W programie zaimplementowałem metody trapezów,prostokątów oraz Simpsona

W każdej z funkcji nadpisałem metodę run z Thread.

@Override  
public void run() {  
 super.run();  
}

W metodzie oblicz() zwracam obliczoną przybliżoną całkę

*sqrt*(1.3 \* x + 2.1) / *log*(x + 0.9);

W metodzie main tworzę przykładowy przedział całkowania.W pętli for każdy proces dostaje swój własny oparty na poprzednim przebiegu pętli.Kolejno dodaje wyniki działań procesów i wypisuję ID procesu.W tym przykładzie każda metoda jest wykonana przez 1000 procesów

public static void main(String[] args) {  
 double poczatek = 0.5;  
 double koniec = 1.0;  
 int ilosc\_watkow = 1000;  
 double przedzial = (koniec-poczatek)/ilosc\_watkow;  
 double simson=0,trapez=0,prostokaty=0;  
 double zakres = poczatek + przedzial;  
  
  
 while (zakres<=koniec)  
 {  
 M\_trapezow f = new M\_trapezow(poczatek, zakres,ilosc\_watkow);  
 f.run();  
 M\_Simpsona s = new M\_Simpsona(poczatek, zakres, ilosc\_watkow);  
 s.run();  
  
 M\_prostokatow p = new M\_prostokatow(poczatek, zakres, ilosc\_watkow);  
 p.run();  
  
// try{  
// f.join();  
// s.join();  
// p.join();  
// }catch(Exception e){System.out.println(e);}  
  
 trapez += f.oblicz();  
 System.*out*.println("Metoda trapezow:" + trapez + " "+f.getId()+" ID");  
  
 simson += s.oblicz();  
 System.*out*.println("Metoda simpsona:" + simson + " "+s.getId()+" ID");  
  
 prostokaty += p.oblicz();  
 System.*out*.println("Metoda prostokatow:" + prostokaty + " "+p.getId()+" ID"+"\n");  
  
  
  
  
 poczatek += przedzial;  
 zakres += przedzial;  
  
 }

