

Java 16

Links :-

1. <http://openjdk.java.net/projects/jdk/16/>
2. <https://cr.openjdk.java.net/~iris/se/16/latestSpec/apidiffs/overview-summary.html> (API differences)
3. <https://www.oracle.com/java/technologies/javase/16-relnote-issues.html#NewFeature>

Stream.toList():-

1. Stream introduced toList() method to collect the result of stream processing.
2. Initially we are rely on Collectors API's toList() method. But now we can overcome from this boilerplate code.
3. For Ex:-

```
public class ToList {  
  
    public static void main(String[] args) {  
        // Old approach to get a list of output of stream processing  
        List<Car> redCars1 = ResourceLoader.getCars().stream()  
            .filter(c -> c.getColor().equalsIgnoreCase( anotherString: "red"))  
            .collect(Collectors.toList());  
  
        // New approach to get a list of output of stream processing  
        List<Car> redCars2 = ResourceLoader.getCars().stream()  
            .filter(c -> c.getColor().equalsIgnoreCase( anotherString: "red"))  
            .toList();  
  
        System.out.println("List1 size:- " + redCars1.size());  
        System.out.println("List2 size:- " + redCars2.size());  
    }  
}
```

Record Enhancements:-

1. Record type is permanent feature from java-16. It is no more preview feature.

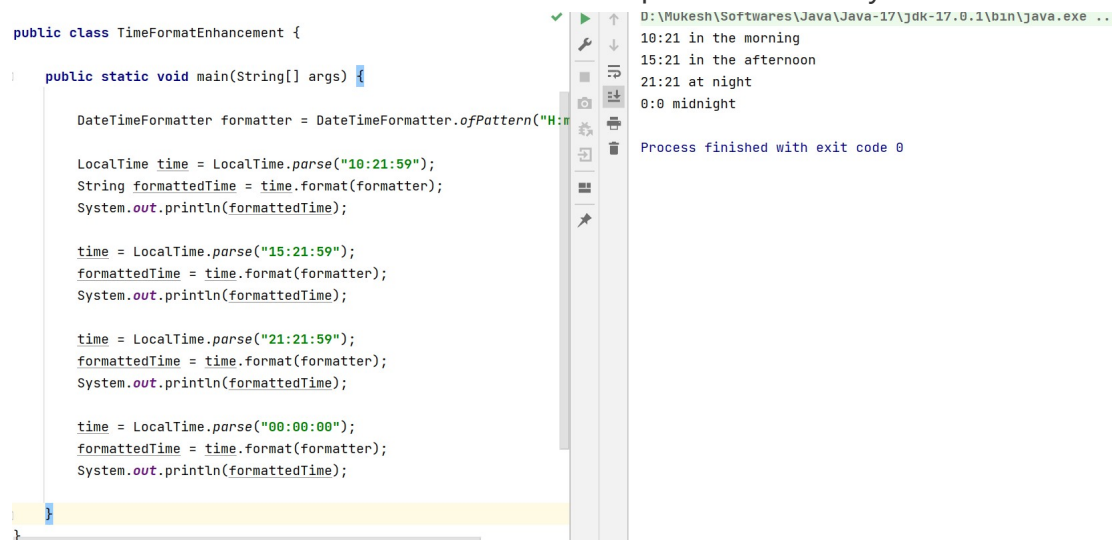
2. There is one enhancement in it. We can use records in inner classes as well. For ex:-

```
public class RecordMain {  
  
    static class VehicalUses {  
        public Vehical v = new Vehical( type: "Truck", manufacturedAt: 1971);  
    }  
  
    public static void main(String[] args) {  
        RecordMain.VehicalUses vehicalUses = new RecordMain.VehicalUses();  
        System.out.println(vehicalUses.v);  
    }  
}
```

3. Vehicle is a record type in above example.

Date Time Format Enhancement:-

1. A new format is added to specify period of the day.
2. We can add "B" in format to know about the period of the day. For ex:-



```
public class TimeFormatEnhancement {  
    public static void main(String[] args) {  
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("H:mm:ss B");  
  
        LocalDateTime time = LocalDateTime.parse("10:21:59");  
        String formattedTime = time.format(formatter);  
        System.out.println(formattedTime);  
  
        time = LocalDateTime.parse("15:21:59");  
        formattedTime = time.format(formatter);  
        System.out.println(formattedTime);  
  
        time = LocalDateTime.parse("21:21:59");  
        formattedTime = time.format(formatter);  
        System.out.println(formattedTime);  
  
        time = LocalDateTime.parse("00:00:00");  
        formattedTime = time.format(formatter);  
        System.out.println(formattedTime);  
    }  
}
```

Output:

```
D:\Mukesh\Softwares\Java\Java-17\jdk-17.0.1\bin\java.exe ...  
10:21 in the morning  
15:21 in the afternoon  
21:21 at night  
0:0 midnight  
Process finished with exit code 0
```

3. "B" pattern is responsible of it.

Sealed and non-sealed Classes/Interfaces enhancement:-

1. sealed and non-sealed are keywords now.
2. Restricted the ability to create local classes that are sub classes of sealed classes. Same as inability to create anonymous classes of sealed classes.
- 3.

Pattern Matching For instanceof:-

1. Good news is, this feature is no more a preview feature.
2. It is now permanent feature of java and can be used at prod.

Default Method Invocation:-

1. From java-16 and onwards, we can call a default method of interfaces with the help of Proxy classes.
2. Proxy class is part of reflection framework.