

CHAPTER 3

Data Abstraction: The Walls

Data Abstraction and Problem Solving with JAVA:
Walls and Mirrors
Carrano / Prichard

Figure 3.1

Isolated tasks: the implementation of task T does not affect task Q

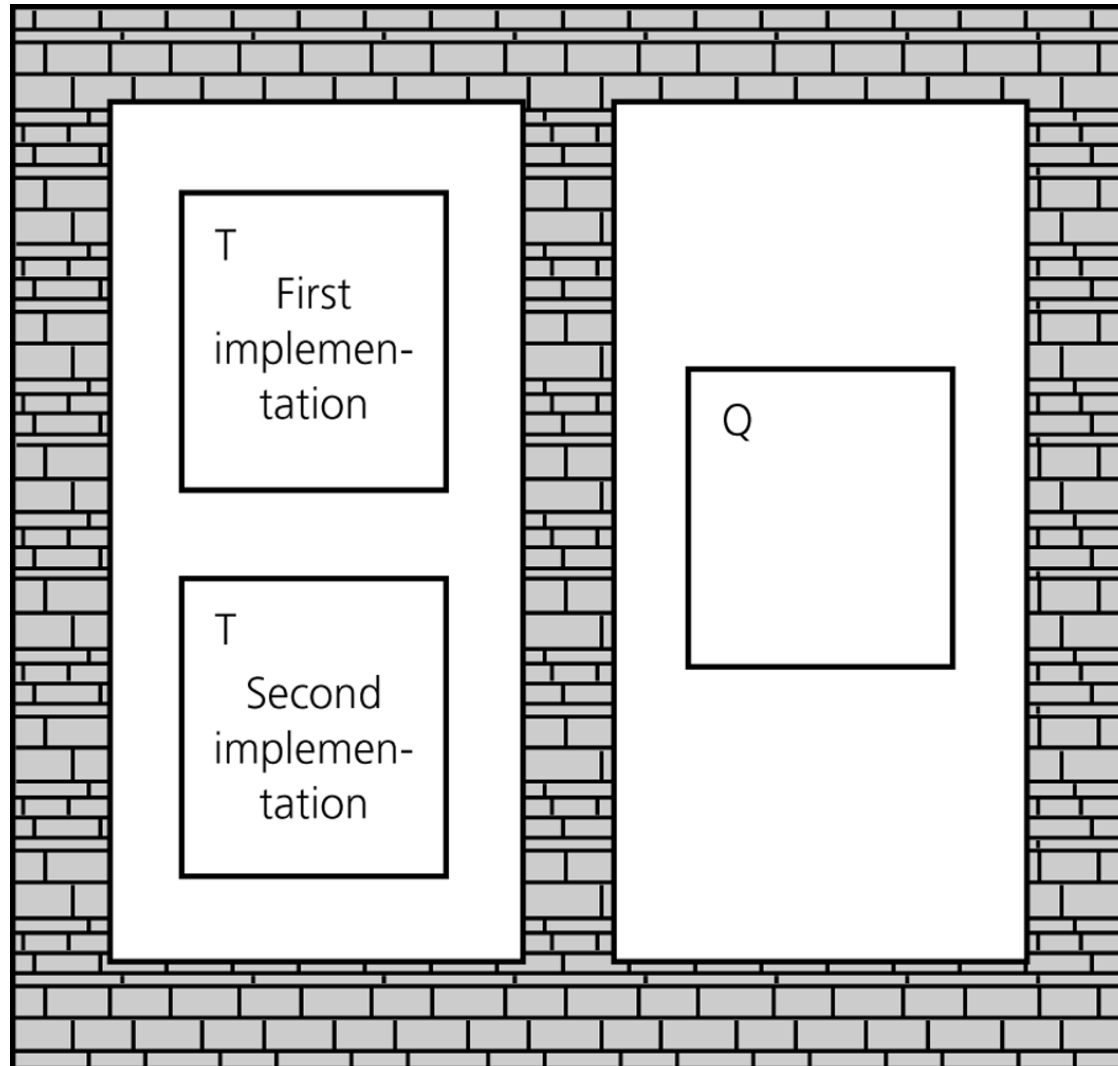


Figure 3.2

A slit in the wall

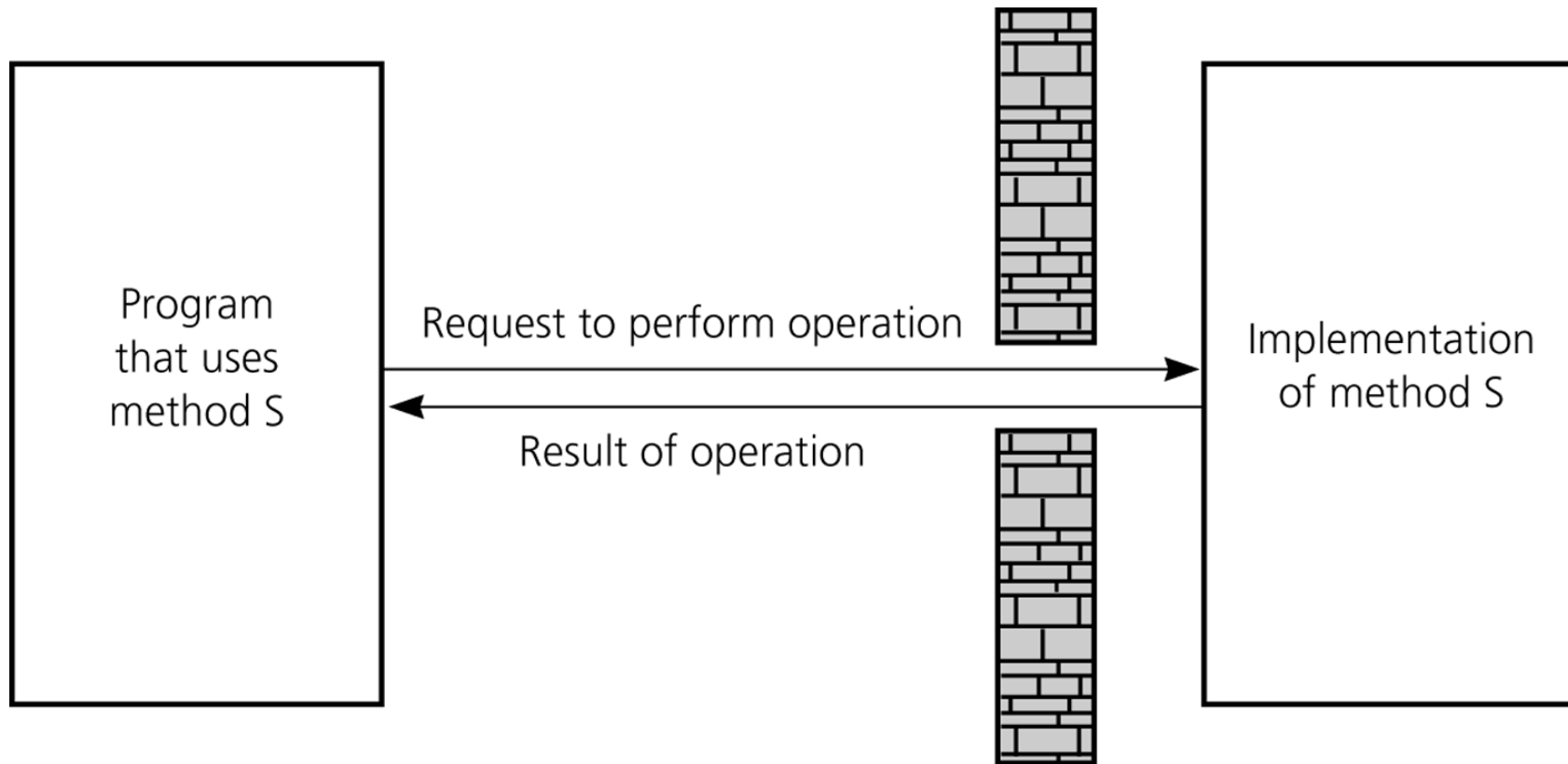


Figure 3.3

