**Task 1**

import java.util.Scanner;

public class Task1 extends Thread {

Scanner sc = new Scanner(System.in);

Task1(String name) {

super(name);

}

@Override

public void run() {

if (getName().equals("add")) {

addition();

}

if (getName().equals("sub")) {

subtraction();

}

if (getName().equals("mul")) {

multiplication();

}

if (getName().equals("div")) {

division();

}

if (getName().equals("oth")) {

others();

}

}

public void addition() {

System.out.println("Please enter 2 numbers");

int x = sc.nextInt(); int y = sc.nextInt();

System.out.println("Addition Result: " + (x + y));

}

public void subtraction() {

System.out.println("Please enter 2 numbers");

int x = sc.nextInt(); int y = sc.nextInt();

System.out.println("Subtraction Result: " + (x - y));

}

public void multiplication() {

System.out.println("Please enter 2 numbers");

int x = sc.nextInt(); int y = sc.nextInt();

System.out.println("Multiplication Result: " + (x \* y));

}

public void division() {

System.out.println("Please enter 2 numbers");

int x = sc.nextInt(); int y = sc.nextInt();

System.out.println("Division Result: " + ((double)x / y));

}

public void others() {

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

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

Task1 thread1 = new Task1("add");

Task1 thread2 = new Task1("sub");

Task1 thread3 = new Task1("mul");

Task1 thread4 = new Task1("div");

Task1 thread5 = new Task1("oth");

System.out.println("Press 1 for add thread\nPress 2 for sub thread\nPress 3 for mul thread\nPress 4 for div thread\nPress 5 for other threads");

int cmd = sc.nextInt();

if (cmd == 1) thread1.start();

if (cmd == 2) thread2.start();

if (cmd == 3) thread3.start();

if (cmd == 4) thread4.start();

if (cmd == 5) thread5.start();

}

}

**Task 2**

public class Task2 extends Thread{

Task2(String name) {

super(name);

}

@Override

public void run() {

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

try {

if (getName().equals("House Stark") || getName().equals("House Targaryen")) {

Thread.sleep(1000);

}

else if (getName().equals("House Lannister") || getName().equals("House Bolton")) {

Thread.sleep(3000);

}

else {

Thread.sleep(5000);

}

} catch (Exception e) {}

}

public static void main(String[] args) {

Task2 thread1 = new Task2("House Stark");

Task2 thread2 = new Task2("House Targaryen");

Task2 thread3 = new Task2("House Lannister");

Task2 thread4 = new Task2("House Bolton");

Task2 thread5 = new Task2("House Tyrell");

thread1.setPriority(Thread.MAX\_PRIORITY);

thread4.setPriority(Thread.MIN\_PRIORITY);

thread1.run(); // running

thread2.run(); // in

thread3.run(); // single threaded

thread4.run(); // mode

thread1.start(); // running

thread5.start(); // in

thread3.start(); // multi threaded

thread4.start(); // mode

try {

thread1.join(); // waiting for

thread3.join(); // these threads

thread4.join(); // to complete their tasks

} catch (InterruptedException e) {

e.printStackTrace();

}

if (thread1.isAlive()) System.out.println("Not Today!");

if (!thread4.isAlive()) System.out.println("You know nothing!");

}

}

**Task 3**

public class Task3 extends Thread {

static long[] fib\_series = new long [50];

static double mean1;

static double mean2;

static double mean3;

static double mean4;

static double final\_mean;

Task3(String name) {

super(name);

}

@Override

public void run() {

if (getName().equals("task 1")) task1();

if (getName().equals("task 2")) task2();

if (getName().equals("task 3")) task3();

if (getName().equals("task 4")) task4();

if (getName().equals("task 5")) task5();

}

void task1() {

long sum = 0;

double c = 0.0;

for (int i = 0; i < fib\_series.length / 2; ++i) {

if (fib\_series[i] % 2 != 0) {

sum += fib\_series[i];

++c;

}

}

mean1 = sum / c;

}

void task2() {

long sum = 0;

double c = 0.0;

for (int i = 0; i < fib\_series.length / 2; ++i) {

if (fib\_series[i] % 2 == 0) {

sum += fib\_series[i];

++c;

}

}

mean2 = sum / c;

}

void task3() {

long sum = 0;

double c = 0.0;

for (int i = fib\_series.length / 2; i < fib\_series.length; ++i) {

if (fib\_series[i] % 2 != 0) {

sum += fib\_series[i];

++c;

}

}

mean3 = sum / c;

}

void task4() {

long sum = 0;

double c = 0.0;

for (int i = fib\_series.length / 2; i < fib\_series.length; ++i) {

if (fib\_series[i] % 2 == 0) {

sum += fib\_series[i];

++c;

}

}

mean4 = sum / c;

}

void task5() {

final\_mean = (mean1 + mean2 + mean3 + mean4) / 4.0;

System.out.println("The PIN i.e Special integer: " + (int)final\_mean);

}

public static void main(String[] args) {

fib\_series[0] = 0; fib\_series[1] = 1;

for (int i = 2; i < fib\_series.length; ++i) { // generating the

fib\_series[i] = fib\_series[i-1] + fib\_series[i-2]; // fibonacci series

}

Task3 thread1 = new Task3("task 1");

Task3 thread2 = new Task3("task 2");

Task3 thread3 = new Task3("task 3");

Task3 thread4 = new Task3("task 4");

Task3 thread5 = new Task3("task 5");

thread1.start();

thread2.start();

thread3.start();

thread4.start();

try {

thread1.join();

thread2.join();

thread3.join();

thread4.join();

} catch (InterruptedException e) {

e.printStackTrace();

}

thread5.start();

}

}