GUI Study Guide: Exam 1

1. JDK (Java Development kit) vs. JRE (Java Runtime Environment)
   1. JDK includes compiler. Compiles code. (This is NOT the IDE).
   2. JRE runs compiled Java programs (it is a virtual machine).
2. Java programming process:
   1. JDK
      1. Edit .java files
      2. Compile (create .class files)
   2. JRE
      1. Load .class files
      2. Verify program can run
      3. Interpret
3. JavaDoc: include at the start of every:
   1. Class
   2. Constructor
   3. Method
   4. Class variables/constants
   5. How to start? /\*\*
4. Java.lang is the only (?) class that is automatically imported. Includes:
   1. Math
   2. System
   3. String
5. Access Modifiers:
   1. Private – only class members can access
   2. Public – anyone can access
   3. Protected – classes that inherit from it can use
   4. Package private – any class w/in same package can access
6. Static – “belongs to the class, not the instance of the class.”
7. In Java, we only have 1 top-level public class.
8. Two types of data:
   1. Primitive
      1. boolean (1 bits)
      2. char (16 bits)
      3. double (64 bits)
      4. float (32 bits)
      5. int (32 bits)
      6. short (16 bits)
      7. long (64 bits)
      8. byte (8 bits)
   2. Reference
      1. Anything not primitive
9. Interning
   1. If string1 = “hello”, and string2 = “hello”, java will automatically set string2 to the address of string1.
   2. Or, if string2 = new String(“hello”), you can do it manually via string2 = string2.intern()
   3. Interning is possible only because strings are immutable
10. In order to create a JavaFX program, the main class must extend application
11. In general: public static void main(String[] args) { }
12. For JavaFX – start method: public void start(Stage primaryStage)
13. Dialog boxes:
    1. Alert (types: None, Information, Error, Waring, Confirmation)
    2. Choice (dropdown menu)
    3. TextInputDialog
    4. Methods: setTitle(), setContentText(), setHeaderText()
14. Remember: getStyleSheets.**add()**
15. Static Import:
    1. Code: import static java.lang.Math.PI;
    2. Now we can use PI instead of Math.PI.
16. Q: “What does ‘static’ mean?”

A: “it belongs to the class, not an instance of the class”

Use the classname to reference static method, not the instance of the class.

1. Components of a JavaFX application
   1. Stage (only one of these typically)
   2. Scene (inside stage, one at a time)
   3. Pane (way to divide scene into pieces)
   4. Node (things on stage or pane, e.g. another pane, shape, imageView, button)
2. Event driven programming
   1. setOn\_\_\_\_\_()
   2. setOnAction(button, checkbox, hyperlink)
3. 5 ways to create event handlers:
   1. Anonymous inner class
   2. Implement the interface
   3. Create Eventhandler instance
   4. Lambda expression
   5. Method reference
4. Enumerated types:
   1. A type of class (cause its Java)
   2. Syntax:

<access modifier> enum <Name> {

MONDAY,

TUESDAY,

WEDNESDAY

}

* 1. Note: the value of MONDAY is MONDAY.
  2. Ordinal(enum) returns index of enumerated value. MONDAY has ordinal value of 0.
  3. Equals() returns Boolean
  4. Equals() vs == :
     1. Equals() checks characters
     2. == checks object
  5. Compareto() returns int (diff btw ordinal values)
  6. Constructor for enum type:
     1. Private by default.
     2. Ex. Days(){

}

1. For each loop:
   1. Syntax:

For(<type> ident. : <list>){

}

Ex:

For(enumMonth mon: enumMonth.values()){

}

1. Wrapper class
   1. Integer (holds an int)
   2. Float (holds a float)
   3. Boolean (holds a boolean)
   4. Double (holds a double)
   5. Character (holds a char)
   6. Short (holds a short)
   7. Long (holds a long)
   8. Byte (holds a byte)
2. Autoboxing (and Auto-unboxing)
   1. Autoboxing – puts a primitive type into the object box. (puts an int into an Interger)
      1. Integer x = 5 //Autoboxing
      2. System.out.println(“Unboxing: ” + x) //Autounboxing
   2. Property of wrapper classes -
   3. Primitive -> integer
3. Event Handler