A dispenser of chilled water, crushed ice, and ice cubes

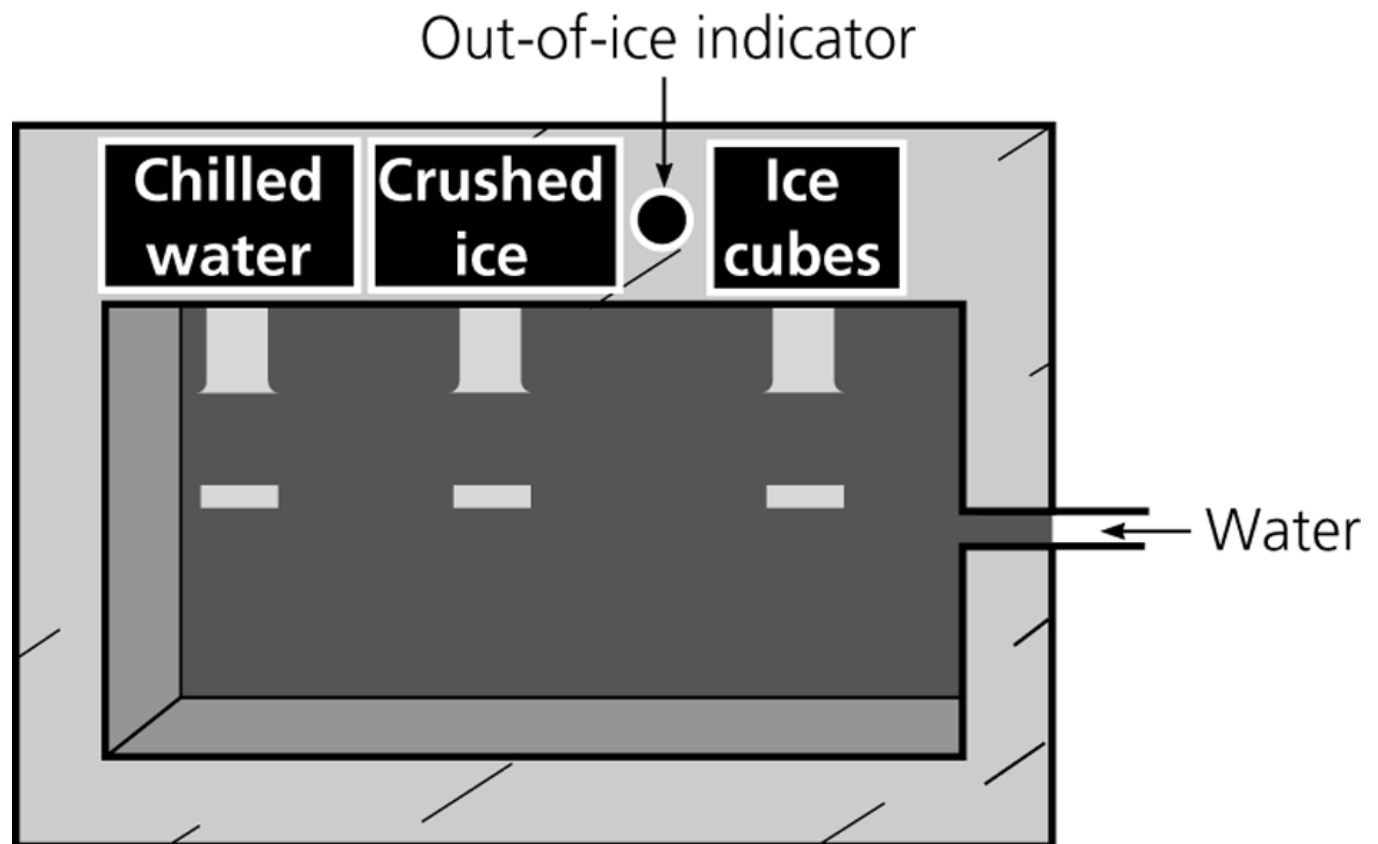


Figure 3.4

A wall of ADT operations isolates a data structure