create two objects \*/

In such case, JVM will create the new String object in normal(non-pool) heap memory and literal “pune” will be placed in string constant pool. The variable s refers to object in heap(non-pool).

Example- 1

**package** com.velocity;

**public** **class** StringDemo {

**public** **static** **void** main(String[] args) {

String s1 = "velocity";

String s2 = **new** String("velocity");

System.***out***.println(s1 == s2);

System.***out***.println(s1.equals(s2));

}

}

Output

false

true

Example- 2

**package** com.velocity;

**public** **class** StringDemo {

**public** **static** **void** main(String[] args) {

String s1 = "velocity";

String s2 = **new** String("pune");

s2=s1;

System.***out***.println(s1 == s2);

System.***out***.println(s1.equals(s2));

}

}

Output-

true

true

Example-3

**package** com.velocity;

**public** **class** StringDemo {

**public** **static** **void** main(String[] args) {

String s1 = **new** String("velocity");

String s2 = **new** String("pune");

s2 = s1;

System.***out***.println(s1 == s2);

System.***out***.println(s1.equals(s2));

}

}

Output-

true

true

Example- 4

**package** com.velocity;

**public** **class** StringDemo {

**public** **static** **void** main(String[] args) {

String s1 = "velocity";

String s2 = **new** String("pune");

System.***out***.println(s1 == s2);

System.***out***.println(s1.equals(s2));

System.***out***.println(s1.hashCode());

System.***out***.println(s2.hashCode());

}

}

Output-

false

false

2134260957

3452540

Example-5

**package** com.velocity;

**public** **class** StringDemo {

**public** **static** **void** main(String[] args) {

String s1 = **new** String("velocity");

String s2 = **new** String("pune");

System.***out***.println(s1 == s2);

System.***out***.println(s1.equals(s2));

}

}

Output-

false

false

**Important methods of string class :**

1. public char charAt(int index);

Returns the character locating at specified index.

1. public String concat(String str);
2. public boolean equals(Object o);

Used for case sensitive comparision

1. public boolean equalsIgnoreCase(String s);

For content comparison where case is not important.

1. public String substring(int begin);

Return the substring from begin index to end of the string.

1. public String substring(int begin, int end);

Returns the substring from begin index to end-1 index.

1. public int length();

Returns the number of characters present in the string.

1. public String replace(char old, char new);

To replace every old character with a new character.

1. public String toLowerCase();

Converts the all characters of the string to lowercase.

1. public String toUpperCase();

Converts the all characters of the string to uppercase.

1. public String trim();

We can use this method to remove blank spaces present at beginning and end of the string but not blank spaces present at middle of the String.

1. public int indexOf(char ch);

It returns index of 1st occurrence of the specified character if the specified character is not available then return -1.

1. public int lastIndexOf(Char ch);

It returns index of last occurrence of the specified character if the specified character is not available then return -1.

Design the program to display velocity message on screen

**package** com.sample;

**public** **class** SampleTest {

**public** **static** **void** main(String[] args) {

String str = "velocity";

System.***out***.println("Institute name is>>" + str);

}

}

Design the program to perform the string operation

**package** com.sample;

**public** **class** SampleTest {

**public** **static** **void** main(String[] args) {

String str = "velocity";

System.***out***.println(str.length());

System.***out***.println(str.charAt(4));

System.***out***.println(str.compareTo("velocity"));

System.***out***.println(str.concat("pune"));

System.***out***.println(str.hashCode());

System.***out***.println(str.toLowerCase());

System.***out***.println(str.toUpperCase());

}

}

Design the program to counting space into string.

**package** com.sample;

**public** **class** SampleTest {

**public** **static** **void** main(String[] args) {

String str = "velocity training center pune";

**int** counter = 0;

**for** (**int** i = 0; i < str.length(); i++) {

**char** ch = str.charAt(i);

**if** (ch == ' ') {

counter++;

}

}

System.***out***.println("total space in string are>>" + counter);

}

}