Data Structures and Algorithms

Agenda

- Data structure libraries
- Stack applications
    - Solve Expressions
    - Parenthesis Balancing
- Interview Questions

VS Code -- Java Compilation

- javac -d . StackMain.java
- java com.sunbeam.StackMain

Data structure Implementations

User defined implementations

- Stack -- LIFO
- Linear Queue -- FIFO
- Circular Queue -- FIFO

Programming Languages - Built-in Libraries

Java -- Collection Framework

- Package: java.util
    - Interfaces: Collection, Set, List, Queue, Map
    - Classes: ArrayList, LinkedList, HashSet, HashMap, ...
    - Helper classes: Collections, Arrays

Stack class

~~~
```Java
Stack<String> s = new Stack<String>();
s.push("A");
s.push("B");
s.push("C");
System.out.println("Topmost Element: " + s.peek()); // C
while(!s.isEmpty()) {
    System.out.println("Popped Element: " + s.pop());   // C, B, A
}
```
~~~

Queue interface/LinkedList class

- Queue<> operations:
    - To add element: offer()
    - To delete element: poll()
    - To get topmost element: peek()

```
//Queue<String> q = new LinkedList<String>();
LinkedList<String> q = new LinkedList<String>();
q.offer("A");   // like push()
q.offer("B");   // like push()
q.offer("C");   // like push()
System.out.println("Topmost Element: " + q.peek()); // A
while(!q.isEmpty()) {
    System.out.println("Popped Element: " + q.poll()); // A,B,C
}
```

C++ -- STL

- Container classes: vector, list, stack, queue, map, multimap, set, ...

Python -- Collections

- Collections
    - List: `[ ... ]`
    - Dictionary: `{ ... }`
    - Tuple: `( ... )`

Expressions

- Infix: "12 + 34 * (999 - 990) / 2"
    - Programmatically need to separate the tokens and process them.

    ```
    String[] infix = infixString.split(" ");
    // ...
    ```