

Homework: Tracing With References

Answer the following questions. For the tracing exercises, you should be able to complete the tracing tables on this page and then print the page from the browser.

- Consult the contract for Standard interface's method `transferFrom` specified [here](#) and then complete the tracing table from [slides 78-79](#) from Meeting 23 reproduced here where `m` and `k` are variables of type `NaturalNumber`.

Statement	Variable Values
	<code>m</code> → 143 <code>k</code> → 70
<code>m.transferFrom(k);</code>	
	<code>m</code> → <input type="text" value="70"/> <code>k</code> → <input type="text" value="cleared"/>

- Explain why an *immutable* type **cannot** have a `transferFrom` method as specified in the Standard interface.
- Complete the following tracing tables. Each of them has a short method declaration followed by short client code that invokes the method. Carefully complete each tracing table starting from the client code and tracing over the method call and through the method body.

- Swapping for primitive types:

Statement	Variable Values
<code>private static void swap1(int i1, int i2) {</code>	
	<code>i1</code> = <input type="text" value="7"/> <code>i2</code> = <input type="text" value="12"/>
<code>int tmp = i1;</code>	
	<code>i1</code> = <input type="text" value="7"/> <code>i2</code> = <input type="text" value="12"/> <code>tmp</code> = <input type="text" value="7"/>
<code>i1 = i2;</code>	
	<code>i1</code> = <input type="text" value="12"/> <code>i2</code> = <input type="text" value="12"/> <code>tmp</code> = <input type="text" value="7"/>
<code>i2 = tmp;</code>	
	<code>i1</code> = <input type="text" value="12"/> <code>i2</code> = <input type="text" value="7"/> <code>tmp</code> = <input type="text" value="7"/>
<code>}</code>	
Start tracing here	
<code>int x = 7, y = 12;</code>	
	<code>x</code> = <input type="text" value="7"/> <code>y</code> = <input type="text" value="12"/>
<code>swap1(x, y);</code>	

	x = <input type="text" value="12"/> y = <input type="text" value="7"/>
--	---

2. Swapping for (immutable) reference types:

Statement	Variable Values
<code>private static void swap2(String s1, String s2) {</code>	
	s1 → <input type="text" value="legends"/> s2 → <input type="text" value="leaders"/>
<code>String tmp = s1;</code>	
	s1, tmp → <input type="text" value="legends"/> s2 → <input type="text" value="leaders"/>
<code>s1 = s2;</code>	
	s1, s2 → <input type="text" value="leaders"/> tmp → <input type="text" value="legends"/>
<code>s2 = tmp;</code>	
	s1 → <input type="text" value="leaders"/> s2, tmp → <input type="text" value="legends"/>
<code>}</code>	
<i>Start tracing here</i>	
<code>String x = "legends", y = "leaders";</code>	
	x → <input type="text" value="legends"/> y → <input type="text" value="leaders"/>
<code>swap2(x, y);</code>	
	x → <input type="text" value="leaders"/> y → <input type="text" value="legends"/>

3. Swapping for (mutable) reference types:

Statement	Variable Values
<code>private static void swap3(NaturalNumber n1, NaturalNumber n2) {</code>	
	n1 → <input type="text" value="41"/> n2 → <input type="text" value="78"/>
<code>NaturalNumber tmp = n1;</code>	
	n1, tmp → <input type="text" value="41"/> n2 → <input type="text" value="78"/>
<code>n1 = n2;</code>	
	n1, n2 → <input type="text" value="78"/> tmp → <input type="text"/>

	<input type="text" value="41"/>
n2 = tmp;	
	n1 → <input type="text" value="41"/> n2, tmp → <input type="text" value="41"/>
}	
<i>Start tracing here</i>	
NaturalNumber x = new NaturalNumber2(41), y = new NaturalNumber2(78);	
	x → <input type="text" value="41"/> y → <input type="text" value="78"/>
swap3(x, y);	
	x → <input type="text" value="41"/> y → <input type="text" value="41"/>