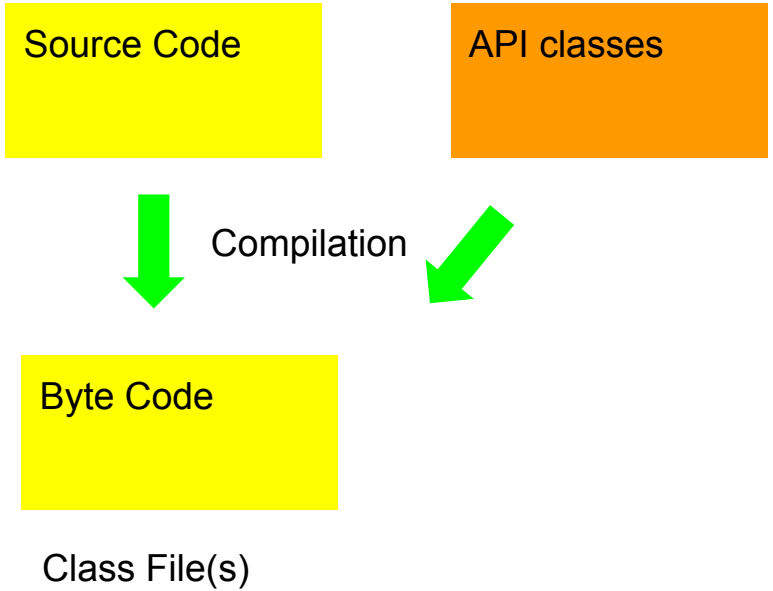
Tinker Academy

AP Computer Science Prep(Java DS & Algo)
Lecture 4 - Java Fundamentals 2 (DataTypes)

Winter 2015

Lecture 4 - Java Fundamentals 2



Lecture 4 - Java Fundamentals 2

Class File

- The .class file gets generated after compilation
- Precise description of a class, fields, methods
- code in methods stored as bytecode
- .class file interpreted by the JVM

Lecture 4 - Java Fundamentals 2

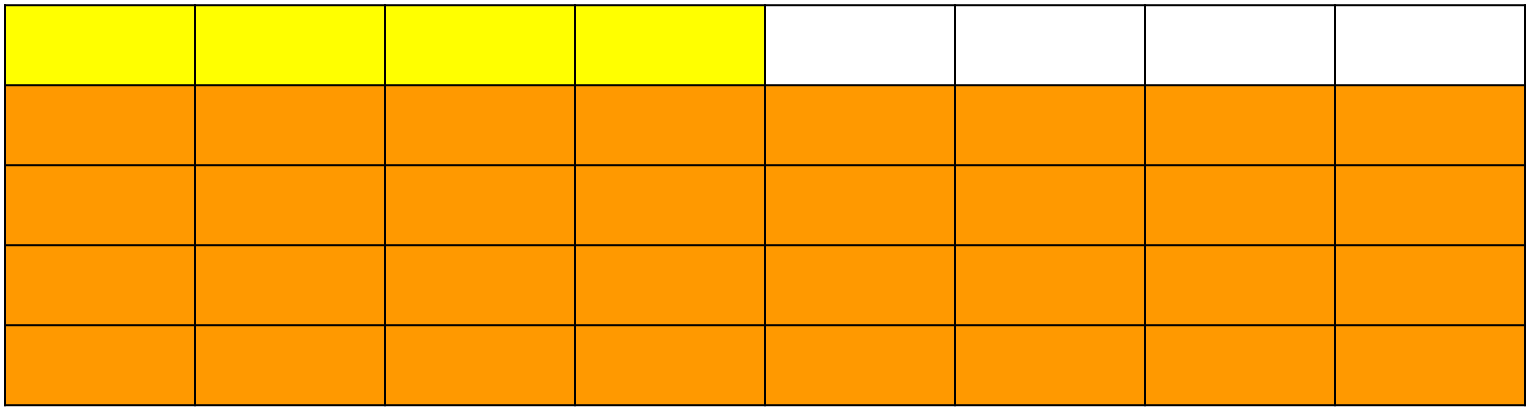
Class File

- When the program runs, the JVM executes bytecode starting from the main method
- JVM creates new instances of the object and stores the object in the computers memory

Lecture 4 - Java Fundamentals 2

Class File

- But how many memory locations should be used to store an object?
- Is the # of locations marked yellow enough?



Lecture 4 - Java Fundamentals 2

Class File

- If the JVM uses too much memory, the program will eventually run out of memory
- Too little and the program will not run correctly

Lecture 4 - Java Fundamentals 2

DataType

- Java and (all modern languages) solve this issue by the concept of data types
- Every value that requires memory also needs to have a corresponding datatype

00000000	00000000	00000000	00000001				
00000000	00000000	00000000	00000000	00000000	00000000	00000000	0000001

Integer Datatypes occupy 4 bytes
Long Datatypes occupy 8 bytes

Lecture 4 - Java Fundamentals 2

- Datatype indicates how much storage to be used for that value

00000000	00000000	00000000	00000001				
00000000	00000000	00000000	00000000	00000000	00000000	00000000	0000001

Integer Literals occupy 4 bytes
Long Integer Literals occupy 8 bytes

Lecture 4 - Java Fundamentals 2

DataTypes In Java

- Primitive DataTypes
- Reference DataTypes
- Special Datatype String

Primitive Types

Lecture 4 - Java Fundamentals 2

Primitive Data Types

Name
int
long
byte
short
float
double
char
boolean**

Lecture 4 - Java Fundamentals 2

Primitive Data Types

Name	Storage required							
int								
long								
byte								
short								
float								
double								
char								
boolean**								

* The JVM does not specify size only value ranges for datatypes

**The Java language does not specify size of boolean explicitly - conceptually its 1 byte

Lecture 4 - Java Fundamentals 2

Primitive DataTypes

- Well defined set
- Same across ALL machine architectures
- Supports efficiency by using up just enough locations in memory and no more

Class Activity