OBJECT BASICS

zyBook 9.1

@ Dr. Jessica Young Schmidt and NCSU Computer Science Faculty

OBJECTS

Object → A programming entity that contains state (data) and behavior (methods)

- State
  - A set of values (internal data) stored in an object.
  - Represented by fields within the class.
- Behavior
  - A set of actions an object can perform, often reporting or modifying its internal state.
  - Represented by instance methods within the class.

An object is defined by a class.

CLASSES AND OBJECTS

A class is a blueprint or template for constructing objects.

- Class
  - Define collections of procedures or actions
  - Define new data/object types
- Object
  - A programming entity that contains state (data) and behavior (methods)

POINT OBJECTS

https://docs.oracle.com/en/java/javase/11/docs/api/java.desktop/java/awt/Point.html

- (State) Data stored for each Point object
  - x : The point's x coordinate
  - y : The point's y coordinate
- (Behavior) Methods that can be run on Point objects
  - public void setLocation(double x, double y) → Sets the location of this point to the specified double coordinates.
  - public void translate(int dx, int dy) → Translates this point, at location (x,y), by dx along the x axis and dy along the y axis so that it now represents the point (x+dx, y+dy).

Class

Point blueprint

state:
- x
- y

behavior:
- distance(point)
- setLocation(x, y)
- translate(dx, dy)

creates

Instance

Point #1

state:
- x = 0
- y = 0

behavior:
- distance(point)
- setLocation(x, y)
- translate(dx, dy)

x 0 y 0

Point #2

state:
- x = 3
- y = 7

behavior:
- distance(point)
- setLocation(x, y)
- translate(dx, dy)

x 3 y 7

Point #3

state:
- x = 5
- y = 6

behavior:
- distance(point)
- setLocation(x, y)
- translate(dx, dy)

x 5 y 6

CLIENT CODE

- Objects themselves are not complete programs; they are components that are given distinct roles and responsibilities.
- Objects can be used as part of larger programs to solve programs.
- A major benefit of objects is that they provide reusable pieces of code that can be used in many client programs (Reges and Stepp)
- Client (or Client Code)
  - code that interacts with a class or objects of that class.

ABSTRACTION

Abstraction → Focusing on essential properties rather than inner details.

- A distancing between ideas and details.
- We have been using Java Class Library objects without knowing how they work.
- When creating our own new class, we are abstracting the functionality of the class for client programs

Abstraction of Point

- You understand its external behavior.
- You do not need to understand its inner details

HUNT LIBRARY EXAMPLE

Hunt Library features an automated book retrieval system. Let's design its book retrieval software:

- Catalog of books are kept in an array
- Want to find a book by title, author, publication year, etc.
- Want to track books that are checked out and when they are due

Bad Implementation of Hunt Library:

```
import java.util.Scanner;

public class HuntLibrary {
    public static int NUM_BOOKS = 1000;

    public static void main(String[] args) {
        // Parallel arrays for all book items
        String[] authors = new String[NUM_BOOKS];
        String[] titles = new String[NUM_BOOKS];
        int[] pubYear = new int[NUM_BOOKS];

        Scanner input = new Scanner(System.in);
        for (int i = 0; i < authors.length; i++) {
            System.out.print("Author: ");
            authors[i] = input.nextLine();
            System.out.print("Title: ");
            titles[i] = input.nextLine();
            System.out.print("Publication year: ");
            pubYear[i] = input.nextInt();
        }
    }
}
```

Parallel arrays make it difficult to keep data straight!

A Book object would help with this implementation.

Book Object

Book object would hold data about itself (state):

Type	Field Name	Description
`String`	`title`	Name of the book
`String`	`author`	Author of the book
`int`	`pubYear`	Publication year of the book
`String`	`checkedOut`	Person who has checked out the book
`int`	`location`	Location of the book in the stacks
`Date`	`dueDate`	Date a checked out book is due

Each Book object should be able to (behavior):

Return Type	Method Name	Description
`boolean`	`checkOut(unityId)`	Check out a book
`void`	`checkIn()`	Check in a book
`int`	`getLocation()`	Info on where the book is stored
`void`	`setLocation(loc)`	Put book at this location in the library

`Date` : `https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/Date.html`

RECAP – CLASSES AND OBJECTS

A class is a blueprint or template for constructing objects.

- Class
  - Define collections of procedures or actions
  - Define new data/object types
- Object
  - A programming entity that contains state (data) and behavior (methods)