**TASK01**:

import java.util.\*;

class Task01Thread extends Thread{

//Scanner var = new Scanner(System.in);

int n1,n2;

String thread;

public Task01Thread(String name, int a, int b){

super(name);

thread = name;

n1= a;

n2 = b;

}

@Override

public void run(){

int addition, subtraction, multiplication, division;

if(thread == "add"){

addition = n1+n2;

System.out.println("Addition value: "+addition);

System.out.println("Threrad name: "+Thread.currentThread().getName());

}

else if(thread == "sub"){

subtraction = n1-n2;

System.out.println("Subtraction value: "+subtraction);

System.out.println("Threrad name: "+Thread.currentThread().getName());

}

else if(thread == "mul"){

multiplication = n1\*n2;

System.out.println("Multiplication value: "+multiplication);

System.out.println("Threrad name: "+Thread.currentThread().getName());

}

else if(thread == "div"){

division = n1/n2;

System.out.println("Division value: "+division);

System.out.println("Threrad name: "+Thread.currentThread().getName());

}

else {

System.out.println("oth: No valid operation");

System.out.println("Threrad name: " + Thread.currentThread().getName());

}

}

}

public class Task01\_Final {

public static void main(String [] args){

Scanner var = new Scanner(System.in);

System.out.println("Please enter a number");

int n1 = var.nextInt();

System.out.println("Please enter a number");

int n2 = var.nextInt();

//System.out.println("Please enter a user input");

//String input = var.nextLine();

Task01Thread addition = new Task01Thread("add", n1, n2);

Task01Thread subtraction = new Task01Thread("sub", n1,n2);

Task01Thread multiplication = new Task01Thread("mul", n1,n2);

Task01Thread division = new Task01Thread("div",n1,n2);

Task01Thread other = new Task01Thread("oth", n1, n2);

addition.start();

subtraction.start();

multiplication.start();

division.start();

other.start();

}

}

**TASK02**:

class House\_Stark extends Thread{

public House\_Stark(String name){

super(name);

}

@Override

public void run(){

System.out.println("The house is : "+ getName());

try {

sleep(1000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

class House\_Targaryen extends Thread{

public House\_Targaryen(String name){

super(name);

}

@Override

public void run(){

System.out.println("The house is : "+ getName());

try {

sleep(1000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

class House\_Lannister extends Thread{

public House\_Lannister(String name){

super(name);

}

@Override

public void run(){

System.out.println("The house is : "+ getName());

try {

sleep(3000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

class House\_Bolton extends Thread{

public House\_Bolton(String name){

super(name);

}

@Override

public void run(){

System.out.println("The house is : "+ getName());

try {

sleep(3000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

class House\_Tyrell extends Thread{

public House\_Tyrell(String name){

super(name);

}

@Override

public void run(){

System.out.println("The house is : "+ getName());

try {

sleep(5000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

public class Task02\_Final {

public static void main(String[]args){

House\_Stark stark = new House\_Stark("House Stark");

House\_Targaryen targaryen = new House\_Targaryen("House Targaryen");

House\_Lannister lannister = new House\_Lannister("House Lannister");

House\_Bolton bolton = new House\_Bolton("House Bolton");

House\_Tyrell tyrell = new House\_Tyrell("House Tyrell");

//System.out.println("Run");

stark.run();

targaryen.run();

lannister.run();

bolton.run();

//System.out.println("Start");

stark.start();

tyrell.start();

lannister.start();

bolton.start();

try{

stark.join();

//targaryen.join();

lannister.join();

bolton.join();

//tyrell.join();

}catch(Exception e){

System.out.println("Exception" + e );

}

if(stark.isAlive())

System.out.println("Not Today!");

if(!bolton.isAlive())

System.out.println("You know nothing!");

}

}

**TASK03**:

class FirstHalfEven extends Thread{

long [] array = new long[9];

long sum = 0;

public FirstHalfEven(long [] arr){

//super(arr);

array = arr;

}

@Override

public void run() {

for(int i=0;i<9;i++){

sum=sum+ array[i];

