Platform

13 December 2021 10:49

Concurrency: It is a mechanism that enables a program to handle multiple tasks simultaneously. Concurrency is commonly used for supporting

1. **Asynchrony** - It allows the caller of a procedure to resume execution before that procedure returns to increase responsiveness of the program.
2. **Parallelism** - It allows different iterations of a long running loop to execute on separate processors/cores available on the hardware to increase performance of the program.

Thread: It is the basic unit of concurrency which can execute its own routine along with other threads of its program and share the program data (not local to its routine) with those threads.
In Java an instance of java.lang.Thread class is used to start and schedule a thread whose routine is specified by an implementation of java.lang.*Runnable* interface.

Persistence: It is a mechanism that enables a program to retain its state (in-memory data) across its multiple execution cycles.

File: It is a platform managed resource in which bits of data can be saved so that it can be loaded again at any time in the future.
Java runtime library exposes file related operations using

1. **java.io.FileInputStream/FileOutputStream** for sequentially reading/writing an array of bytes from/to a file.
2. **java.nio.channels.FileChannel** for accessing file data by mapping its region to an object of *java.nio.ByteBuffer* which provides get/put method for reading/writing a byte at any given position within that region.

Object Serialization: It is a mechanism of converting the entire state of an object into a stream (series) of bytes from which it can be reconstructed through a process called deserialization. It is commonly used for persisting objects into files or to marshal (transport) objects across their process boundaries.
Java runtime library provides java.io.ObjectOutputStream/ObjectInputStream for serializing/deserializing an object with following characteristics

1. Its class must inherit from java.io.*Serializable* type which is an empty interface.
2. It must not contain a reference to a non-serializable object through an instance field defined without *transient* modifier by its class.
3. Its class should define a static final long serialVersionUID field which should be initialized with a different value whenever this

class changes the set of its instance field definitions.