Static Variables and Methods

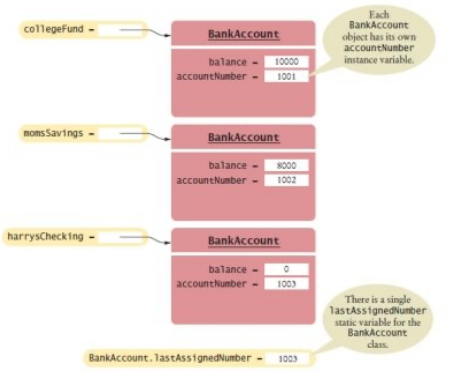

Figure 5 A Static Variable and Instance Variables

- An instance variable for the total is updated in methods that increase or decrease the total amount.
- A counter that counts events is incremented in methods that correspond to the events.
- An object can collect other objects in an array or array list.
- An object property can be accessed with a getter method and changed with a setter method.
- If your object can have one of several states that affect the behavior, supply an instance variable for the current state.
- To model a moving object, you need to store and update its position.
- When developing a solution to a complex problem, first solve a simpler task.
- Make a plan consisting of a series of tasks, each a simple extension of the previous one, and ending with the original problem.
- A class should represent a single concept from a problem domain, such as busi- ness, science, or mathematics.
- The public interface of a class is cohesive if all of its fea- tures are related to the concept that the class represents.
- A class depends on another class if its methods use that class in any way.
- An immutable class has no mutator methods.
- References to objects of an immutable class can be safely shared.
- A side effect of a method is any externally observable data modification.
- When designing methods, minimize side effects.
- In Java, a method can never change the contents of a variable that is passed to a method.
- In Java, a method can change the state of an object reference argument, but it cannot replace the object reference with anot

Modifying a Numeric Parameter Has No Effect on Caller

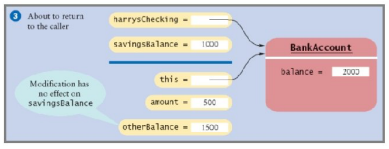

Figure 3(3):
Modifying a Numeric Parameter Has No Effect on Caller


```
public class CashRegister

{
```

```
public static final double QUARTER_VALUE = 0.25; public static final double DIME_VALUE = 0.1;

public static final double NICKEL_VALUE = 0.05;

.. .

public void receivePayment(int dollars, int quarters,

int dimes, int nickels, int pennies) .. .

}
```