from the program that uses it

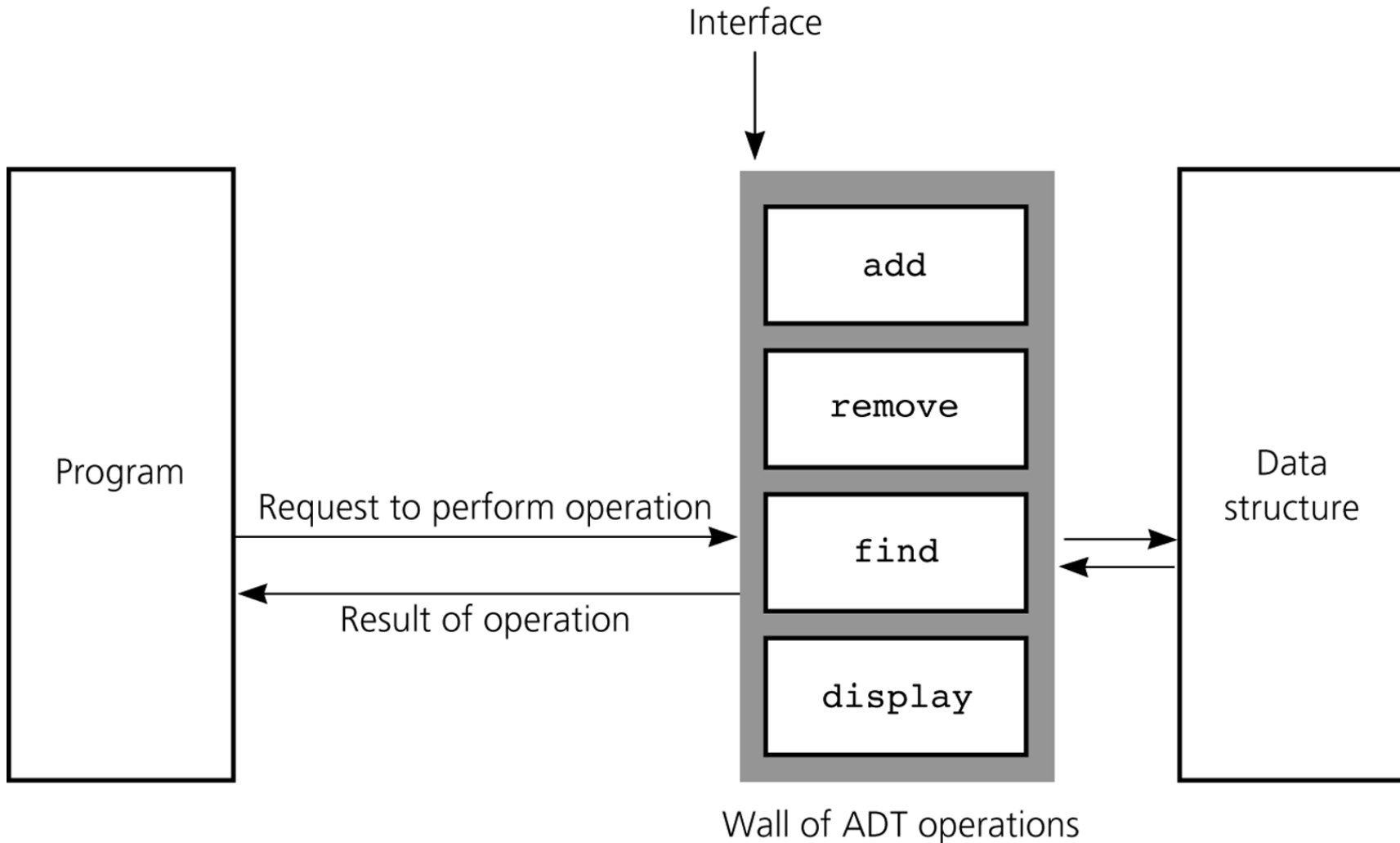


Figure 3.5

A grocery list

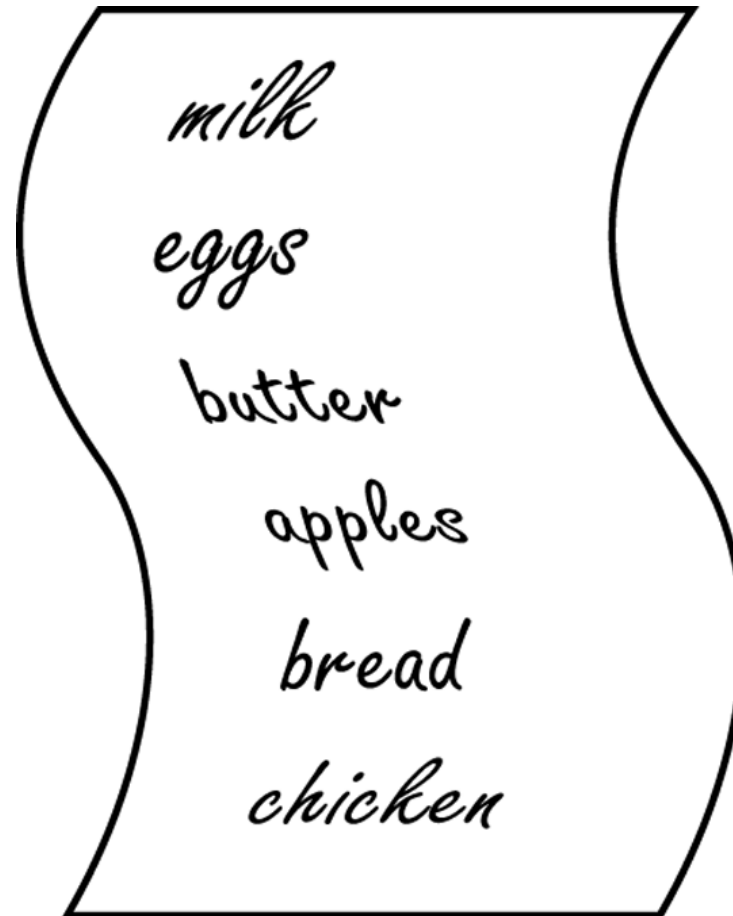


Figure 3.6

