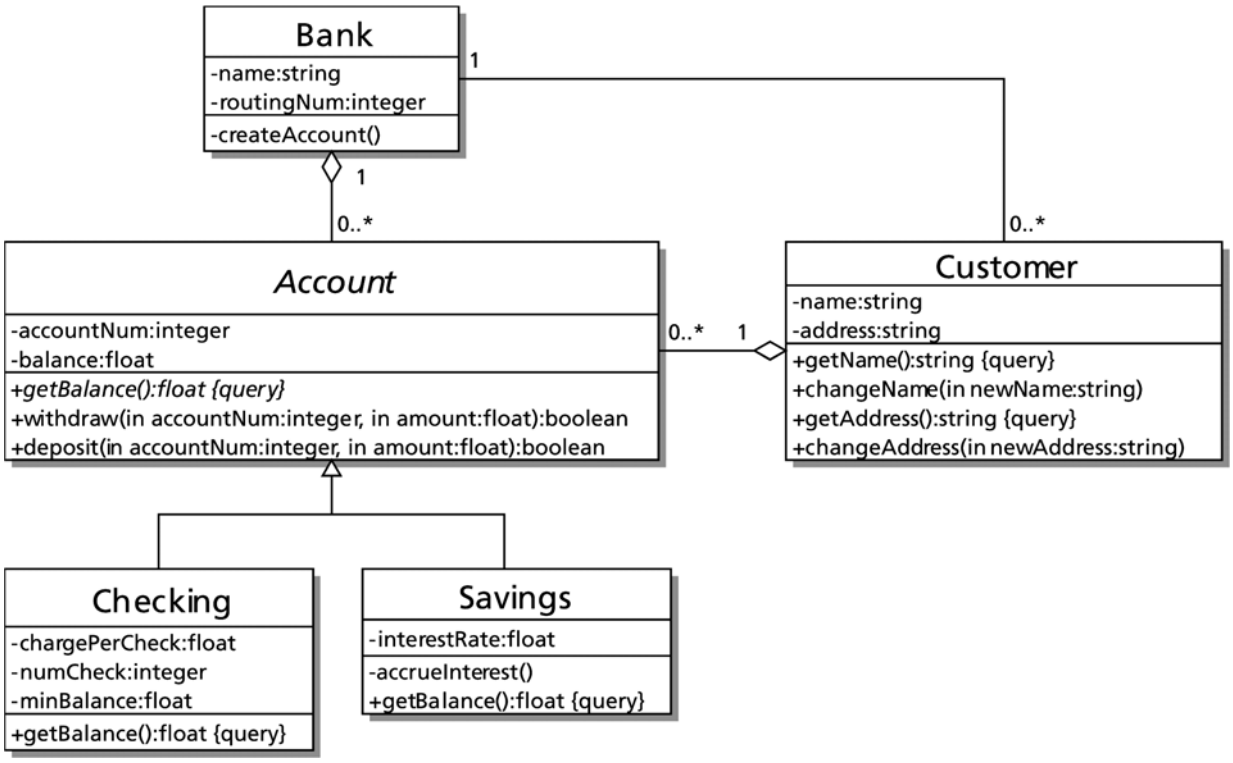
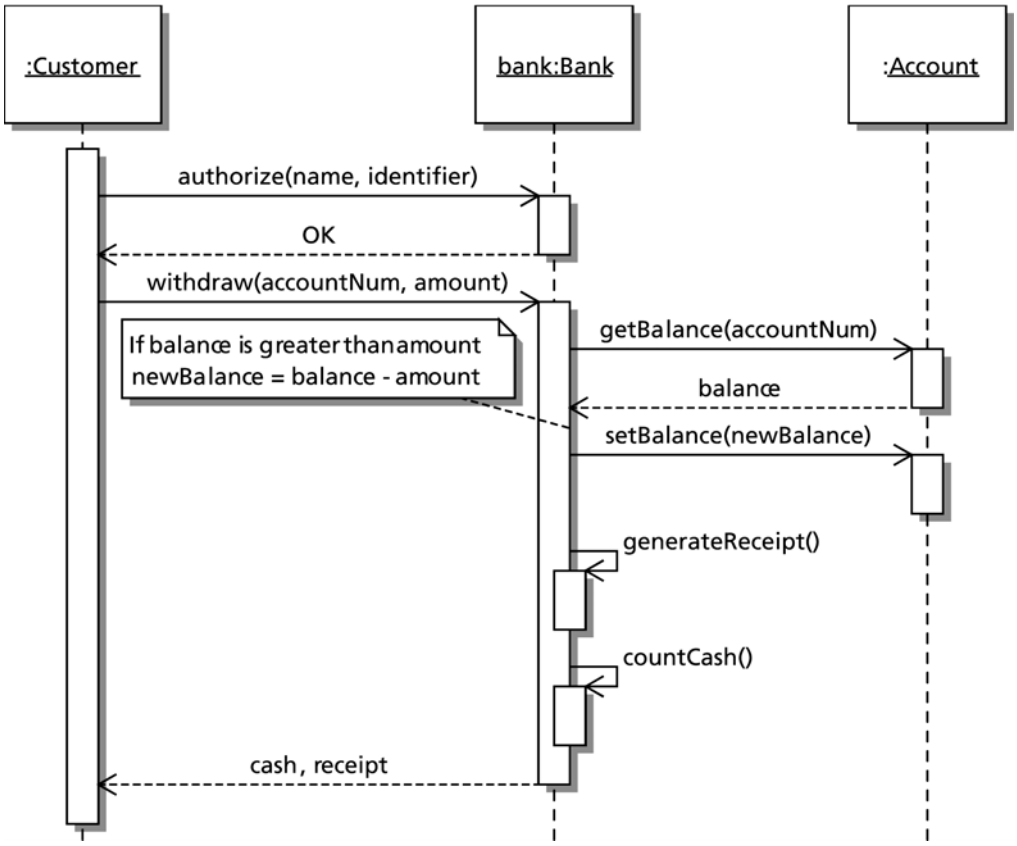
- Class diagrams represent ***static*** relationships. Why?



- What about modeling ***dynamic*** behavior?
- **Interaction** diagrams model how groups of object
  collaborate to perform some behavior
  - Typically captures the behavior of a single use case

Withdraw:

1. Authorization: Customer provides name, ID
2. Customer provides account # and amount to Bank
3. Bank checks balance
4. Bank: Generate receipt
5. Bank: count Cash
6. Return cash and receipt to customer



- Each **object** has a **lifeline** denoted by a vertical dashed line.
- Each **object** is represented as a box containing its name followed by a colon and its type, all underlined.
- Each lifeline can have one or more activations bars (**open boxes**) that show when the object is active and the responsibilities of the class.
- A **method call** appears as a solid arrow from the calling object's activation bar to the top of a new activation bar on the lifeline of the called object. Such arrows are **labeled with a message expression** that describes the task performed.
- Order of messages sequences top to bottom

- Messages labeled with message name
  - Optionally arguments and control information
- Control information may express conditions:
  - such as [hasStock], or iteration
- Returns (dashed lines) are optional
  - Use them to add clarity