Uruchamianie i zatrzymywanie równoległego działania kodów

Zbudować klasę StringTask, symulująca długotrwałe obliczenia, tu polegające na konkatenacji napisow. Konstruktor klasy otrzymuje jako argument napis do powielenia oraz liczbę oznaczającą ile razy ten napis ma być powielony.
Klasa winna implementować interfejs Runnable, a w jej metodzie run() wykonywane jest powielenia napisu, przy czym to powielenia ma się odbywac za pomoca operatora '+' stosowanego wobec zmiennych typu String (to właśnie długotrwała operacja). **Użycie '+' jest warunkiem obowiązkowe**.

Obiekt klasy StringTask traktujemy jako zadanie, które może się wykonywać równolegle z innymi.
Możliwe stany zadania to:

- CREATED - zadanie utworzone, ale nie zaczęło się jeszcze wykonywać,
- RUNNING - zadanie się wykonuje w odrebnym wątku
- ABORTED - wykonanie zadania zostało przerwane
- READY - zadanie zakończyło się pomyślnie i sa gotowe wyniki.

W klasie StringTask zdefiniować metody:

- public String getResult() - zwracającą wynik konkatenacji
- public TaskState getState() - zwracającą stan zadania
- public void start() - uruchamiającą zadanie w odrębnym watku
- public void abort() - przerywającą wykonanie kodzu zadania i działanie watku
- public boolean isDone() - zwracająca true, jeśli wykonanie zadania się zakończyło normalnie lub przez przerwanie, false w przeciwnym razie

Poniższy kod program:

```
public class Main {

  public static void main(String[] args) throws InterruptedException {
    StringTask task = new StringTask("A", 70000);
    System.out.println("Task " + task.getState());
    task.start();
    if (args.length > 0 && args[0].equals("abort")) {
    /*<- tu zapisać kod  przerywający działanie tasku po sekundzie
         i uruchomic go w odrębnym wątku
    */
    }
    while (!task.isDone()) {
      Thread.sleep(500);
      switch(task.getState()) {
        case RUNNING: System.out.print("R."); break;
        case ABORTED: System.out.println(" ... aborted."); break;
        case READY: System.out.println(" ... ready."); break;
        default: System.out.println("uknown state");
      }

    }
    System.out.println("Task " + task.getState());
    System.out.println(task.getResult().length());
  }

}
```

uruchominy bez argumentu powinien wyprowadzić coś w rodzaju:
Task CREATED
R.R.R.R.R.R.R.R. ... ready.
Task READY

70000

a uruchomiony z argumentem "abort" może wyprowadzić:
Task CREATED
R. ... aborted.
Task ABORTED
31700

Uwaga 1. Plik Main.java może być modyfikowany tylko w miejscu oznaczonym /*<- */
Uwaga 2. Nie wolno uzywac metody System.exit(...)
Uwaga 3. W tym zadaniu nie należy stosować Executor/ExceutorService