The wall between *displayList* and the implementation of the ADT list

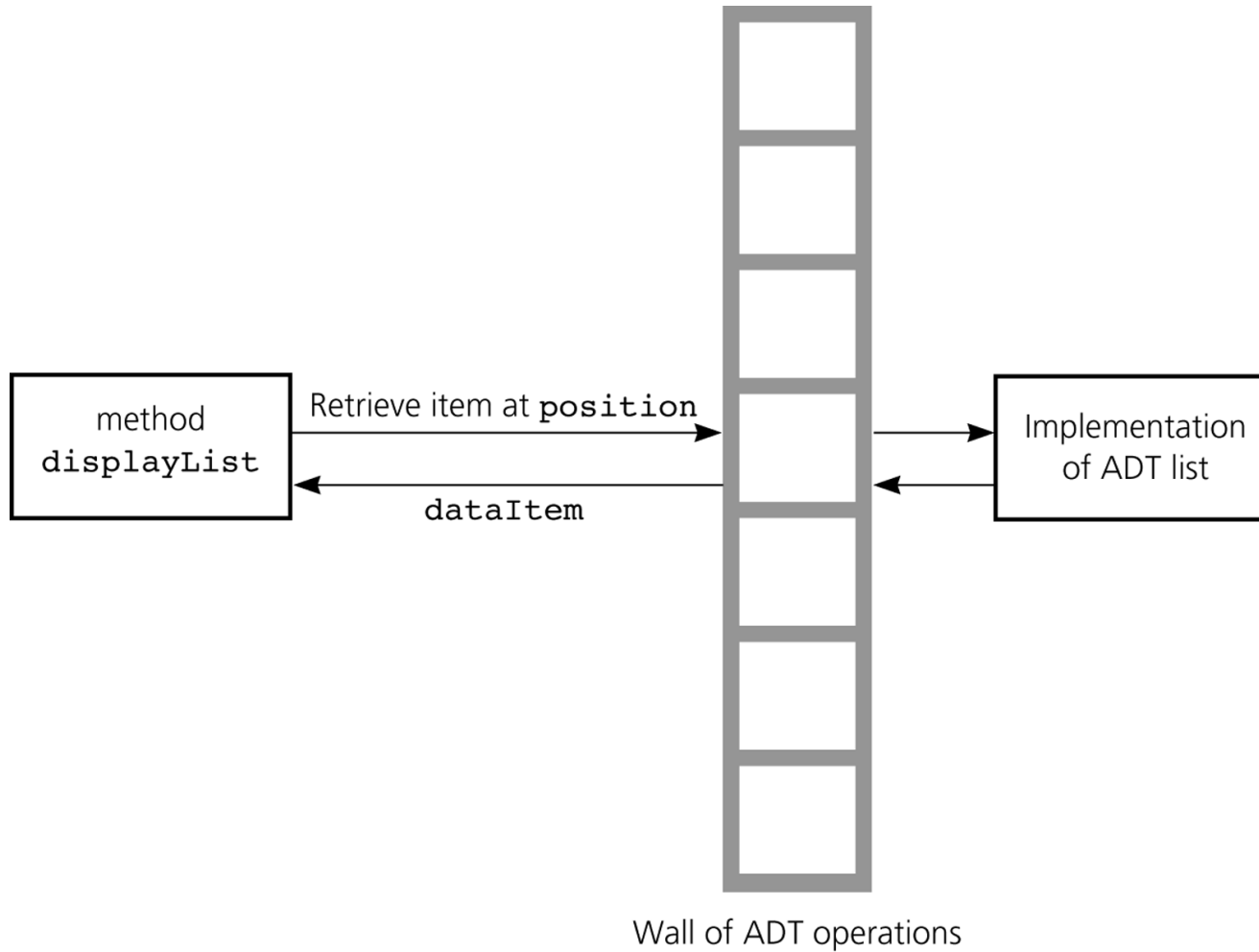


Figure 3.7

ADT operations provide access to a data structure

