An Overview of Java

Why Program in Java

Java is an object-oriented programming language. *Object-Oriented Programmin* (OOP) evolved out of the need to develop complex programs in a systematic, modular approach.

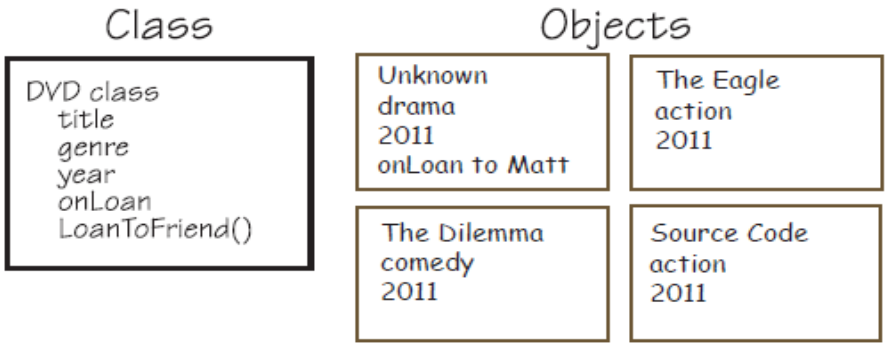
- OOP programmers create modules, called **classes** that can be used over and over in a variety of programs
- A class groups related data and the instructions for performing actions on that data
- Properly designed classes
  - use a concept called encapsulation to hide implementation details
  - are versatile enough to be extended through inheritance
  - give the programmer options through polymorphism

Features of every object-oriented language

- The Java Platorm runs on **all major operating systems** allowing Java applications to run on just about any computer (known as *platform-independent applications*)

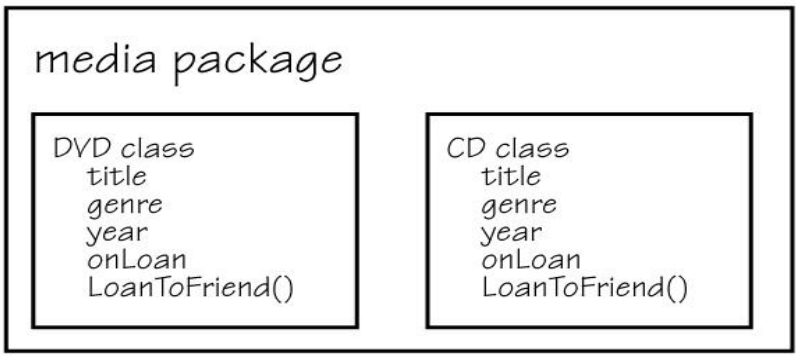
Objects, Classes and Packages

- OOP development involves selecting objects to perform tasks outlined in program specification
- An *object* consists of:
  - Related data
  - Instructions for performing actions on that data
- A *class*
  - Defines the type of data and actions that will be associated with an object of that class (but not the actual data for an individual object)
  - Is required to create objects

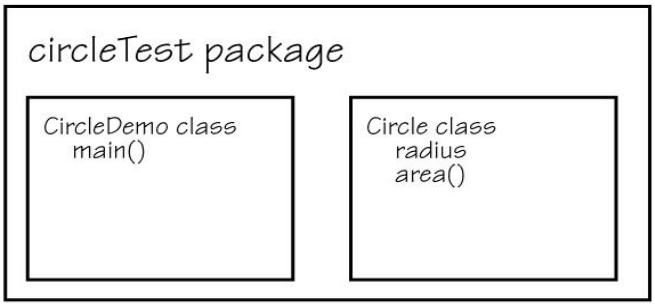


A class can be used to create many objects — each object will have the same type of data and possible actions, but each object maintains its own set of data.

