Quality Tip 10.2

Clone Mutable Instance Variables in Accessor Methods

Consider the following class:

```
public class Customer
{
   private String name;
   private BankAccount account;

   public Customer(String aName)
   {
      name = aName;
      account = new BankAccount();
   }

   public String getName()
   {
      return name;
   }

   public BankAccount getAccount()
   {
      return account;
   }
}
```

This class looks very boring and normal, but the `getAccount` method has a curious property. It *breaks encapsulation*, because anyone can modify the object state without going through the public interface:

```
Customer harry = new Customer("Harry Handsome");
BankAccount account = harry.getAccount();
   // Anyone can withdraw money!
account.withdraw(100000);
```

Maybe that wasn't what the designers of the class had in mind? Maybe they wanted class users only to inspect the account? In such a situation, you should *clone* the object reference:

```
public BankAccount getAccount();
{
      return (BankAccount) account.clone();
   }
```

Do you also need to clone the `getName` method? No—that method returns a string, and strings are immutable. It is safe to give out a reference to an immutable object.