Head First Java: Chapter 2 Notes

August 27, 2019

Object oriented programming (OOP):

- An *object* is made according to a defined *class*. A class is NOT an object, but rather the **blueprint** for objects.
- A class tells the JVM *how* to create that specific object.
- An object is an *instance* of a class, often saying "instance" is just another way of saying "object".
- What an object "knows", or its "state", is called *instance variables.* E.g., an object named "Alarm" might have instance variables like "alarmTime" and "alarmMode".
- What an object "does", or its "behavior" is called *methods.* For the above Alarm object it might have methods like "setAlarmTime()", "getAlarmTime()", "isAlarmset()", and "snooze()".
- It is common for objects to have methods that write or modifies the values of their own instance variables.
- In a Java OO program, global variables or methods/functions are not common, but they can be approximated by marking a method `public` and `static` or a variable `public`, `static`, and `final`.
- In each `.java` file there can only be **one** public class; if there are multiple public classes they must be contained separately in different `.java` files. The `.java` file must be named **exactly** (including case) as the public class in the file.