    - practice invoking static and instance methods
- Subsequent lesson on defining a new Java class
    - Delay introduction of constructors and methods
    - Initial emphasize on object state and object references
    - Use visual debuggers to clarify object concepts, avoid common misconceptions

Today's Lesson - Defining a new Java class

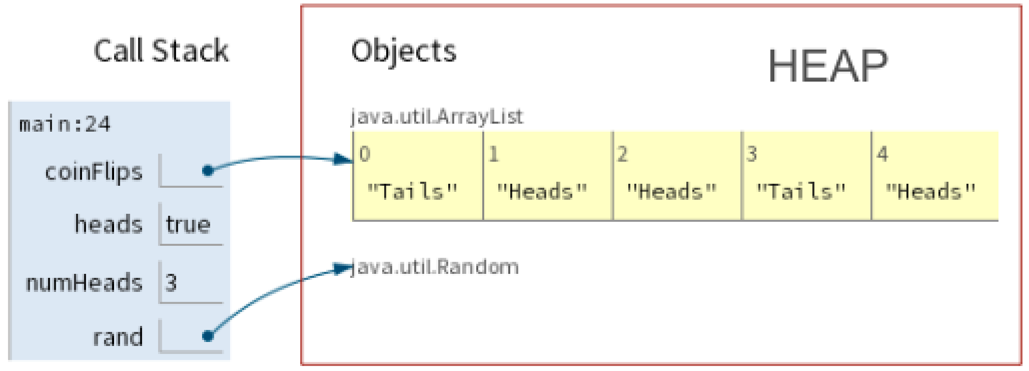
We've seen how to use existing Java core and utility classes (String, ArrayList, etc.) to solve some interesting problems.

Today we'll see how to define a **new** class to model some real world objects.

Review: What is an object?

Objects have state (properties/data) and behavior (operation that access/modify state)

```
    ArrayList<String> coinFlips = new ArrayList<String>();
    Random rand = new Random();
    int numHeads = 0;
    boolean heads = rand.nextBoolean();
    while (numHeads < 3) {
        if (heads) {
            numHeads++;
            coinFlips.add("Heads");
        }
        else {
            coinFlips.add("Tails");
        }
        heads = rand.nextBoolean();
    }
    System.out.println("Total coin flips:" + coinFlips.size());
    System.out.println(coinFlips);
}
```



CHALLENGE

Consider the following code:

```
public class Cat {

    String name;
    boolean isPurring;

    public static void main(String[] args) {
        Cat calico = new Cat();
        Cat tabby = new Cat();
        Cat favorite = calico;

        tabby.name = "Maru";
        calico.name= "Chestnut";
        favorite.isPurring = true;

        System.out.printf("calico: %s %b%n", calico.name, calico.isPurring);
        System.out.printf("tabby %s %b%n", tabby.name, tabby.isPurring);
        System.out.printf("favorite: %s %b%n", favorite.name, favorite.isPurring);
```