Threading

Extending the Thread class

Threads can be created by extending the Thread class.

Creating Threads Using Lambda Expressions

Threads can be created using lambda expressions.

Purpose of Threads

Threads are used in Java to improve performance for processes that can be split into code that can run concurrently.

Java Thread Lifecycle

A Java thread can, throughout its lifecycle, be in New, Running/Active, Blocked, Waiting, or Terminated/Joined states.

Thread Communication in Java

Threads can communicate with each other in Java by polling the state of a shared resource.

Java Thread Lifecycle Methods

“A programmer can control Java thread lifecycle states using the following methods:

- .start()
- .sleep()
- .join()
- .isAlive() “

Threads Communication Methods In Java

Threads can communicate with each other in Java using the `.wait()` , `.notify()` and `.notifyAll()` methods.

Synchronized in Java

In Java, the `synchronized` keyword is used to ensure that methods in two different threads do not access a shared resource at the same time.

Thread

A thread is a part of a program that follows a path of execution independently. Multiple threads can run at the same time (concurrently).

Race Conditions in Java Threads

A race condition occurs when a multi-threaded program produces unexpected behavior due to the timing or interleaving of the execution of threads.