

Java Keywords

Key Concepts

Key Concepts #2

OOPS in java

Java Collections#1

Java Collections #2

Exceptions #1

Exceptions #2

Threads#1

Threads#2

InnerClass

More InnerClass

Serialization

Immutable Class

Cloning in Java

Garbage Collection

JSP #1

JSP #2

Programming Q#1

Programming Q#2

Programming Q#3

Threads in java Interview Questions

Multithreading or Concurrency is one of the popular topic in java interview questions. Large scale applications such as Banking, Big data processing, scaling for millions of users often uses multithreading and async functionality pretty heavily

Q1) If a class has a synchronized method and non-synchronized method, can multiple threads execute the non-synchronized methods?

Ans)Yes. If a class has a synchronized and non-synchronized methods, multiple threads can access the non-synchronixed methods.

Q2) If a thread goes to sleep does it hold the lock?

Ans) Yes when a thread goes to sleep, invoked by Thread.sleep(xx) it holds a lock.

Q3) Can a thread hold multiple locks at the same time?

Ans) Yes. A thread can hold multiple locks at the same time. Once a thread acquires a lock and enters into the synchronized method / block, it may call another synchronized method and acquire a lock on another object.

Q4) Can a thread call multiple synchronized methods on the object of which it hold the lock?

Ans) Yes. Once a thread acquires a lock in some object, it may call any other synchronized method of that same object using the lock that it already holds. The locks are Reentrantant.

Q5) Can static methods be synchronized?

Ans) Yes. Static methods are class methods and have only one copy of static data for the class, only one lock for the entire class is required. Every class in java is represented by java.lang.Class instance. The lock on this instance is used to synchronize the static methods.

Q6) Can two threads call two different static synchronized methods of the same class?

Ans) No. The static synchronized methods of the same class always block each other as only one lock per class exists. So no two static synchronized methods can execute at the same time.

Q7) Does a static synchronized method block a non-static synchronized method?

Ans)No, the thread executing the static synchronized method holds a lock on the class and the thread executing the non-satic synchronized method holds the lock on the object on which the method has been called, these two locks are different and these threads do not block each other.

Q8) Once a thread has been started can it be started again?

Ans) No. Only a thread can be started only once in its lifetime. If you try starting a thread which has been already started once an IllegalStateException is thrown, which is a runtime exception. A thread in runnable state or a dead thread can not be restarted.

Q9) When does deadlock occur and how to avoid it?

Ans) When a locked object tries to access a locked object which is trying to access the first locked object. When the threads are waiting for each other to release the lock on a particular object, deadlock occurs .

Q10) What is a better way of creating multithreaded application? Extending Thread class or implementing Runnable?

Ans) If a class is made to extend the thread class to have a multithreaded application then this subclass of Thread can not extend any other class and the required application will have to be added to this class as it can not be inherited from any other class. If a class is made to implement Runnable interface, then the class can extend other class or implement other interface.

Q11) Can the start() method of the Thread class be overridden? If yes should it be overridden?

Ans) Yes the start() method can be overridden. But it should not be overridden as itâ€™s implementation in thread class has the code to create a new executable thread and is specialised.

Q12) What are the methods of the thread class used to schedule the threads?

Ans) The methods are as follows:

- public static void sleep(long millis) throws InterruptedException
- public static void yield()
- public final void join() throws InterruptedException
- public final void setPriority(int priority)
- public final void wait() throws InterruptedException
- public final void notify()
- public final void notifyAll()

Q13) Which thread related methods are available in Object class?

Ans) The methods are:

- public final void wait() throws Interrupted exception
- public final void notify()
- public final void notifyAll()

Q14) Which thread related methods are available in Thread class?

Ans) Methods which are mainly used :

- public static void sleep(long millis) throws Interrupted exception
- public static void yield() public final void join() throws Interrupted exception
- public final void setPriority(int priority)
- public void start()
- public void interrupt()
- public final void join()
- public void run()
- public void resume()

Q15) List the methods which when called the thread does not release the locks held?

Ans) Following are the methods.

- join()
- sleep()
- yield()

Q16) List the methods which when called on the object the thread releases the locks held on that object?

Ans) Following are the methods.

- notify()
- notifyAll()
- wait()

Q17) Does each thread has its own thread stack?

Ans) Yes each thread has its own call stack. For eg

```
Thread t1 = new Thread();
Thread t2 = new Thread();
Thread t3 = t1;
```

In the above example t1 and t3 will have the same stack and t2 will have its own independent stack.

Q18) What is thread starvation?

Ans)In a multi-threaded environment thread starvation occurs if a low priority thread is not able to run or get a lock on the resoruce because of presence of many high priority threads. This is mainly possible by setting thread priorities inappropriately.

Q19) What is threadLocal variable?

Ans) ThreadLocal is a class. If a variable is declared as threadLocal then each thread will have a its own copy of variable and would not interfere with the other's thread copy. Typical scenario to use this would be giving JDBC connection to each thread so that there is no conflict.

ThreadLocal class by JAVA API

```
public class ThreadLocal {
    public Object get();
    public void set(Object newValue);
    public Object initialValue();
}
```

Implementation of ThreadLocal

```
public class ConnectionDispenser {
    private static class ThreadLocalConnection extends ThreadLocal {
        public Object initialValue() {
            return DriverManager
                .getConnection(ConfigurationSingleton.getDbUrl());
        }
    }
    private static ThreadLocalConnection conn = new ThreadLocalConnection();
    public static Connection getConnection() {
        return (Connection) conn.get();
    }
}
```

If you reach so far, check [More Questions on Thread](#)

Recommend Reading

- [How java manages Memory?](#)
- [Why is it advised to use hashCode and equals together?](#)
- [Comparable and Comparator in java](#)
- [How to create Singleton class in Java?](#)
- [Difference between equals and ==?](#)
- [When to use abstract class over interface?](#)
- [Cap Theorem in Details](#)
- [Why is it recommended to use Immutable objects in Java](#)

1 Comment

www.java-questions.com

Login

Recommend

Share

Sort by Best

Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS

Akash Pal

2 years ago

15, 16)

Methods which release lock: wait(), notify() and notifyAll()

Methods which does not release lock: yield(), join(), sleep()

^ | v · Reply · Share

Subscribe

Add Disqus to your site

Privacy

DISQUS

Recommended Reading

- [Java 8 Lambda Explained](#)
- [How java manages Memory?](#)
- [When to use hashCode and equals?](#)
- [Comparable and Comparator in java](#)
- [How to create a Singleton class in Java?](#)
- [When to use abstract class over interface?](#)

READY TO DO AN INTERNSHIP IN ASIA?

CUSTOMIZED INTERNSHIP PLACEMENT

WELL LOCATED AND SAFE ACCOMMODATION

24/7 ASSISTANCE AND SUPPORT

LEARN MORE

READY TO DO AN INTERNSHIP IN ASIA?

CUSTOMIZED INTERNSHIP PLACEMENT

WELL LOCATED AND SAFE ACCOMMODATION

24/7 ASSISTANCE AND SUPPORT

LEARN MORE

Recommend Reading

- [How java manages Memory?](#)
- [Why is it advised to use hashCode and equals together?](#)
- [Producer Consumer Problem](#)
- [How to create Singleton class in Java?](#)
- [Lambdas in Java8](#)
- [Difference type of Map implementation](#)
- [Log Structured Merge Trees](#)
- [How Concurrent HashMap Works?](#)

Subscribe to get latest updates.

Email Address *

Subscribe