**What is parallel stream**

privatelongcountPrimes(intmax) {

returnrange(1, max).parallel().filter(this::isPrime).count();

}

We have the method countPrimes that counts the number of prime numbers between 1 and our max. A stream of numbers is created by a range method. Thestream is then switched to parallel mode; numbers that are not primes are filteredout and the remaining numbers are counted. Parallelization is just a matter of calling the parallel()  method. When we do that,the stream is split into multiple chunks, with each chunk processed independently and with the result summarized at the end.

**What is flatmap**

Stream flatMap(Function mapper) returns a stream consisting of the results of replacing each element of this stream with the contents of a mapped stream produced by applying the provided mapping function to each element. Stream flatMap(Function mapper) is an intermediate operation. These operations are always lazy. Intermediate operations are invoked on a Stream instance and after they finish their processing, they give a Stream instance as output.

Syntax: &lt;R&gt; Stream&lt;R&gt; flatMap(Function&lt;? Super T, ? extends Stream&lt;?

extends R&gt;&gt; mapper

Eg:

import java.util.\*;

import java.util.stream.Stream;

class GFG {

    // Driver code

    public static void main(String[] args)

    {

        // Creating a List of Strings

        List&lt;String&gt; list = Arrays.asList(&quot;5.6&quot;, &quot;7.4&quot;, &quot;4&quot;,  &quot;1&quot;, &quot;2.3&quot;);

        // Using Stream flatMap(Function mapper)

        list.stream().flatMap(num -&gt; Stream.of(num)).

                         forEach(System.out::println);

    }

}

Output:-

5.6

7.4

4

1

2.3

**Peek() method**

The java.util.Stack.peek() method in Java is used to retrieve or fetch the first element of the Stack or the element present at the top of the Stack. The element retrieved does not get deleted or removed from the Stack.

Eg: Stack&lt;String&gt; STACK = newStack&lt;String&gt;();

STACK.push(&quot;Welcome&quot;);

System.out.println(&quot;The element at the top of the&quot;  + &quot; stack is: &quot; + STACK.peek())

**How can we create a user defined unchecked exception**

<https://makeinjava.com/create-custom-user-defined-unchecked-exception-java-example/>

* Create a user defined exception in java.
* Create a custom exception, by extending unchecked exceptions.
  1. We will create CustomArithmeticException by extending ArithmeticException class
  2. We will create CustomNullPointerException by extending NullPointerException class.

**Can we catch the exception using catch block of throwable**

[Throwable](https://docs.oracle.com/javase/8/docs/api/java/lang/Throwable.html) is the superclass of all exceptions and errors. You can use it in a catch clause, but you should never do it.

If you use Throwablein a catch clause, it will not only catch all exceptions; it will also catch all errors.

Errors are thrown by the JVM to indicate serious problems that are not intended to be handled by an application.

Typical examples for that are the [OutOfMemoryError](https://docs.oracle.com/javase/8/docs/api/java/lang/OutOfMemoryError.html" \t "_blank) or the [StackOverflowError](https://docs.oracle.com/javase/8/docs/api/java/lang/StackOverflowError.html" \t "_blank). Both are caused by situations that are outside of the control of the application and can’t be handled.

So, better don’t catch a Throwable unless you’re absolutely sure that you’re in an exceptional situation in which you’re able or required to handle an error.

**Min() and Max() in timer class**

We use the min() and max() methods to find the min &amp; max value in streams.These methods are used for finding min &amp; max values in different types of streams such as stream of chars, strings, dates.

List<LocalDate>dates =Arrays.asList(LocalDate.now(),

LocalDate.now().minusDays(9),

LocalDate.now().minusMonths(2),

LocalDate.now().minusDays(30));

//getting max date

LocalDatemaxdate=dates.stream()

.max(Comparator.comparing(LocalDate::toEpochDay))

.get();

//getting min date

LocalDatemindate=dates.stream()

.min(Comparator.comparing(LocalDate::toEpochDay))

.get();

**Distinct Method in java**

The Stream API provides the distinct() method that returns different elements of a list based on the equals() method of the Object class.

distinct() uses hashCode() and equals() methods to get distinct elements.

List&lt;String&gt; list = Arrays.asList(&quot;AA&quot;, &quot;BB&quot;, &quot;CC&quot;, &quot;BB&quot;, &quot;CC&quot;, &quot;AA&quot;,

&quot;AA&quot;);

long l = list.stream().distinct().count();

System.out.println(&quot;No. of distinct elements:&quot;+l);

String output = list.stream().distinct().collect(Collectors.joining(&quot;,&quot;));

System.out.println(output);