- A *package* (sometimes called a library) is a group of related classes
- Here is an illustration of a DVD class and a CD class put into a Media Package:



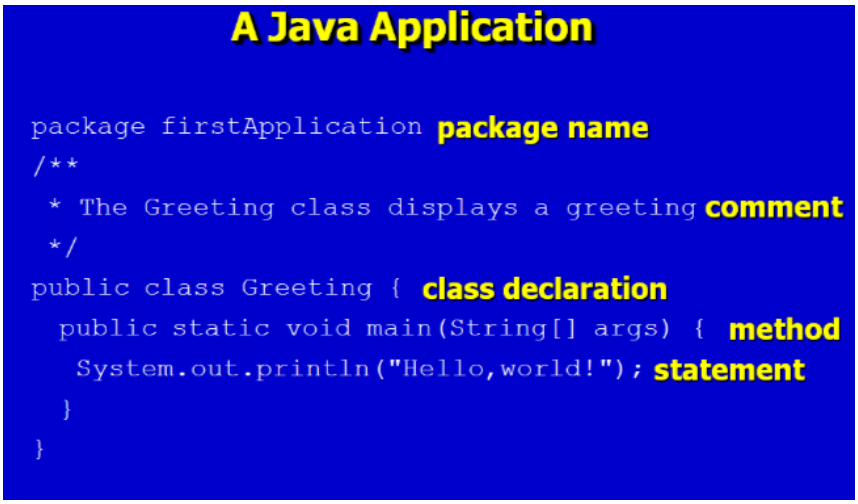
- An **application** can also be contained in a package.



Classes are importable – an application can use existing classes from another package

Our First Java Application

- A *Java Application* is a package with at least one class that contains a main() method. The following Greeting application is a simple, yet complete, Java program:



The Parts of a Java Application

Statements:

- The set of instructions that make up the program
- Each statement ends with a semi-colon ;
- Related statements are enclosed in a curly bracket {}
- In our application the statements are

    - `package firstApplication`
        - declares a package name, if undeclared it will be Default.
        - Not usually included in simple applications

    - `public class Greeting`
        - class declaration
        - the class is available to everyone (public)
        - it is called "greeting
        - no objects or actions are defined by the class (we will get to this later)
        - it does contain a main method (which is where the program will start)

    - `public static void main(String[] args)`
        - defines the `main` method
        - a method is a named set of statements that perform a single, well-defined task

The controlling class is a program's starting point. The main method is in the controlling class - it's statements are automatically run when the program is executed.

Comments:

- comments provide information about the program to the reader of the code
- Comments have no effect on the program run
- Allow the reader to quickly understand the purpose and logic behind segments of code
- Complex applications are often developed and maintained y more than one programmer, properly commented code allows for easier modifications and can decrease the number of mistakes
- Java programs can contain three types of comments:

    1) `/*  */`

        Enclose single or multiline comments at the beginning of a program to describe the application or inside to clarify a segment of code

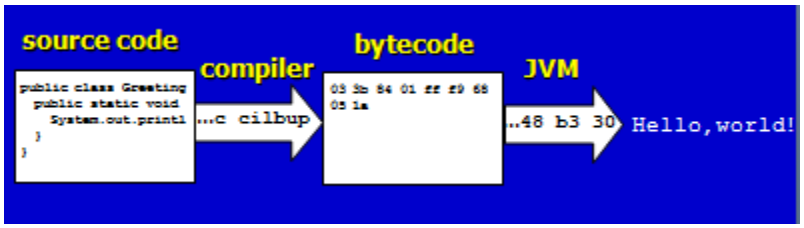
    2) `//`

        Used for adding a comment to a specific statement or for a single line comment -- Useful for debugging

3) /** */

Used for documentation
The javadoc tool copies these comments into a separate HTML document to create an instruction manual or external documentation
These are appropriate for describing a class or method

Executing a Java Application



- The programmer types in the Source Code (files with extension .java)
- Compiling: In order to execute, or run, it must be translated into code the computer understands (files with extension .class)
- Compiled java source code is called bytecode which is interpreted with a Java Virtual Machine
- The Java Virtual Machine can reside on any computer, regardless of operating system or environment

Syntax Errors

- A syntax error occurs in a statement that violates the rules of Java
- For example: A missing semicolon
- A program containing syntax errors will not compile