Methods

Boxing and Unboxing

Boxing and Unboxing

- **Boxing**: converting a primitive into its wrapper class
  - (putting primitive in the "box")

- **Unboxing**: converting a wrapper class to a primitive
  - (getting a primitive out of the "box")

```
// explicit
int a = 3;
Integer b = Integer.valueOf(a);
  // int -> Integer   (boxing)
int c = b.intValue();
  // Integer -> int   (unboxing)


// implicit
int a = 3;
Integer b = a;
  // int -> Integer   (autoboxing)
int c = b;
  // Integer -> int   (unboxing)
// Java will also autocast a smaller primitive into the larger one
// BUT Java will not do both automatic operations at the same time!!


int x = 7;
long y = x;
  // autocasting, OK
Long z = x;
  // autocasting and autoboxing cannot be done at once => NOK!
```

```java
// if you need both autocasting and autoboxing,
// one of these operations should be done by hand (or both)
int x = 7;

// explicit boxing (w/ autocasting)
Long z = Long.valueOf(x);

// explicit casting (w/ autoboxing)
Long z = (long) x;

// explicit everything
Long z = Long.valueOf((long)x);
```

// be careful when working with primitive literals

Long x = $\boxed{10}_{\text{int}}$;

=> NOK, autocasting and autoboxing is required

Long y = $\boxed{10L}_{\text{long}}$;

=> OK, only autoboxing is required