Homework 2-1

What is JVM?

Java Virtual Machine acts as a run-time engine to run Java applications.

- Class Loader
- Runtime Data Area
- Java Native Interface (JNI)
- Native Method Library

Class Loader

- Loading - load everything into JVM

  All the classes and dependences will be loaded.

  - Booststrap
    - load the Java packages like java.lang.* and java.net.*
  - Extension
    - load the extension classes like JDBC driver
  - Application
    - load classes in class path

  super relationship (Booststrap → Extension → Application)

- Linking - three stages (compile)
  - verify
    - verify the correctness of all .class files
  - prepare
    - load? all static fields, classes and Interface
  - resolve
    - replace symbolic reference to directly reference
- Initialization
  - initialize everything
    - static fields
    - final variables
    - static methods
    - static classes

Runtime Data Area

- Method Area - all the class level data
  - String/Integer constant pool
  - all classes
- Heap Area

    - new some object will be in the heap
- Stack Area
    - different kinds of stack frames
        - each stack frames contains local variables, operating stacks and some frame data
- PC Register
    - addresses of current execution JVM instruction
- Native Method Stack
    - create stacks and call other libraries written by C/C++ ...

Execution Engine

- Interpreter - byte code
- JIT Compiler - improve the performance of Interpreter
    - improve the performance of Java applications by compiling platform-neutral Java byte code into native machine code at run time.
- Garbage Collector

Native Method Interface(JNI) - Interface, to access NML

- bridge, make java invoke the Native Method Library

Native Method Library

- written by other languages (C/C++, the lower level languages)