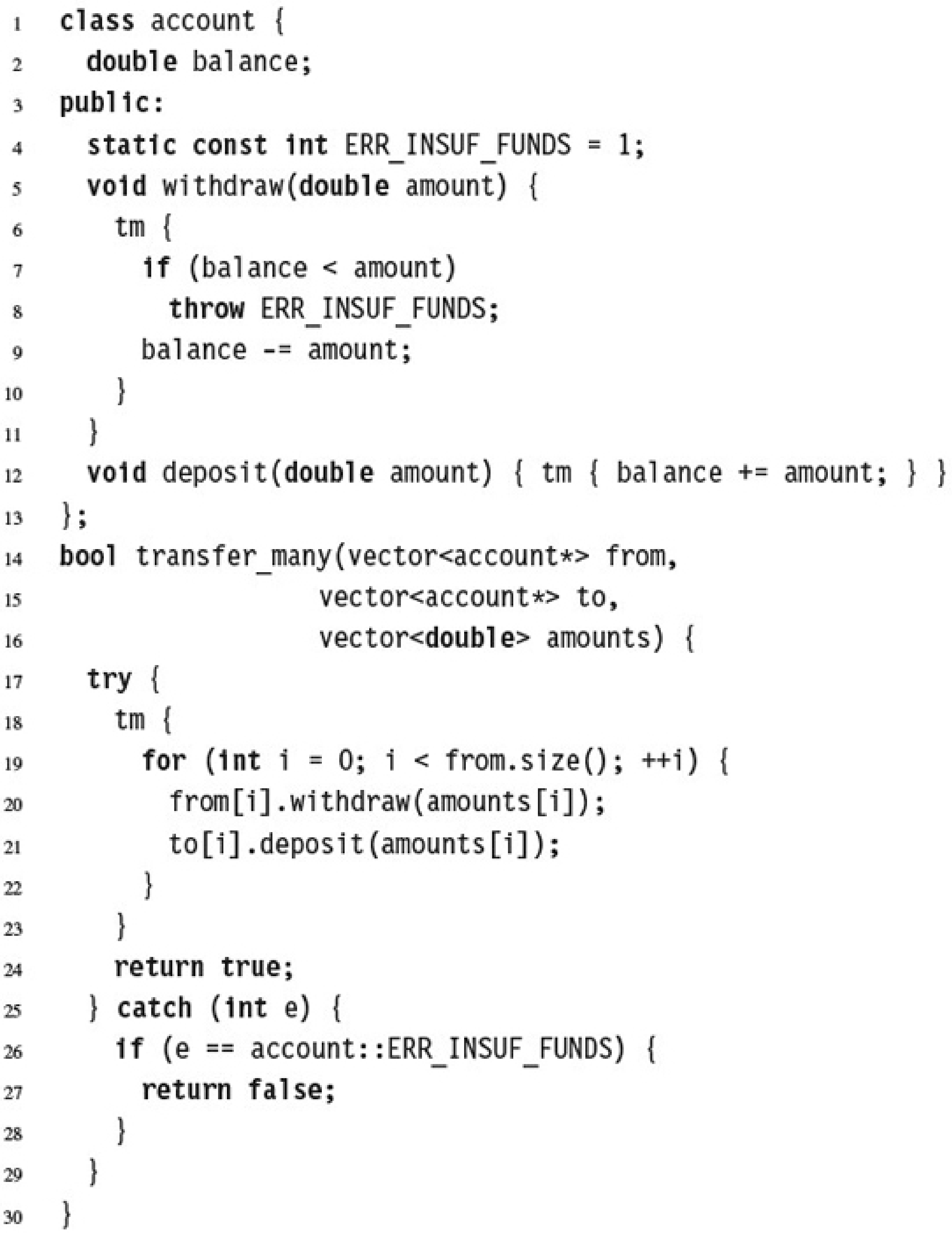

```
class account {
  double balance;
public:
  static const int ERR_INSUF_FUNDS = 1;
  void withdraw(double amount) {
    tm {
      if (balance < amount)
        throw ERR_INSUF_FUNDS;
      balance -= amount;
    }
  }
  void deposit(double amount) { tm { balance += amount; } }
};
bool transfer_many(vector<account*> from,
                   vector<account*> to,
                   vector<double> amounts) {
  try {
    tm {
      for (int i = 0; i < from.size(); ++i) {
        from[i].withdraw(amounts[i]);
        to[i].deposit(amounts[i]);
      }
    }
    return true;
  } catch (int e) {
    if (e == account::ERR_INSUF_FUNDS) {
      return false;
    }
  }
}
```

FIGURE 20.9 Code to atomically perform multiple transfers between accounts, using exception-based self-abort.