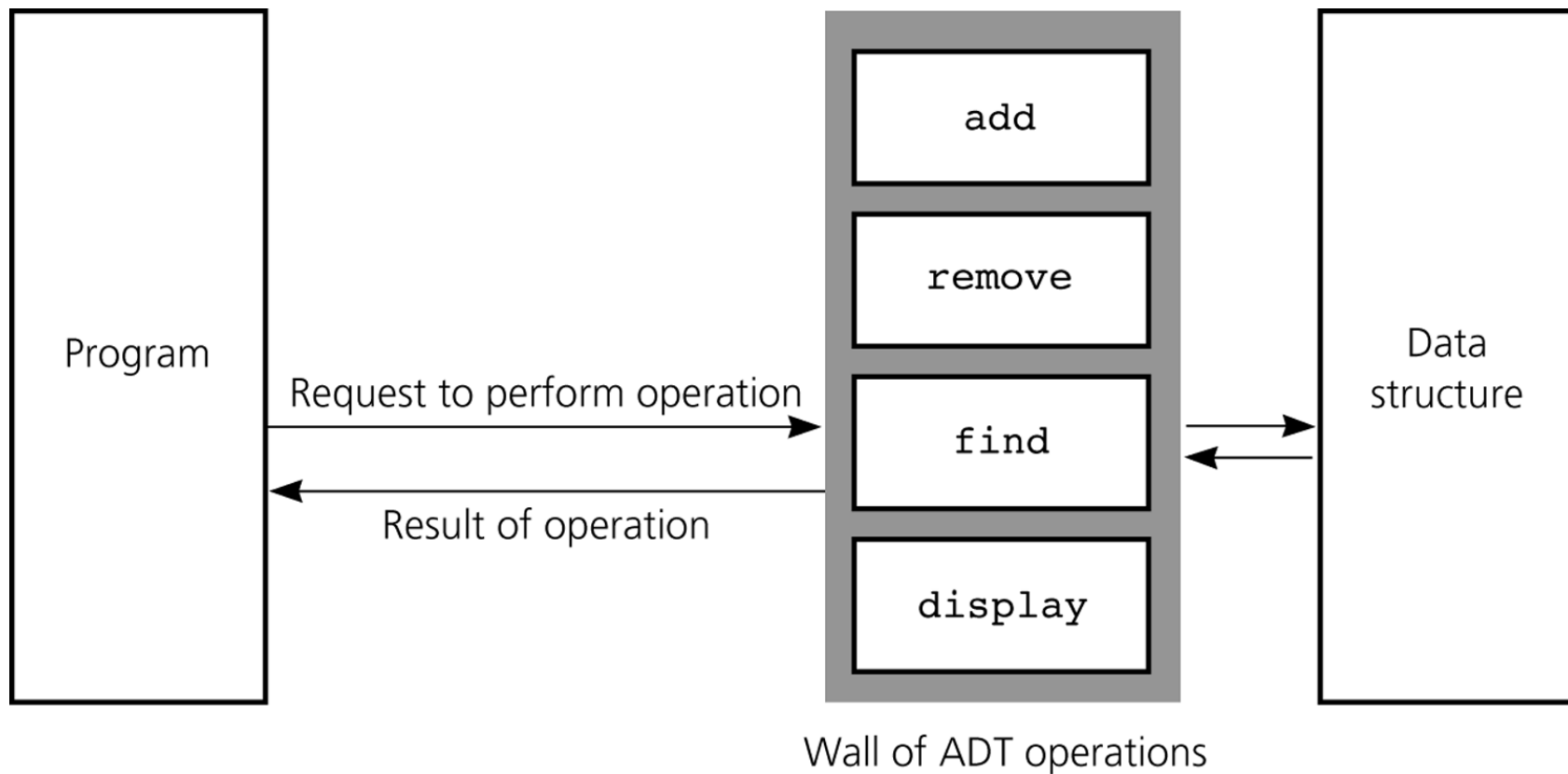


Figure 3.8

Violating the wall of ADT operations

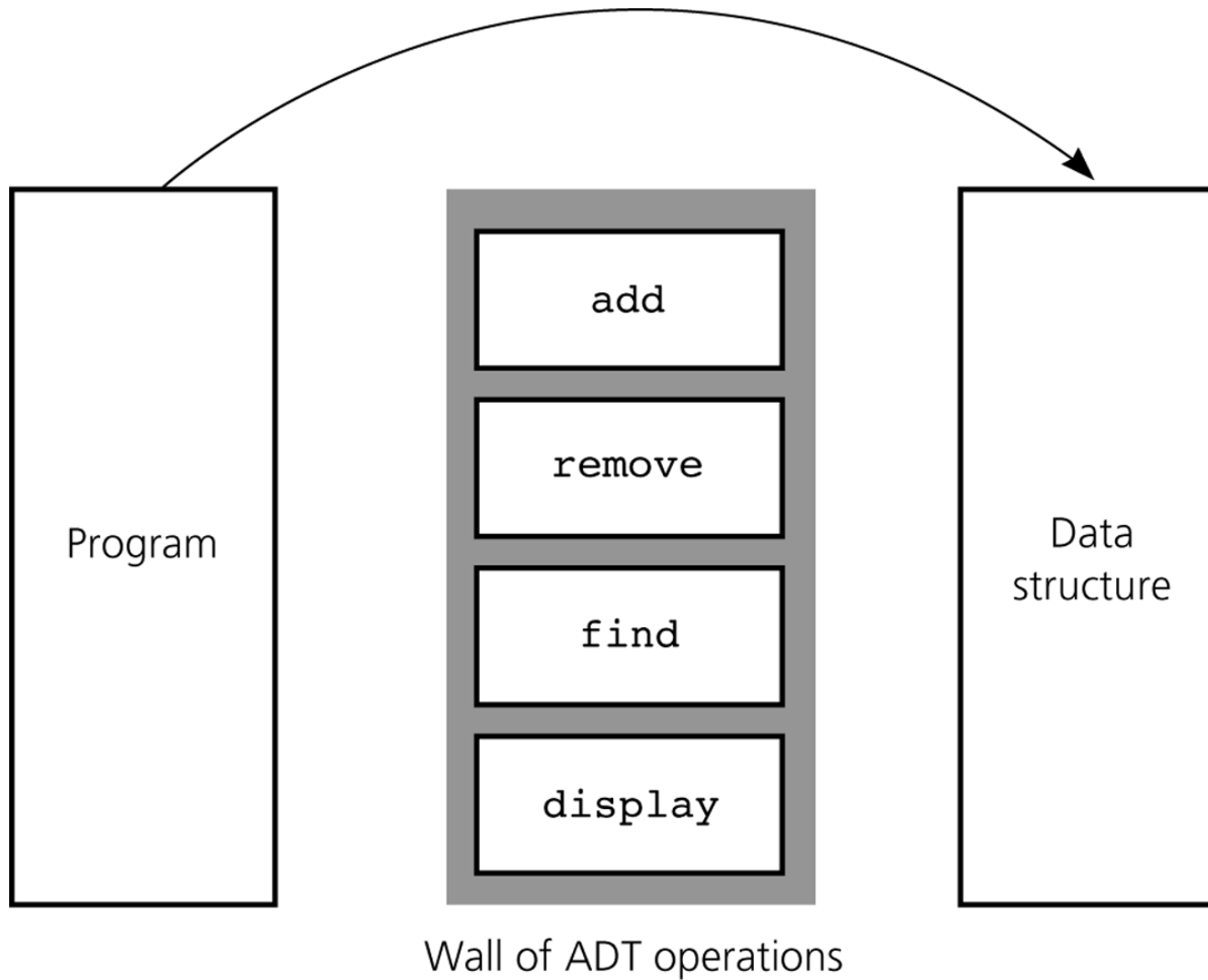


Figure 3.9

