CS 213 : Software Methodology

Spring 2017

Sesh Venugopal

Lecture 4: Jan 26

Design Aspects of Static Members

Static/Non-Static Mix: Another Example

- Parsing a string into an integer, e.g. “123” -> 123 – where to provide this functionality?

OPTIONS:

- Have a `String` instance method, say, `parseAsInteger` that returns an `int`, e.g.

  ```
  int i = “123”.parseAsInteger();
  ```

  Bad design: Parsing an int is not an inherent/characteristic property of a String – not all strings can be parsed as integers.

- Have a `String` static method, say, `parseAsInteger` that returns an `int`, e.g.

  ```
  int i = String.parseAsInteger(“123”);
  ```

- Have an `Integer` static method, say, `parseInt` that returns an `int`, e.g.

  ```
  int i = Integer.parseInt(“123”);
  ```

- Of the second and third choices, which one is better? Why? `Integer.parseInt` is better

  Think of converting strings to doubles, floats also –
  having all these types of conversions in `String` would require `String` to know about formats of other types, which is NOT its business.
  Best to localize functionality in the class(es) corresponding to the converted type.