Java Curriculum

Setting up Our Development Environment

- Installing Java
- Installing Intellij

Intellij

- Keyboard Shortcuts
    - Alt+Insert: Generate code
    - Alt+Enter: Pull up error/warning menu
    - Shift+”: when we have text highlighted, this surrounds it in quotes
    - sout+Tab: System.out.prinln()
        - Note: type the letters sout then press Tab
    - psvm+Tab: public static void main(String[] args) {}
        - Note: type the letters psvm then press Tab
    - Ctrl+b:

Variables & Data Types

- Variables
    - Declaration
    - Manipulation
    - Casting
- Data Types
    - String
    - Int
    - Float
    - Boolean
    - Char
    - Primitive vs reference types

String Manipulation

- String class
- Concatenation
- String methods (not an exhaustive list)
    - Length
    - IndexOf
    - charAt
    - CompareTo

Basic Data Structures

- Array
    - Definition: A group of elements stored in contiguous memory locations
    - Adding or removing: must make a new array
    - Grabbing an element: very fast operation as that element can be referenced directly
- LinkedList
    - Definition: A group of elements stored in non-contiguous memory locations
    - Adding or removing: computationally cheap because you can just change the pointers around the element you want to add
    - Grabbing an element: computationally expensive because you have to traverse the linked list until you get the element you want
- HashMap
    - Definition: Stores data in a key->value pairing
- Set
    - When to use different types of sets
- Stack
- Queue
    - Queue vs Deque Java Classes
        - Queue: Can only insert elements into one end and grab them off the other end
        - Deque: Short for “double sided queue” and you can add and remove elements from either end

Conditionals and Comparison Operators

- Comparison Operators
- If statements
- Else statements
- Else if statements
- Chaining conditional statements
- Nesting conditional statements
- Switch statements

Loops

- For loops
    - Use when you know how many elements you are looping over
- While loops
    - Use when you don’t know how many elements you are looping over
- Do-while loops
    - Use when you want the code inside the loop to be executed at least once

Gathering User Input

- Scanner Class
    - Used to read data from a specified location
        - Can read from the terminal to gather user input
    - The .next() methods specify how to gather an input
    - The .hasnext() methods return true if there is more input to read either in general or of a specific type
    - .close() tells Java that the scanner object is done and to get rid of it
        - Not a huge deal if you don't do this
        - Speeds up code slightly by closing it since Java can garbage collect it more easily
    - Make sure to read the next line after gathering an int so that you can gather more things after it!

Object-Oriented Programming

- Classes vs. objects
- Creating a class
    - Attributes
    - Constructors
    - Getters and setters
    - Static vs class methods
    - private/protected/public methods

Access Levels

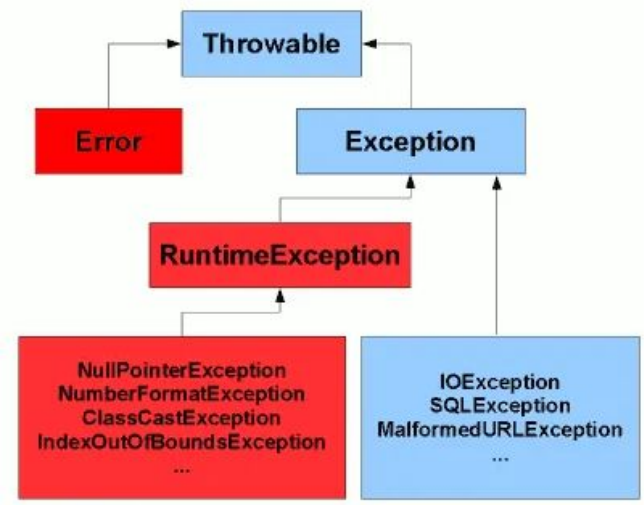
Modifier	Class	Package	Subclass	World
`public`	Y	Y	Y	Y
`protected`	Y	Y	Y	N
no modifier	Y	Y	N	N
`private`	Y	N	N	N

    - equals() method
        - Used to compare objects. This includes when other data structures like HashSets need to see if two objects are the same
        - By default, this just compares memory location
    - hashcode() method
        - Used to turn an object into an integer
        - Need to implement for HashSets, HashMaps, and other Hash objects to work properly

- Inheritance
    - Interfaces
        - Used to enforce standardization of classes and assist with polymorphism
    - Abstract classes
        - Like an interface. The difference is you can implement methods in it
    - Polymorphism
        - Objects can be more than one type through inheritance
- Enums
    - Special "class" that represents a group of constants (unchangeable variables, like final variables).

Error Handling

- Java Exception: Java word for error
- Checked vs unchecked exceptions



    - Checked (Blue): Exceptions that are checked at compile time
    - Unchecked (Red): Exceptions that aren't checked until runtime
    - When to use checked vs unchecked exceptions
        - From the Oracle Java Documentation: "If a client can reasonably be expected to recover from an exception, make it a checked exception. If a client cannot do anything to recover from the exception, make it an unchecked exception."
        - Basically, if the error represents an error outside the control of the program, throw a checked exception. If the exception reflects some error inside the program logic itself, use an unchecked exception
    - Good article on checked vs unchecked exceptions
- try/catch/finally
- Throwing exceptions

File I/O

- Creating files
- Reading files
- Writing to files
- Closing files
- Renaming files
- Removing files

Spring Basics

- Spring projects
    - Spring Boot: generates a spring project with minimal configuration
    - Spring Framework: Core Spring code
    - Spring Data:
- Getting your first spring project up and running
- Inversion of Control (IoC)
    - Principle in software engineering which transfers the control of objects or portions of a program to a container or framework
- Dependency injection
    - Pattern we can use to implement IoC, where the control being inverted is setting an object's dependencies.
- Spring annotations
    - Perform some functionality in the backend