Lesson 56: `String` Methods and Properties #2 (W19D2)

Balboa High School

Michael Ferraro

January 5, 2016

Do Now

In a new project, `Lesson56`, create class `OddCharacters`.

1. Write method `oddChars()` that takes a `String` as a parameter and returns a `String` having only the "odd" characters (i.e., the first, third, fifth, ... characters).

   For example:

   ```
   String odds = oddChars("AaBbCcDd");
   // odds = "ABCD"
   ```

2. Have `main()` call `oddChars()` with the `Strings` below, printing the results.
   1. `snack`
   2. `sparkles`

Aim

Students will continue learning about the properties of `Strings` in Java and work on related programming problems.

String Methods: substring()

- substring() returns part of a String

`String` Methods: `substring()`

- `substring()` returns part of a `String`
- `substring()` is an *overloaded* method

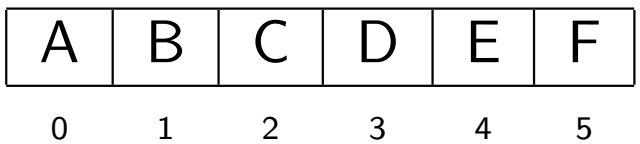
String Methods: substring()

- substring() returns part of a String
- substring() is an *overloaded* method
  - substring(fromPos): returns all characters starting from the specified position

String Methods: substring()

- substring() returns part of a String
- substring() is an *overloaded* method
  - substring(fromPos): returns all characters starting from the specified position

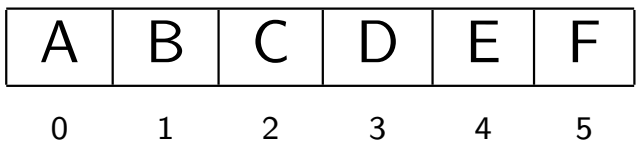
Ex:



String Methods: substring()

- substring() returns part of a String
- substring() is an *overloaded* method
  - substring(fromPos): returns all characters starting from the specified position

Ex:

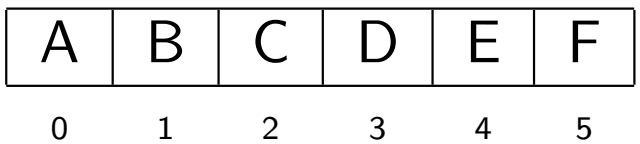



```
System.out.println( "ABCDEF".substring(3) );
```

String Methods: substring()

- substring() returns part of a String
- substring() is an *overloaded* method
  - substring(fromPos): returns all characters starting from the specified position

Ex:

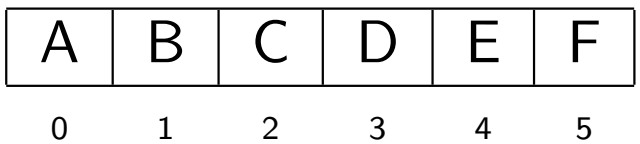



```
System.out.println( "ABCDEF".substring(3) ); ← DEF
```

String Methods: substring()

- substring() returns part of a String
- substring() is an *overloaded* method
  - substring(fromPos): returns all characters starting from the specified position

Ex:

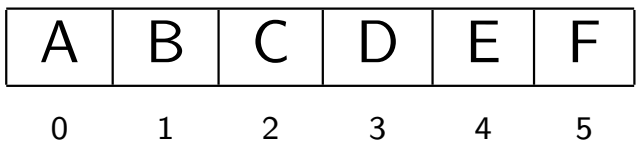



```
System.out.println( "ABCDEF".substring(3) ); ← DEF

System.out.println( "ABCDEF".substring(0) );
```

String Methods: substring()

- substring() returns part of a String
- substring() is an *overloaded* method
  - substring(fromPos): returns all characters starting from the specified position

Ex:




```
System.out.println( "ABCDEF".substring(3) ); ← DEF
System.out.println( "ABCDEF".substring(0) ); ← ABCDEF
```

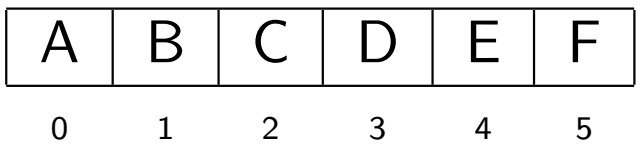
String Methods: `substring()`

- `substring()` returns part of a `String`
- `substring()` is an *overloaded* method
  - `substring(fromPos, toPos)`: returns characters starting from `fromPos` and ending **before** `toPos`

String Methods: `substring()`

- `substring()` returns part of a `String`
- `substring()` is an *overloaded* method
  - `substring(fromPos, toPos)`: returns characters starting from `fromPos` and ending **before** `toPos`

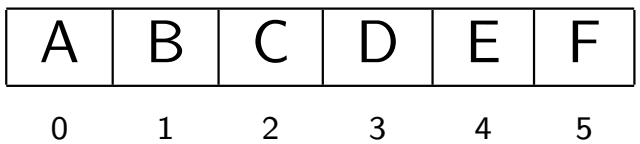
Ex:



String Methods: substring()

- substring() returns part of a String
- substring() is an *overloaded* method
  - substring(fromPos, toPos): returns characters starting from fromPos and ending **before** toPos

