Week 11 - LAQ's

Instructions

Explain legacy classes in Java.

Legacy classes in Java refer to the classes that were part of the original version of Java (Java 1.0) and were later retrofitted to work with the Java Collections Framework introduced in Java 2 (Java 1.2). These classes are still available for use, but they are considered outdated compared to the newer collection classes. Legacy classes primarily include:

1. Vector:

- A resizable array implementation of the List interface. It can grow and shrink dynamically as elements are added or removed.
- It is synchronized, meaning it is thread-safe, but this can lead to performance overhead in single-threaded scenarios.
- Example usage:

```
Vector<String> vector = new Vector<>();
vector.add("Element 1");
vector.add("Element 2");
```

2. Stack:

- A subclass of Vector that implements a last-in, first-out (LIFO) stack of objects. It provides methods like push(), pop(), and peek().
- Example usage:

```
Stack<Integer> stack = new Stack<>();
```

```
stack.push(1);

stack.push(2);

int top = stack.pop(); // Returns 2
```

3. Hashtable:

- A collection that maps keys to values. It is similar to HashMap, but it is synchronized and does not allow null keys or values.
- Example usage:

```
Hashtable<String, Integer> hashtable = new Hashtable<>();

hashtable.put("Key1", 1);

hashtable.put("Key2", 2);
```

4. Properties:

- A subclass of Hashtable that represents a persistent set of properties that can be saved to or loaded from a stream. It is often used for configuration settings.
- Example usage:

```
Properties properties = new Properties();

properties.setProperty("username", "admin");

properties.setProperty("password", "password123");
```

Characteristics of Legacy Classes

- Synchronization: Most legacy classes are synchronized, which means they are thread-safe. However, this can lead to performance issues in single-threaded applications.
- Enumeration Interface: Legacy classes use the Enumeration interface for iterating over elements, which is considered less powerful

than the Iterator interface introduced in the Collections Framework. The Enumeration interface does not support removing elements during iteration.

- Compatibility: Legacy classes are still part of the Java API for backward compatibility. They can be used in existing applications that were built before the introduction of the Collections Framework.

Transition to the Collections Framework

With the introduction of the Java Collections Framework, newer classes such as ArrayList, HashMap, and HashSet provide more flexible and efficient ways to handle collections of objects. These newer classes support generics, provide better performance, and offer more powerful iteration capabilities through the Iterator interface.