Object-oriented programming (OOP)

Overview

- Models and mimics real-world concepts through OOP elements
  - Tailored to the human mind rather than the machine
  - The focus is on the task for which the computer is used rather than the way a computer handles the task
- Elements of OOP - classes, abstract classes, interfaces, objects, attributes, methods, etc.
- Four major principles of OOP - abstraction, encapsulation, inheritance, polymorphism
- More difficult to master than the Java programming language
- Common mistake - procedural code masked in an object-oriented language

OOP model example

- Stereo system:
  - Speakers - play mid-range and high-frequency sounds
  - Subwoofer - plays low bass frequency sounds
  - Tuner - receives radio broadcast signals
  - CD player - plays audio data from CDs
- Key concepts:
  - The components are self-contained elements that perform a specific function
  - They can be combined and reused
  - They interconnect through standardized connectors

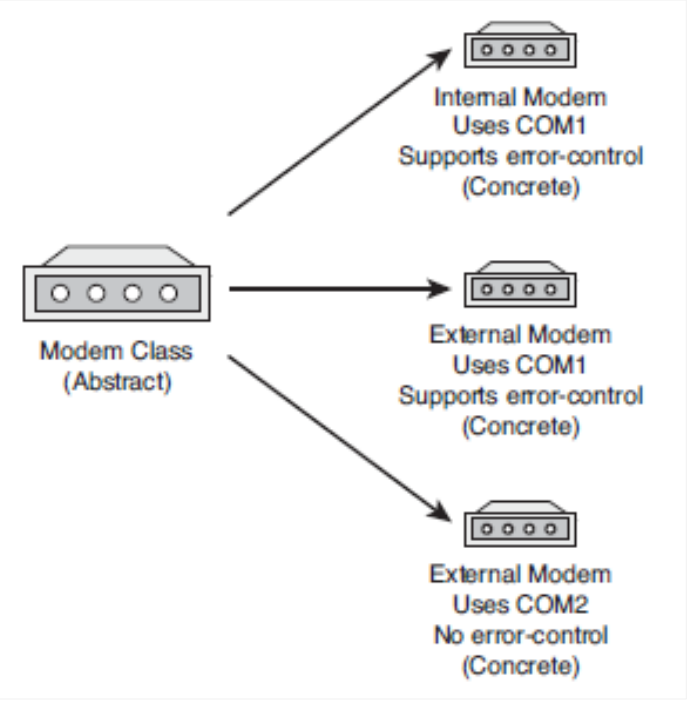
Object and classes

- Objects are software bundles of related **state** and **behavior** which model real-world objects.
- A class is a **template** (blueprint) used to create objects.
- Objects created from the same class have similar features.

Example

A Modem class

- Connects to a computer through a serial port
- Can dial a phone number
- Can send and receive data



Attributes and behavior

- An object's state is expressed through **attributes**.
- Attributes In Java are called **variables**.
  - Instance variables - object specific
  - Class (static) variables - class specific i.e. they relate to an entire class of objects created from a class
- The **behavior** of an object relates to the things that the object can do to themselves and to other objects.
- In Java behavior is expressed through **methods**.

Example: VolcanoRobot

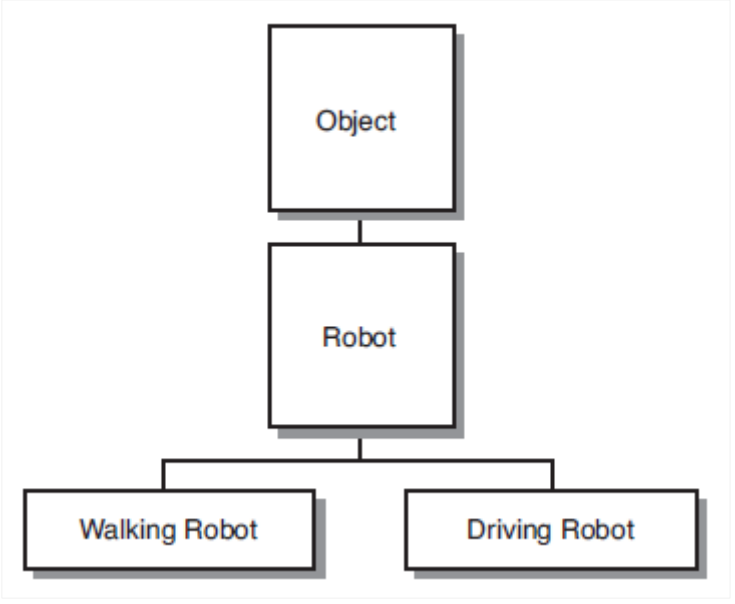
A VolcanoRobot class used to do research inside volcanic craters.

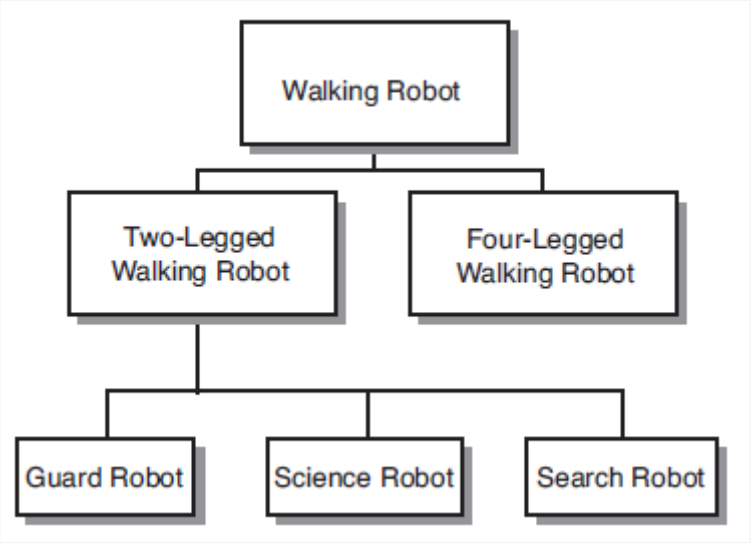
- Variables:
  - Status - Exploring, moving, returning home
  - Speed - Measured in kilometers per hour
  - Temperature - Measured in degrees centigrade
- Methods:
  - Check current temperature
  - Begin a survey
  - Report current speed

Inheritance

- Allows for one class, a **subclass**, to inherit the attributes and behavior of another class called a **supperclass**.
- The subclass specifies how it differs from its supperclass.
- Advantages:
  - Functionality common to multiple classes can be put into a superclass, which enables it to be used repeatedly in all classes below it in the hierarchy.
  - Changes to a superclass automatically are reflected in all its subclasses, their subclasses, and so on.

Class hierarchy example





Inheritance in Java

- One subclass can have only one superclass - single inheritance.
- One superclass can have an unlimited number of subclasses.
- If a class doesn't explicitly inherit from another class, it inherits implicitly from Java's Object class.

Exercises

Exercise: VolcanoRobotVirgil

In the main() method of the VolcanoApplication class, create a second VolcanoRobot robot named virgil, set up its instance variables, and display them.