}

System.out.println(sum);

sum = sum/9;

System.out.println(sum);

}

public long getValue(){

return sum;

}

}

class FirstHalfOdd extends Thread{

long [] array = new long[9];

long sum = 0;

public FirstHalfOdd(long [] arr){

//super(arr);

array = arr;

}

@Override

public void run() {

for(int i=0;i<16;i++){

sum=sum+ array[i];

}

System.out.println(sum);

sum = sum/16;

System.out.println(sum);

}

public long getValue(){

return sum;

}

}

class SecondHalfEven extends Thread{

long [] array = new long[9];

long sum = 0;

public SecondHalfEven(long [] arr){

//super(arr);

array = arr;

}

@Override

public void run() {

for(int i=0;i<8;i++){

sum=sum+ array[i];

}

System.out.println(sum);

sum = sum/8;

System.out.println(sum);

}

public long getValue(){

return sum;

}

}

class SecondHalfOdd extends Thread{

long [] array = new long[9];

long sum = 0;

public SecondHalfOdd (long [] arr){

//super(arr);

array = arr;

}

@Override

public void run() {

for(int i=0;i<17;i++){

sum=sum+ array[i];

}

System.out.println(sum);

sum = sum/17;

System.out.println(sum);

}

public long getValue(){

return sum;

}

}

class Average extends Thread{

long sum,value1, value2, value3, value4;

public Average (long val1, long val2, long val3, long val4){

value1=val1;

value2= val2;

value3 = val3;

value4 = val4;

}

@Override

public void run() {

sum = value1+value2+value3+value4;

sum = sum/4;

}

public long getValue(){

return sum;

}

}

public class Task03 {

public static void main(String[]args){

long First\_position=0,second\_position=1,temp;

long array [] = new long[50];

array[0] = 0;

array[1]= 1;

for(int i=2;i<50;i++) {

temp = First\_position + second\_position;

array[i] = temp;

//System.out.println(array[i]+" ");

First\_position=second\_position;

second\_position = temp;

}

// for(int i =0; i< array.length;i++){

// System.out.println(array[i]);

// }

long first\_half\_odd [] = new long[16];

long first\_half\_even [] = new long[9];

long second\_half\_odd [] = new long[17];

long second\_half\_even [] = new long[8];

int count1=0, count2=0, count3=0,count4=0;

for(int i = 0; i<25; i++){

if(array[i]%2==0) {

first\_half\_even[count1] = array[i];

//System.out.println(first\_half\_even[count1]);

count1++;

}

else {

first\_half\_odd[count2] = array[i];

//System.out.println(first\_half\_odd[count2]);

count2++;

}

}

for(int i = 25; i<50; i++){

if(array[i]%2==0) {

second\_half\_even[count3] = array[i];

//System.out.println(second\_half\_even[count3]);

count3++;

}

else {

second\_half\_odd[count4] = array[i];

//System.out.println();

// System.out.println(second\_half\_odd[count4]);

count4++;

}

}

FirstHalfEven firstEven = new FirstHalfEven(first\_half\_even);

firstEven.start();

FirstHalfOdd firstOdd = new FirstHalfOdd(first\_half\_odd);

firstOdd.start();

SecondHalfOdd secondOdd = new SecondHalfOdd(second\_half\_odd);

secondOdd.start();

SecondHalfEven secondEven = new SecondHalfEven(second\_half\_even);

secondEven.start();

try{

firstEven.join();

firstOdd.join();

secondOdd.join();

secondEven.join();

//average.join();

}catch(Exception e){

System.out.println("Exception" + e );

}

long value1 = firstEven.getValue();

long value2 = firstOdd.getValue();

long value3 = secondOdd.getValue();

long value4 = secondEven.getValue();

System.out.println(value1);

System.out.println(value2);

System.out.println(value3);

System.out.println(value4);

Average average = new Average(value1, value2, value3, value4);

average.start();

try{

average.join();

}catch(Exception e){

System.out.println("Exception" + e );

}

long avg = average.getValue();

System.out.println("Mean value: "+avg);

}

}