Intro to GUI’s – Study Guide:

|  |  |
| --- | --- |
| JDK:  JRE: | Java Development Kit  Java Runtime Environment |
| Java Program Process: | Edit  Compile  Load  Verify  Interpret |
| Command Line compile/run commands | javac – compiles (javac HelloWorld.java)  java – runs (java HelloWorld) |
| JavaDoc | /\*\*  Comment  \*/ |
| Classes |  |
| Methods |  |
| Data Types   * Primitive * Reference | boolean,  byte,  char,  short,  int,  long,  float,  double  Anything not primitive |
| Immutable vs mutable | immutable – cannot be modified  mutable – can be modified |
| Wrapper Classes |  |
| Interning | if string1 = “hello” and string2 = “hello”, java will set string2 = address of string1.  Can also use string2.intern() |
| Static | Belongs to class, not an instance of the class  Use classname to reference static method, not the instance of the class  \*\* Use private default constructor  Popular static classes: Math and System |
| Static Import | import static java.lang.Math.PI;  now can use PI instead of Math.PI |
| Input/Outpupt | Scanner  System.in |
| JavaFX Dialogs | Alert (types: None, Information, Error, Warning, Conformation)  Choice (dropdown menu)  TextInputDialog |
| Comparing strings   * compareTo * equals() * == |  |
| To create a javaFX program, main class must extend: \_\_\_\_\_\_\_\_\_\_\_\_\_ | Application |
| General syntax for creating java program | public static void main(String[] args){} |
| For javaFX start method | public void start(Stage primaryStage){} |
| JavaFX components | Stage  Scene (inside stage, one displayed at a time)  Pane (compartmentalize scene)  Node (hbox, vbox, etc.) |
| Event Handling | setOnXXXX()  setOnAction() |
| Creating Event Handler Objects | Anonymous Inner Class  Implement interface  EventHandler object identifier  Method Reference  Event Handler object called handler |
| ImageView & Image |  |
| Formal vs Actual parameters |  |
|  |  |
|  |  |
| Autoboxing:  Autounboxing: | puts a primitive type into the object box. (puts an int into an Interger)  Integer x = 5 //Autoboxing  System.out.println(“Unboxing: ” + x) //Autounboxing |
| Enum Type |  |
|  |  |
|  |  |
| Property  Property Binding  property1.bind(property2) | Holds a value we can use. Is an object  Two properties bound together. When one changes, the other one changes  When property2 changes, property1 changes  <target>.bind(<source>) |
| Media   * 3 classes * Important method: | 1. Media -> Apply file 2. MediaPlayer -> add media to media player (includes controls) 3. MediaVeiw -> view video  * setOnReady(Runable run) – runs when everything is ready to go |
| Midi   * 3 classes | 1. MidiChannel 2. MidiSystem 3. Synthesizer |
| FileChooser   * Methods: | setInitialDirectory()  setInitialFileName()  setSelectedExtensionFilter()  showOpenDialog()  showSaveDialog() |
| Layouts   * HBox * VBox * BorderPane * FlowPane * TilePane * StackPane * GridPane * Note: SplitPane | * horizontal layout * vertical layout * top, bottom, left, right, center * wraps nodes at end of line (set vgap/hgap instead of spacing) * Like flow pane, but all nodes have same amount of space. * Stack nodes on “z-axis” * Like Excel * SplitPane is a control, not a pane |
| Checkboxes/radiobuttons |  |
| Inheritance |  |
| Polymorphism |  |
| Interfaces   * Java only has \_\_\_\_\_\_\_\_\_ inheritance * What is the purpose of having a default method? * Functional Interface * Marker Interface * Cloneable interface: * Clone method is part of an * Deep copy vs shallow copy: Java uses? * Clone method is part of an \_\_\_\_\_ * CompareTo(T object) | Interface: a characteristic that we can apply to a class. It is a class where every method is abstract (no definitions.   * Single inheritance * Code maintenence * Has exactly 1 method * Has 0 methods * Allows you to Clone an object and is a marker interface * object * Shallow copy only. (Can do deep copy with multiple shallow copies) * Object * Functional interface. Override to give desired behavior |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Abstract methods/classes |  |
| Method overriding | late binding vs. early binding]  Java ALWAYS uses late binding |
| Object class | toString()  equals() |
| instanceof operator |  |
| JCF | Java Collections framework |
| Threads   * Concurrency * Parallelism * Overhead   Lifecycle of threads:  Threads must implement:  Synchronized Block  Timer   * TimerTask (implements runnable) * Timer   Application Thread (aka: \_\_\_\_\_\_\_\_\_\_)  Q: What is the command to update the GUI/application thread? | * *Appearance* of multiple things happening at once * *Actual* occurrence of multiple things happening at once * Extra time spent switching tasks in concurrency   See diagram below  runnable  Synchronized(object monitor) {  //usually ‘this’ is the monitor  <critical section> // know this term  }   * Inside timerTask, use this.cancel. outside, use timer.cancel * Event thread, JavaFX thread, Main Thread   A: Platform.runlater() (use whenever you’re on a separate thread and want to update the GUI) |
| Generic Classes   * Example | ArrayList<String> or ArrayList<ColorPanel>  A generic class (like ArrayList<T>) does not have type T specified. |
| Callbacks | A method that is passed as an argument to another method to be executed at a later time  Are asynchronous (don’t know order they happen in) |
| Delegates | Delegator  The same method call on the delegator can execute the method on different delegates |
| Scenebuilder   * initialize | FMXLLoader.load()   * initialize gets called after constructor. Lets us know that all FXML objects have been built |
| StringBuffer  StringBuilder | similar to String, but:   * is mutable   StringBuffer is thread safe (only one thread can access at a time)   * is synchronized   similar to StringBuffer, but:   * is not thread safe * is not synchronized |
| Exception Handling   * Process * Runtime Exception | 1. Exception created 2. Throw the exception 3. Catch the exception  * Special type of exception where you don’t have to handle it. |
| \*\* Look at Delegate demo code \*\* |  |
|  |  |
| JSON | Javascript Object Notation  3rd party library: GSON |
| To get network data, use an \_\_\_\_\_\_ | API (Application Programming Interface) |
| TableView, TableColumn |  |

Lifecycle of threads:

Running

Blocked

Finished

Ready to execute

New Thread created