**String in Java**

|  |  |
| --- | --- |
| String, StringBuffer, StringBuilder | String : Thread safe, immutable  StringBuffer: Mutable , Thread safe  StringBuilder : Mutable, Not thread safe.  1. StringBuilder is not thread-safe. StringBuffer is thread-safe.  2. StringBuilder is not synchronized and StringBuffer is synchronized.  3. StringBuilder is faster while StringBuffer is slower as it is thread-safe. |
| Why String is immutable in JAVA | 1. String pool is possible only because String is immutable in Java 2. If String is not immutable then it would cause a severe security threat to the application. 3. Since String is immutable, it is safe for multithreading. 4. Since String is **immutable**, its **hashcode** is cached at the time of creation and it doesn’t need to be calculated again. This is why String is the most widely used as HashMap keys. 5. String is mostly used as a key in HashMap class because it implements **equals() and hashCode()** methods which is required for an Object to be used as key in HashMap. |
| How many objects are created using new keyword?  String str = new String("JavaHungry"); | 2, Two objects are created by the above statement. One object in the heap memory and one object in the String constant pool. |
| convert String to char Array? | toCharArray() |
| How many different ways you can create a String object? | String str = "javahungry"; // String literal  String str = new String("javahungry"); // using new operator |
| Are String thread-safe in java? | String objects are immutable. It means they are thread-safe also. |
| Can we use String in switch statement? | Yes, you can use String in switch statement in java 7. Prior to java 7 , you had to use if-else statements to achieve the task. |
| Is String primitive type or object (derived) type in java? | String is object(derived) type in java. |
| What is String intern() method? | When the intern method is invoked, if the String constant pool already contains a string equal to the String object as determined by the equals(Object) method, then the string from the pool is returned.  Otherwise the String object is added to the pool and a reference to the String object is returned. |
| How will you create an immutable class in java? | You can create immutable class in java by implementing below points:  1. Make the class final so it can not be extended(inherited)  2. Make all fields private so one can not access them from outside the class.  3. Do not provide setter methods for the variables.  4. Declare all mutable fields as final so that it's value can be assigned only once. |
| How will you create mutable String objects in java? | As we have discussed, by using StringBuffer and StringBuilder objects. |
| == operator compares the reference  equals() = Value comparison | String str1 = "abc";  String str2 = new String("abc");  System.out.println(str1 == str2); //false  System.out.println(str1.equals(str2)); //true |
|  |  |
| Explain the difference between str1.equals("abc") and "abc".equals(str1), where str1 is any String object? | If str1 value is "abc" then both statements will give the result true. Main difference between the two statement arises when we pass str1 value as NULL. If the str1 is null then first statement will throw null pointer exception while second statement will return false. |
| What is String Constant Pool? | String constant pool increases the reusability of the existing String objects.  It also saves memory as no two objects with same content are created.  String constant pool is the memory space allocated in the heap memory to store the objects which are  created using String literals. String constant pool is unique, there are no two String o objects which has the same value(content). |
| There are lot of String concatenation and String modification operations in my code. Which class should I use among String,StringBuffer and StringBuilder? Given I also want thread-safe code? | This is scenario based question. You should give answer StringBuffer.  You can use String also but with every modification and concatenation operation, a new String is formed as String is immutable. It will lead to the memory allocation issues.  StringBuilder can not be used as it is not synchronized, i.e thread-safe.  So, the clear answer is StringBuffer. |
| Why char Array is preferred over String in storing passwords? | One of the main reason to prefer char Array over String is security risk of stealing passwords. Since String are reusable in the constant pool , there are high chances that they remain in the memory for the long duration. Anyone who has access to the memory dump can find the password in clear text.  That's why password should be encrypted. |
| Find the length of the String without using length() method? | static int length(String s)  {  try  {  for(i=0,c=0;0<=i;i++,c++)  s.charAt(i);  }  catch(Exception e)  //Array index out of bounds and array index out of range are different exceptions  {  System.out.print("length is ");  // we can not put return statement in catch  }  return c;  } |
| String to int  Integer to String | int i = Integer.valueOf("90");  String str = String.valueOf(90); |
| Optimize java string creation | We can optimize by using String literal.. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Java String Reverse |  |
| toString uses |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |