

# Chapter 25

## ATM Case Study, Part 1: Object-Oriented Design with the UML

C++ How to Program, 9/e

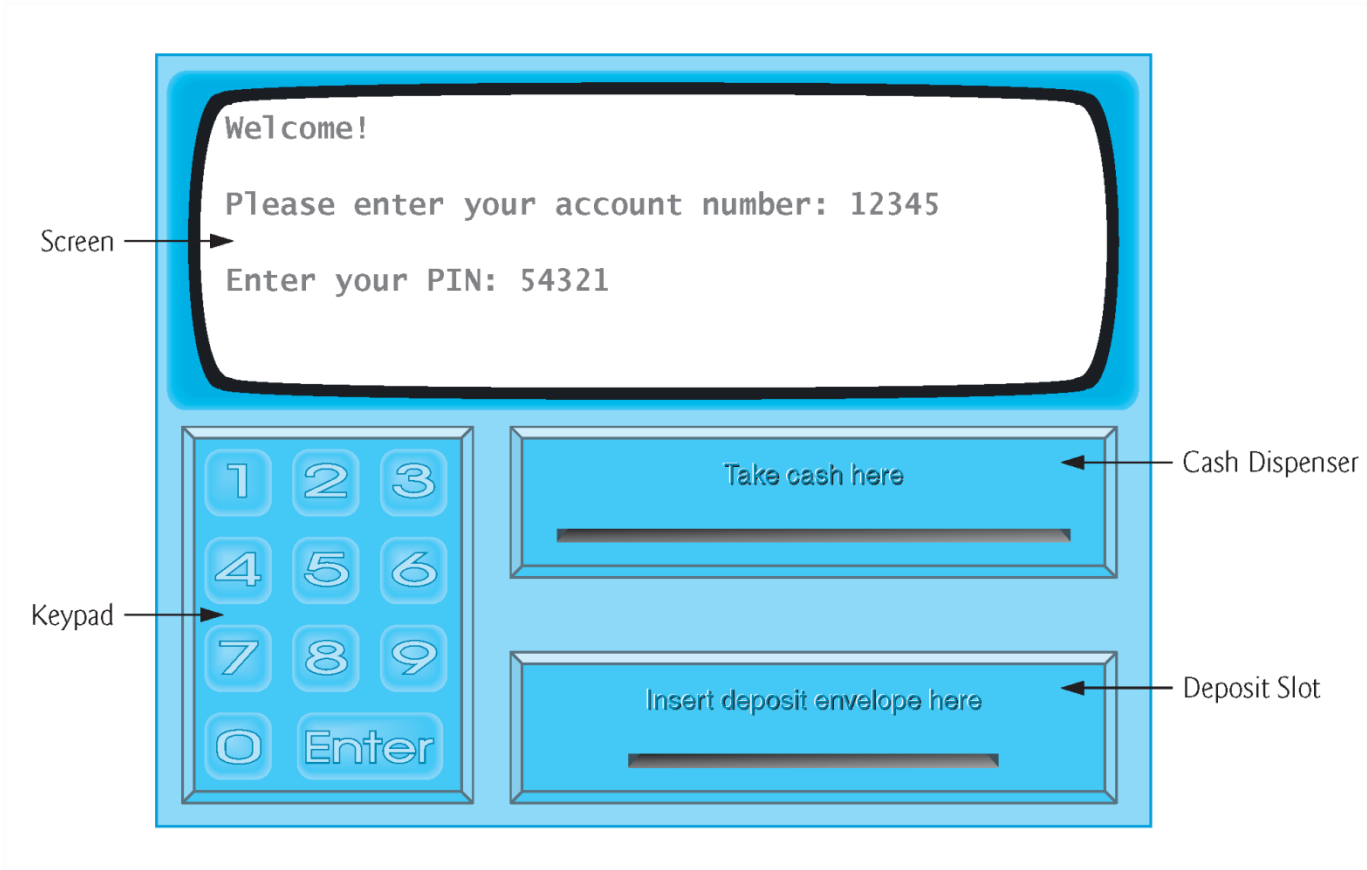
## OBJECTIVES

In this chapter you'll:

- Learn a simple object-oriented design methodology.
- Learn what a requirements document is.
- Identify classes and class attributes from a requirements document.
- Identify objects' states, activities and operations from a requirements document.
- Determine the collaborations among objects in a system.
- Work with the UML's use case, class, state, activity, communication and sequence diagrams to graphically model a simple object-oriented system.

- 25.1** Introduction
- 25.2** Introduction to Object-Oriented Analysis and Design
- 25.3** Examining the ATM Requirements Document
- 25.4** Identifying the Classes in the ATM Requirements Document
- 25.5** Identifying Class Attributes
- 25.6** Identifying Objects' States and Activities
- 25.7** Identifying Class Operations
- 25.8** Indicating Collaboration Among Objects
- 25.9** Wrap-Up

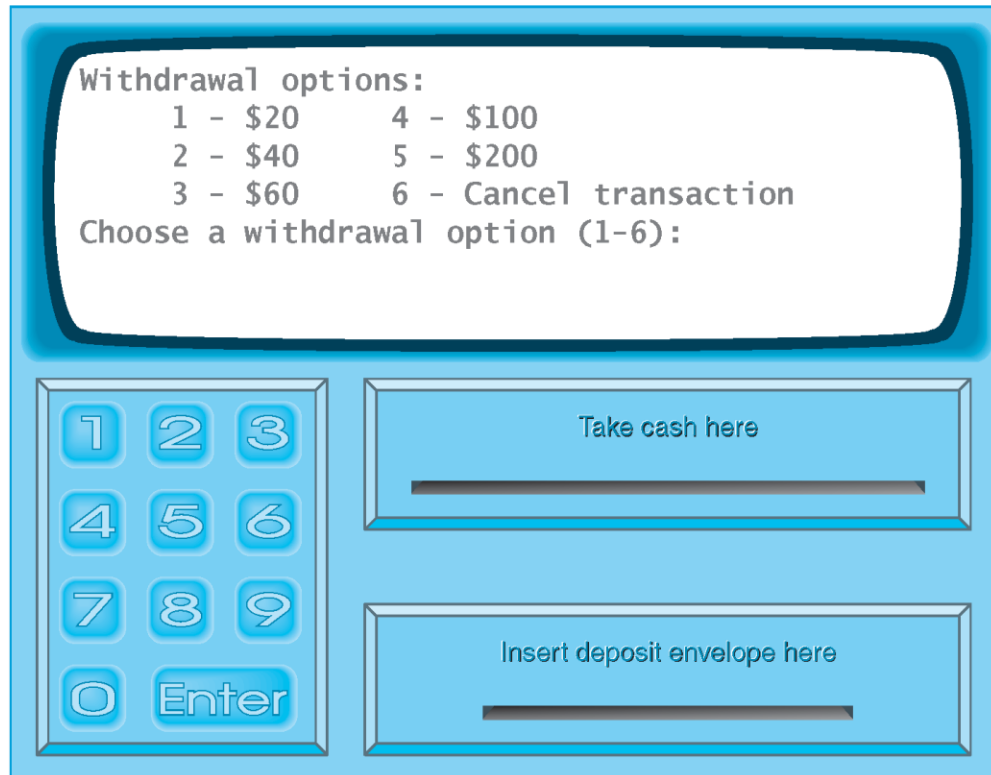
# 25.3 Examining the ATM Requirements Document



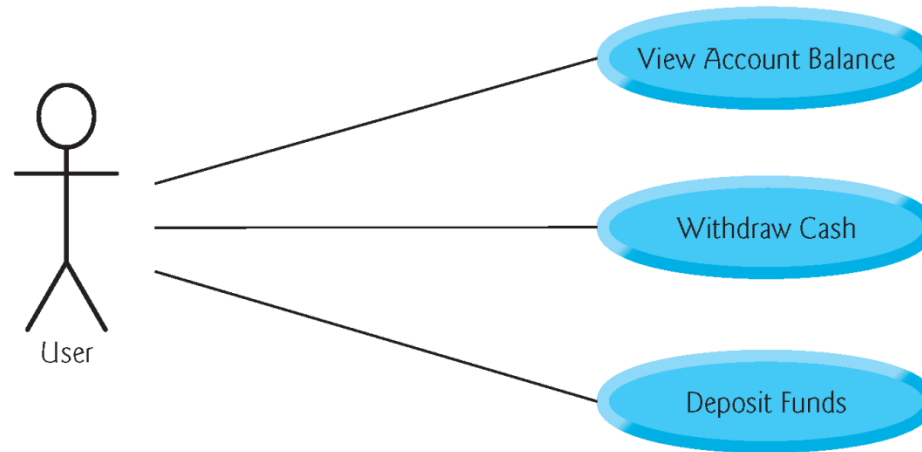
**Fig. 25.1** | Automated teller machine user interface.



**Fig. 25.2** | ATM main menu.



**Fig. 25.3** | ATM withdrawal menu.



---

**Fig. 25.4** | Use case diagram for the ATM system from the User's perspective.



## 25.4 Identifying the Classes in the ATM Requirements Document

## Nouns and noun phrases in the requirements document

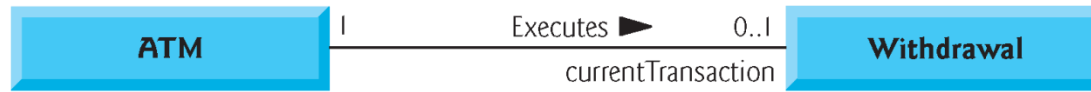
bank	money / fund	account number	ATM
screen	PIN	user	keypad
bank database	customer	cash dispenser	balance inquiry
transaction	\$20 bill / cash	withdrawal	account
deposit slot	deposit	balance	deposit envelope

**Fig. 25.5** | Nouns and noun phrases in the requirements document.



---

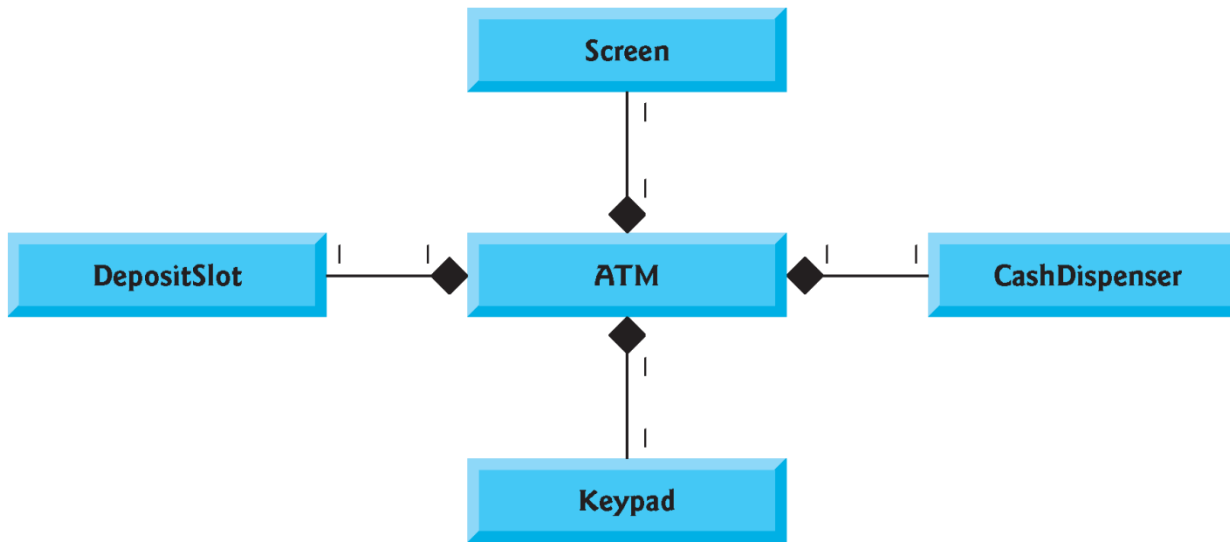
**Fig. 25.6** | Representing a class in the UML using a class diagram.



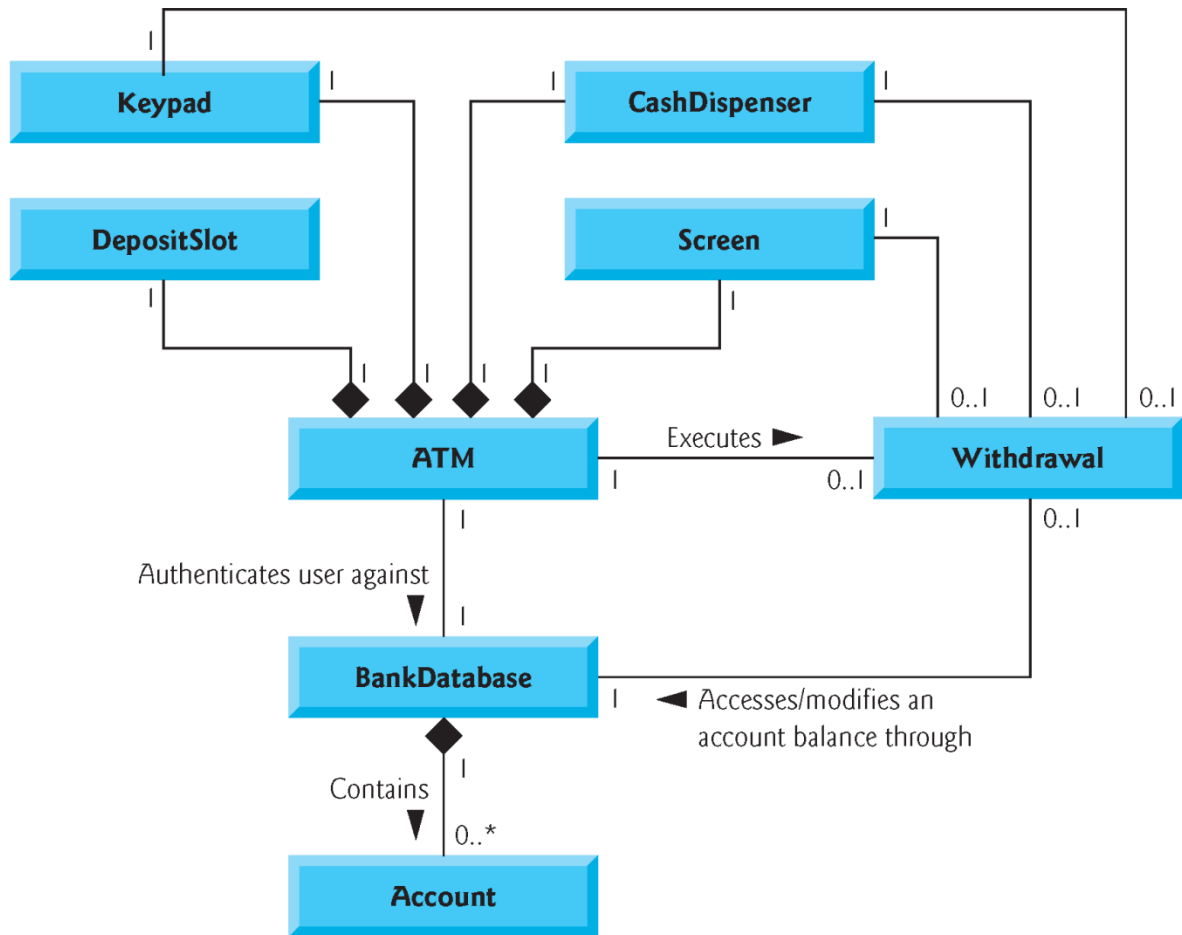
**Fig. 25.7** | Class diagram showing an association among classes.

Symbol	Meaning
0	None
1	One
$m$	An integer value
0..1	Zero or one
$m, n$	$m$ or $n$
$m..n$	At least $m$ , but not more than $n$
*	Any nonnegative integer (zero or more)
0..*	Zero or more (identical to *)
1..*	One or more

**Fig. 25.8** | Multiplicity types.



**Fig. 25.9** | Class diagram showing composition relationships.



**Fig. 25.10** | Class diagram for the ATM system model.