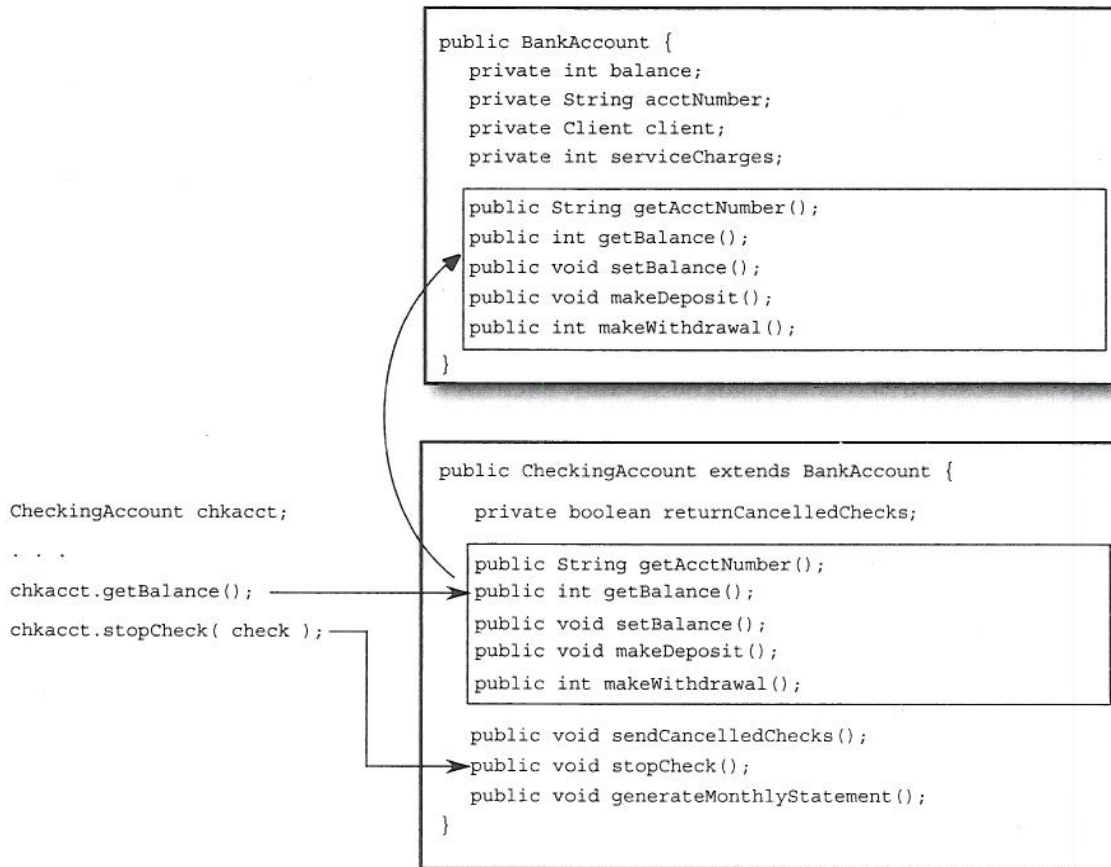


Figure 1.15 What CheckingAccount inherits from BankAccount.



method defined in `BankAccount` and inherited by `CheckingAccount`. From a client's perspective, it is as if `getBalance()` was defined in `CheckingAccount`. The second message is to a public method defined in `CheckingAccount`.

### What Have We Done So Far?

We have abstracted a set of characteristics (attributes, behaviors, and constraints) shared by all `BankAccounts` to define the general class `BankAccount`. For example, all `BankAccounts` have an account number, a client name and address, a balance, and so on, and have the shared behavioral characteristics that they can `getAccountNumber()`, `getBalance()`, `getClientName()`, and so on. Then, we identified subclasses that all share the characteristics of the superclass, but each subclass added characteristics that distinguished it from both its superclass and its "sibling" classes.

We say that moving from a superclass to a subclass defines a **specialization**. A `SavingsAccount` is a kind of `BankAccount` specialized for saving money that accrues interest and can be drawn on (a withdrawal operation) or added to (through interest and deposits). This distinguishes `SavingsAccounts` from the general class of `BankAccounts` as well as from its sibling class `CheckingAccount`.

Specialization