An object's data and methods are encapsulated

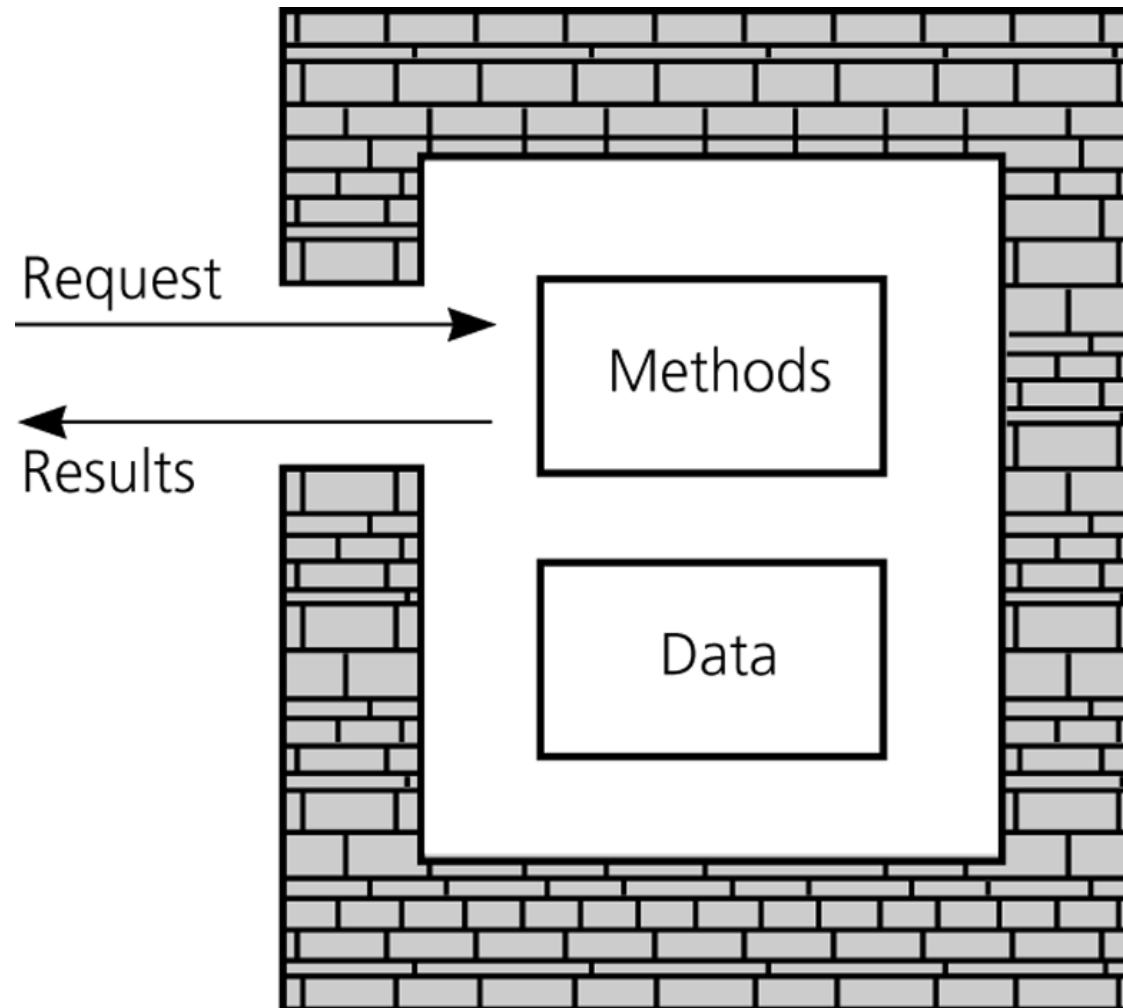


Figure 3.10

An array-based implementation of the ADT list

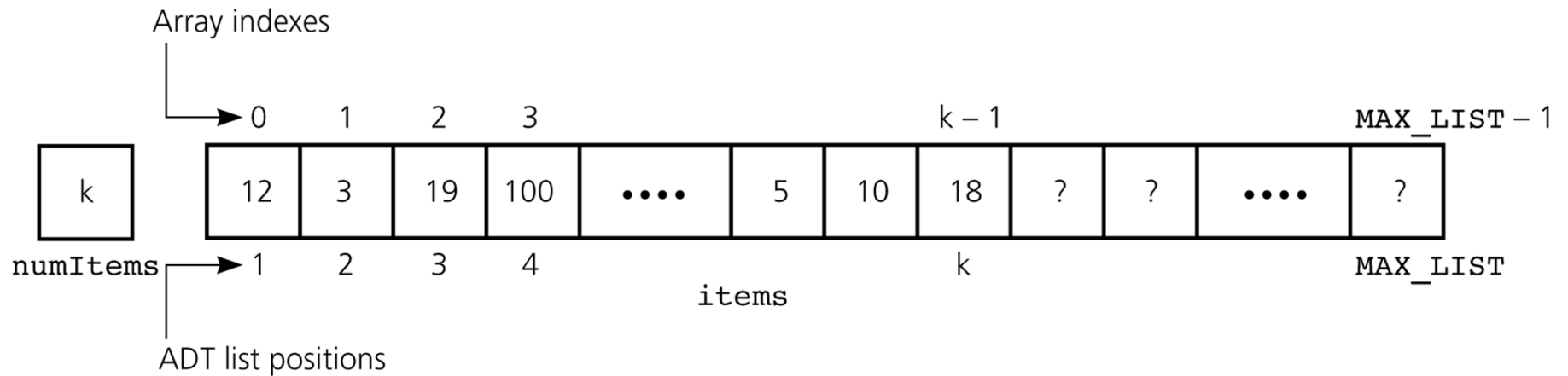


