জাভাতে থ্রেডিং প্রোগ্রাম তৈরি

মো: আবদুল কাদের

থ্রেডিং হলো একটি কাজ, আর মাল্টিথ্রেডিং হলো অনেকগুলো কাজ। সাধারণত যে অপারেটিং সিস্টেম একসাথে অনেকগুলো কাজ করতে পারে তাকে মাল্টিথ্রেডেড অপারেটিং সিস্টেম বলে। অপারেটিং সিস্টেম উদ্ভাবনের প্রাথমিক পর্যায়ে একটি মাত্র কাজ করতে পারত। সেসব অপারেটিং সিস্টেমকে সিঙ্গেল থ্রেডেড অপারেটিং সিস্টেম বলা হয়। তবে, বর্তমানে প্রচলিত সব অপারেটিং সিস্টেমই এক সাথে অনেক কাজ করতে পারে যেমন একসাথে গান শোনার সাথে সাথে প্রয়োজনীয় অন্যান্য কাজও সম্পাদন করা যায়। এজন্য এগুলোকে মাল্টিথ্রেডেড অপারেটিং সিস্টেম বলে। মাল্টিথ্রেড কমপিউটার অ্যাপ্লিকেশনগুলোকে বহুমাত্রিকতা দিয়েছে এবং মানুষের কাঙ্ক্ষিত জীবনকে আরও প্রযুক্তি নির্ভর করে দিয়েছে।

সব কাজগুলোই প্রসেসরের মাধ্যমে রান করে। তাই একই সাথে অনেকগুলো কাজ করার সময় কোনো কাজকে সাময়িক বন্ধ রেখে, আবার কোনো কাজকে পুরোপুরি বন্ধ করে বা প্রসেসিংকে কাজগুলোর মধ্যে শেয়ার করে পরিচালিত করে। জাভা থ্রেডিং প্রোগ্রামকে sleep, stop() মেথড ব্যবহার করে বন্ধ করে আবার resume() মেথড ব্যবহার করে আবার থ্রেডকে চালায়। ফলে প্রসেসরের উপর স্বয়ংক্রিয়ভাবে চাপ কমে এবং জাভা প্রোগ্রাম সুন্দরভাবে রান করে।

জাভাতে দুটি পদ্ধতিতে থ্রেডিং অ্যাপ্লিকেশন করা হয়।

০১. Thread ক্লাসকে এক্সটেন্ড করে।
০২. Runnable interface ইমপ্লিমেন্ট করে।

Thread ক্লাস

এই ক্লাসের প্রয়োজনীয় কনস্ট্রাক্টর এবং মেথড রয়েছে যার মাধ্যমে Thread নিয়ে কাজ করা যায়।

Thread ক্লাসে বেশি ব্যবহার হওয়া কনস্ট্রাক্টরসমূহ

ক. Thread()
খ. Thread(String name)
গ. Thread(Runnable r)
ঘ. Thread(Runnable r,String name)

Thread ক্লাসের বেশি ব্যবহার হওয়া মেথডসমূহ

run() : থ্রেডের কোনো কাজ করতে ব্যবহার হয়।

start() : থ্রেড এক্সিকিউশন করতে ব্যবহার হয়। এর মাধ্যমে জাভা ভার্চুয়াল মেশিন থ্রেডের run() মেথডকে কাজ শুরু করতে বলে।

sleep(long miliseconds) : এই মেথডে দেয়া সংখ্যাকে মিলিসেকেন্ড হিসেবে ধরে থ্রেডকে চলার সময় বিরত রাখে।

getPriority() : থ্রেডের প্রায়োরিটি রিটার্ন করে।

setPriority(int priority) : থ্রেডের প্রায়োরিটি সেট করার জন্য ব্যবহার হয়।

getName() : থ্রেডের নাম দেখায়।

setName(String name) : থ্রেডের নাম সেট করতে ব্যবহার হয়।

currentThread() : বর্তমানে চলমান থ্রেডের রেফারেন্স রিটার্ন করে।

getId() : থ্রেডের আইডি রিটার্ন করে।

getState() : থ্রেডের বর্তমান অবস্থা সম্পর্কে অবহিত করে।

isAlive() : থ্রেড বর্তমানে Alive আছে কি না তা নিশ্চিত করে।

suspend() : থ্রেড সাসপেন্ড করতে ব্যবহার হয়।

resume() : সাসপেন্ডেড থ্রেডকে পুনরায় চলার জন্য আহবান করে।

stop() : থ্রেড বন্ধ করতে ব্যবহার হয়। সাসপেন্ড এবং স্টপ এর মধ্যে পার্থক্য হলো সাসপেন্ডে কিছু সময়ের জন্য থ্রেড বন্ধ থাকে। আর স্টপ মেথডের মাধ্যমে থ্রেডের কাজকে সম্পূর্ণভাবে বন্ধ করা হয়।

