Tuesday, May 3, 2022 9:35 AM

Multithreading ------

java.lang.Thread

What is the path of execution of default thread ?

```
    p.s.v.main(String [] ) { ---path starts



    }//end of main method ---- path ends
```

What is the path of execution of other threads ?

```
     public void run() { ---path starts
    ....
    ...
    ..
    } //end of run method ----path ends
```

Method --- sequence of instructions = path
Method(){ statements ---- instructions
I1
I2
I3
}

We have an interface java.lang. Runnable in the library .

```
interface Runnable
{
    void run();
}
```

WHO ever is a (subclass of) Runnable will have run method !!

---

extends limitation
One class can extends only one other class!!!

Another way of creating threads
We implement Runnable

Let us create two threads ---
Thread1
Printing tables of numbers

HW

Thread 2 printing factorials of numbers

Two ways of thread creation ---------

1. Extends Thread
2. Implements Runnable

Thread API ----

static method sleep	
Non static methods of Thread class setName , getName ---- we can give a name to the thread , and get name of that thread	
Static method currentThread() ---------- it is useful to get thread object where Runnable is used	
non static method ------- join --- current thread waits for the joined thread to terminate	
setDaemon non static method ---- it will make the thread as daemon thread ---	
Daemon thread -- a thread that terminates automatically when all other non-daemon threads terminate---service thread	
setPriority , getPriority --- non static methods to set the priority of the thread	

HW

OVERRIDING Rules

In super class==>	public void f1()	public void f2() throws IOException	public void f3() throws Exception	void f4()	void f5()
In subclass	void f1() (default scope)	public void f2() throws Exception	public void f3() throws IOException	public void f4()	void f5() throws Exception
	CHECK if scope can be reduced in overridden method	CHECK if overridden method can throw more exceptions that superclass method	Check if overridden method can throw lesser exceptions than super class method	Check if scope can be expanded in overridden method	Check if overridden method can throw exceptions when superclass does not throw any

---

To share data using threads ---------------

HW -------

1. Write a class Account ---property balance , deposit, withdraw , showbalance , parameterized constructor
2. Write a Thread deposit
3. Write a Thread Withdraw
4. Write mainthread

We are seeing race condition

Solution to race condition --- mutual exclusion of critical section

Java uses **Monitors** for synchronization of threads

Monitors lock the critical section based on a lock

To mark the critical section we use a keyword called as **synchronized**

synchronized static method
synchronized non static method
synchronized block // MORE POPULAR

Java Monitors use objects as locks
static method uses ----- object of class Class as a lock
non static method uses ---- "this" as a lock
synchronized block -------- it uses any object passed to it as a lock

If two critical sections are using the same shared data then they should use same LOCK !!!

Monitors allow MUTEX ---- can be done using **synchronized** keyword
Monitors also allow inter thread communication through **conditional wait** using-----
wait
notify
notifyAll

Balance = 5000
Withdraw thread wants 5100 --- can it withdraw ??
withdraw thread waits on the account object ---- in the wait state

After some time deposit runs and adds 1000 to the account
And it notifies all threads waiting on the account object

All the waiting threads return to ready queue

HW --- study the wait notify program discussed in class

Extra HW
Think or implement Producer Consumer Problem !!!

Producer
Bounded buffer --- Shared object **Hint**: class Tray { int[] = new int[5] ....}
Consumer

Producer creates item and adds to the buffer
Consumer consumes the item and removes from buffer

If tray is full producer waits
If tray is empty consumer waits

Both producer and consumer will notify the other threads