Ex:

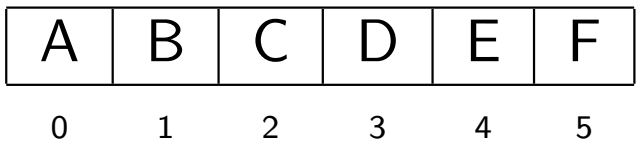



```
System.out.println( "ABCDEF".substring(1,4) );
```

String Methods: substring()

- substring() returns part of a String
- substring() is an *overloaded* method
  - substring(fromPos, toPos): returns characters starting from fromPos and ending **before** toPos

Ex:

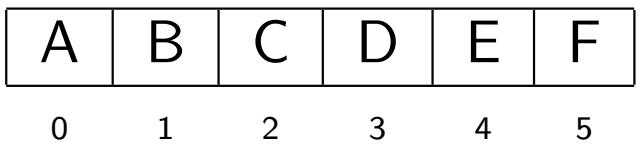



```
System.out.println( "ABCDEF".substring(1,4) ); ← BCD
```

String Methods: substring()

- substring() returns part of a String
- substring() is an *overloaded* method
  - substring(fromPos, toPos): returns characters starting from fromPos and ending **before** toPos

Ex:

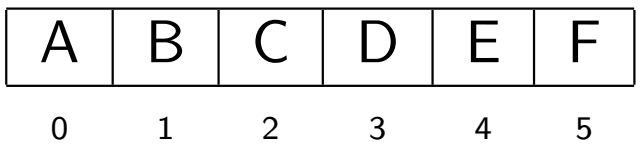



```
System.out.println( "ABCDEF".substring(1,4) ); ← BCD

System.out.println( "ABCDEF".substring(4,5) );
```

String Methods: substring()

- substring() returns part of a String
- substring() is an *overloaded* method
  - substring(fromPos, toPos): returns characters starting from fromPos and ending **before** toPos

Ex:

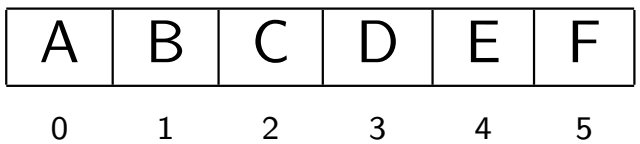



```
System.out.println( "ABCDEF".substring(1,4) ); ← BCD
System.out.println( "ABCDEF".substring(4,5) ); ← E
```

String Methods: substring()

- substring() returns part of a String
- substring() is an *overloaded* method
  - substring(fromPos, toPos): returns characters starting from fromPos and ending **before** toPos

Ex:

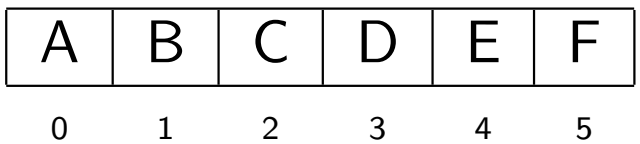



```
System.out.println( "ABCDEF".substring(1,4) ); ← BCD

System.out.println( "ABCDEF".substring(4,5) ); ← E
System.out.println( "ABCDEF".charAt(4)      );
```

String Methods: substring()

- substring() returns part of a String
- substring() is an *overloaded* method
  - substring(fromPos, toPos): returns characters starting from fromPos and ending **before** toPos

Ex:



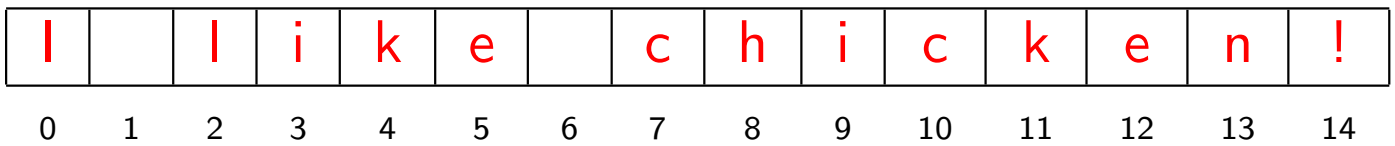

```
System.out.println( "ABCDEF".substring(1,4) ); ← BCD

System.out.println( "ABCDEF".substring(4,5) ); ← E
System.out.println( "ABCDEF".charAt(4)      ); ← E
```

String Methods: substring()

Build a class to test the following on `fun`.

```
String fun = "I like chicken!"
```

- `substring( 0 );`
- `substring( 2 );`
- `substring( 6, 11 );`
- `substring( 6, 11 ).trim();`
- `substring( 10, 11 );`
- `substring( 10, 10 );`
- `substring( 5, fun.length() );`



PS #10: §4, Book Questions #1

Once you have an understanding of how the methods work for class `String` — including ones we didn't cover in class, like `indexOf()` — begin working on §4 of PS #10.

- You can find the referenced problems here.
- Getting stuck? There's a video that gives you strategies you can use on these problems! Take out headphones and see here[1]. Make sure you're in full-screen mode at 720p (HD) quality.

[1] The video refers to Litvin Ch. 10, which corresponds to the older edition of the textbook. The problems are now in Ch. 8.

HW

Complete PS #10, §§1-4, inclusive.