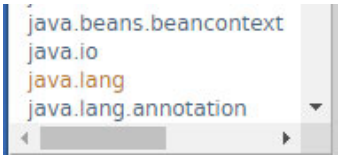
02.02 Virtual Lecture Notes

The Big Picture

Most of the pre-written code you will use is contained in three packages of the Java API.

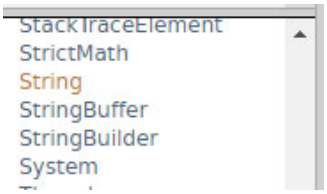
- **java.lang**: Provides classes that are fundamental to the design of the Java programming language.
- **java.util**: Contains the collections framework, legacy collection classes, event model, date and time facilities, internationalization, and miscellaneous utility classes.
- **java.io**: Provides for system input and output through data streams, serialization and the file system.

Let's look at one of its most useful classes of the java.lang package: the **String** class.



Scroll down in the packages frame and select the java.lang option.

The bottom left frame will list all of the classes, interfaces, and exceptions in the selected package (java.lang).

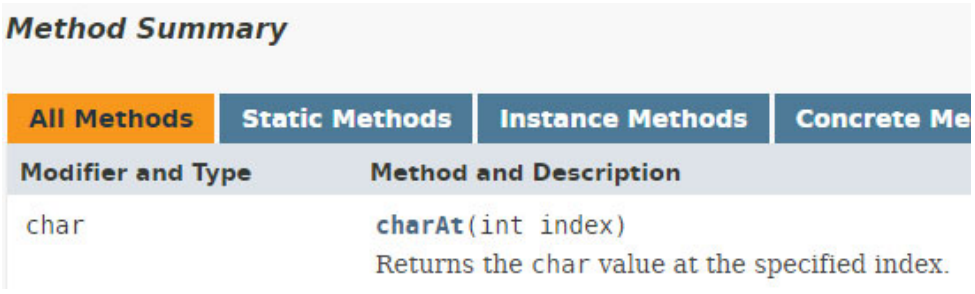


Scroll down in the bottom left frame and select the `String` class

The main window should change and provide *detailed* documentation about the `String` class. As you scroll down the main window, notice all of the `String` methods available to you and that the information is split into two sections: **Summary** and **Detail**.

The Method Summary

Scroll down in the main window until you reach the **Method Summary** table where each `String` class method is briefly described, one method per row. Find the `charAt()` method as shown below.



In the Method Summary view there will always be four pieces of information to notice, three on the right side and one on the left.

1. Every method has a name (e.g. `charAt`). It will be a link to the details for that method.
2. The summary under the name indicates that the `charAt()` method returns the char value at the specified index position.
3. The information in parentheses `(int index)` indicates how many parameters the method takes and the parameter's type. The `charAt()` method takes one int parameter which is named index. (For now think of a parameter as the information a method needs to perform its task.)
4. The word `char` on the left side of the vertical line indicates that when the `charAt()` method is called, it will return a single character to the statement that called it.

The Method Summary in every class is organized exactly the same way. It is designed to give you an overview of how to use a method, but if you need additional information, you must consult the Method Detail section of the API.

Part 4: Method Detail

In the Method Summary, click on the `charAt()` link, and you will jump down to the Method Detail section as shown below.

charAt

```
public char charAt(int index)
```

Returns the char value at the specified index. An index ranges from 0 to length() - 1. The first char value of the sequence is at index 0, the next at index 1, and so on, as for array indexing.

If the char value specified by the index is a surrogate, the surrogate value is returned.

Specified by:

charAt in interface CharSequence

Parameters:

index - the index of the char value.

Returns:

the char value at the specified index of this string. The first char value is at index 0.

Throws:

IndexOutOfBoundsException - if the index argument is negative or not less than the length of this string.

The information supplied in the Method Detail table should be sufficient to use a method in a program.

Each character within a string has an index position. Something interesting to note is that 0, not 1, is the index position of the first character in a `String` literal. The index positions of several letters in the following String literal are indicated below.

```
String phrase = "Hello, Virtual World!"
                 ^      ^       ^
                 0      7       15
```

The following code segment will print the character at the 0 index position in the phrase shown above.

```
int index = 0;
String phrase = "Hello, Virtual World!";
char symbol = phrase.charAt(index);
System.out.print("The character: " + symbol);
System.out.println(", is in index position " + index);
```

What do you predict would be printed for each of the following index positions in the String literal shown above?

0	8	6	10	0	1	9	1	20

If you got the message "Hi tHere!" you have a good understanding of index positions and how the charAt() method works.

Print