isDaemon() : থ্রেডটি কি ইউজার থ্রেড কিনা তা জানায়।

setDaemon(boolean b) : থ্রেডকে ইউজার থ্রেড হিসেবে ডিফাইন করা হয়।

interrupt() : থ্রেডের কাজকে ইন্টারাপ্ট করতে ব্যবহার হয়।

MyThread.java প্রোগ্রাম

```
    class MyThread extends Thread
    {
    public static void main(String args[])
    {
    Thread t=Thread.currentThread();
    System.out.println(“The current thread is “ +
t);
    t.setName(“MyJavaThread”);
    System.out.println(“The thread is now
named:” + t);
    try
    {
    for (int i=0; i<5; i++)
    {
    Thread.sleep(1000);
    System.out.println(“This text is printing after
one second each time”);
    }
    }
    catch(InterruptedException e)
    {
    System.out.println(“Main thread interupted”);
    }
    }
    }
```

প্রোগ্রামটি রান করার পদ্ধতি অন্যান্য জাভা প্রোগ্রামের মতোই। আমরা রান করার জন্য জাভার Jdk1.4 ভার্সন ব্যবহার করব এবং প্রোগ্রামগুলো D:\ ড্রাইভের java ফোল্ডারে সেভ করব। উপরের প্রোগ্রামটি নোটপ্যাডে টাইপ করে **MyThread.java** নামে সেভ করতে হবে।

প্রোগ্রামটিতে currentThread() মেথড ব্যবহার করা হয়েছে। ফলে কোন থ্রেড রান করছে তা দেখাচ্ছে। বাই ডিফল্ট যেকোনো জাভা প্রোগ্রাম মেইন থ্রেড থেকে কাজ করে থাকে। তাই মেইন মেথড দেখিয়েছে। পরবর্তী সময় setName মেথড ব্যবহার করে থ্রেডের নাম পরিবর্তন করা হয়েছে। সবশেষে থ্রেডটি চলার সময় ১ সেকেন্ড পর পর একটি লেখা প্রিন্ট করার জন্য sleep() মেথড ব্যবহার করা হয়েছে যার ভ্যালু দেয়া হয়েছে ১০০০ মিলিসেকেন্ড।

Runnable interface ইমপ্লিমেন্ট

Runnable interface ইমপ্লিমেন্ট করেও থ্রেড প্রোগ্রাম তৈরি করা যায়। এর run() নামে একটি মাত্র মেথড রয়েছে। Runnable interface-কে ইমপ্লিমেন্ট করার প্রোগ্রাম নিচে দেয়া হলো। প্রোগ্রামটি MemorialStand.java নামে সেভ করতে হবে।

```
    import java.awt.*;
    import java.applet.Applet;
    /*<applet code=”MemorialStand.class”
width=750 height=500> </applet>*/
    public class MemorialStand extends Applet
implements Runnable
    {
    int
x1[]={20,60,100,140,180,220,260,300,340,340};
    int y2[]={372,340,310,270,170,120,80,30,5,
420};
    int
x2[]={720,680,640,600,560,520,480,440,400,40
0};
    int j=0, k=0, red=0, green=0, blue=0;   //ini-
tialization
    public void init()
    {
      new Thread (this).start();
    }
    public void update (Graphics g)
    {
      g.fillRect(20,450,700,40);
      //Draw Memorial Stand

      red=(int)(Math.random()*255.0);
      green=(int)(Math.random()*255.0);
      blue=(int)(Math.random()*255.0);
      g.setColor (new Color (red,green, blue));
      for(k=0;k<=9;k++)
      {
    g.drawLine(x1[k],450,380,y2[k]);
      g.drawLine(x2[k],450,380,y2[k]);
      }
      // draw flag
      g.drawLine (380,420,380,5);
      g.setColor(Color.green);
      g.fillRect(380,170,100,70);
      g.setColor(Color.red);
      g.fillOval(410,190,50,35);
    }
    public void run()
    {
      for (j=0;  ;j++)
      {
      try
      {
      Thread.sleep (1000);
      }
      catch(Exception e){}
      if (j==14)j=0;

      repaint();
      }
    }
    }
```

কজ

ফিডব্যাক : ***balaith@gmail.com***