Figure 3.11

Shifting items for insertion at position 3

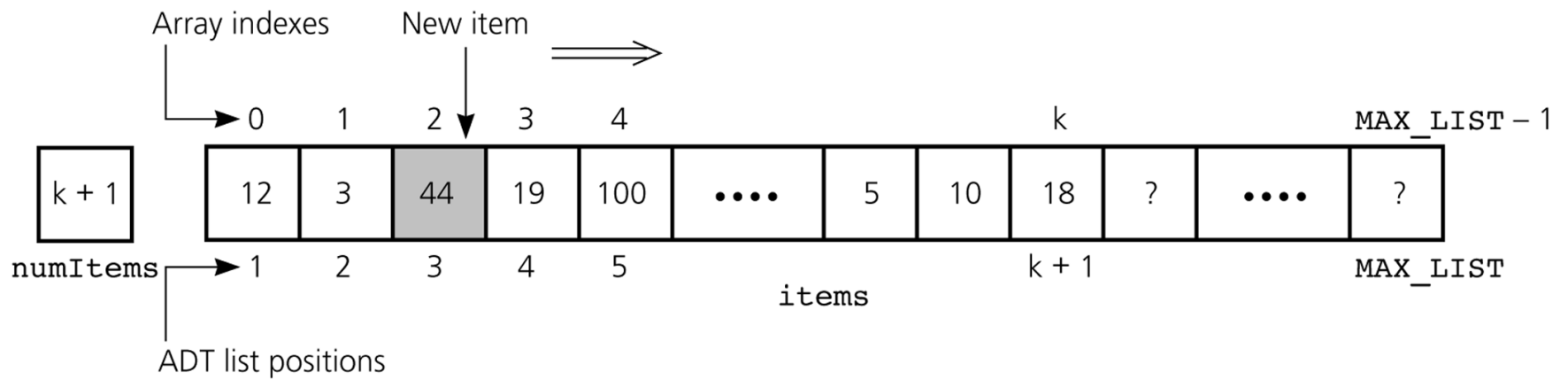


Figure 3.12

a) Deletion causes a gap; b) fill gap by shifting

