

# Exercises: Generic Types and printf

1. Make a static method called “lastEntry” to which you pass a List and get back the last entry of that list. If you pass it an List of Strings, you should get back a String. If you pass it a List of Circles, you should get back a Circle. E.g.:

```
List<Circle> listOfCircles = ...;
Circle lastCircle = ElementUtils.lastEntry(listOfCircles);
List<String> wordList = ...;
String lastWord = ElementUtils.lastEntry(wordList);
```

It is probably easiest to test this using Lists of String and Lists of Integer so that you don't have to copy in any other classes, but be sure that your method does not do anything specific to String or Integer.

2. Next, make a second version of the method that support arrays. That is, you should be able to call ListUtils.lastEntry(someList) or ListUtils.lastEntry(someArray).

```
List<String> wordList = ...;
String lastWord1 = ElementUtils.lastEntry(wordList);
String[] wordArray = ...;
String lastWord2 = ElementUtils.lastEntry(wordArray);
```

3. Make a class called TwoItems that stores two entries of a given type and has getter methods to retrieve them. Give it a useful toString method. Here are two examples of its use:

```
TwoItems<String> twoNames = new TwoItems<>("John", "Jane");
System.out.printf("twoNames=%s.%n", twoNames);
String name1 = twoNames.getItem1(); // Value is "John"
String name2 = twoNames.getItem2(); // Value is "Jane"
System.out.printf("  First item: %s.%n", name1);
System.out.printf("  Second item: %s.%n", name2);

TwoItems<Integer> twoNums = new TwoItems<>(5, 7);
System.out.printf("twoNums=%s.%n", twoNums);
Integer num1 = twoNums.getItem1(); // Value is 4
Integer num2 = twoNums.getItem2(); // Value is 7
System.out.printf("  First item: %s.%n", num1);
System.out.printf("  Second item: %s.%n", num2);
```

Hint: once you know how to define a class that accepts generic types, this is surprisingly simple, and amazingly similar to earlier classes you wrote. Remember that Eclipse will help you write the constructor, the getter methods, and the toString method.

4. Assume that you have four variables, name1, balance1, name2, and balance2. Produce output that shows the values, but line up the balances on the decimal point. Assume that the names are 7 characters or fewer and the balances are less than \$10M. For example:

```
Juanita's bank account balance is $2,345,678.99.
Juan's bank account balance is $ 15,455.26.
```