

# "گزارش تمرین سری ششم"

## 1. CPU\_SIMULATOR

در قسمت TO DO نخست فقط لازم بود تا constructor کلاس Task تکمیل شود. در قسمت run هم همانطور که در کامنت خواسته شده بود ترد به مدت زمان اجرای تسک متوقف شده و برای این کار از متد sleep استفاده شده است. اما بخش قسمت اصلی برنامه از اینجا شروع میشود.

در کد نوشته شده یک Array list برای کل تسک ها و تسک های execute شده ایجاد شده است اما به یک Array list برای کل ترد های استفاده شده نیز نیاز هست. هم چنین در این قسمت نخستین برنامه متد sort task که در انتهای برنامه خودمان آن را با الگوریتم مرتب سازی حبابی براساس زمان اجرای (processing Time) تسک ها به صورت صعودی تعریف کرده ایم، فراخوانی کرده ایم تا در همان ابتدا تسک ها مرتب شوند. سپس با یک حلقه for روی Array list تسک ها به ازای هر تسک یک ترد ساخته میشود، تسک برای اجرا به آن داده میشود، سپس آیدی تسک به عنوان اسم ترد تعریف و ترد ایجاد شده به Array list کل تریدها که در بالاتر ساخته شده اضافه میشود. (برای تسک ها به ترتیبی که مرتب کردیم ترد ایجاد شده، و به تردها داده میشوند) در ادامه با یک حلقه for روی کل تریدهای ایجاد شده تمام ترید ها با متد start شروع به کار میکنند و در نهایت با یک حلقه for دیگر روی ترد ها برنامه منتظر می ماند تا تمام ترد ها یکی پس از دیگری به طور کامل terminate شوند و این امر موجب جلوگیری از توقف برنامه پیش از اجرای کامل تمام تردها میشود. بلوک آخر کد هم همانطور که پیش تر گفته شد متد sort Task برای مرتب سازی تسک ها براساس زمان اجراست.

(سعی شده مطالب اصلی و توضیحات لازم به عنوان کامنت هم در کد اضافه شود.)

## 2. FINDE MULTIPLIES

در ابتدای برنامه متغیر `sum` به صورت `public static` تعریف میشود تا بتوان از آن در `Subclass` به راحتی استفاده کرد. سپس کلاس `Multiplies` که `Runnable` تعریف شده است تا بتوان در آن از `Thread` استفاده کرد، پس از ایجاد `constructor` قرار است وظیفه اصلی برنامه یعنی بررسی اینکه اگر عددی به ۳، ۵ یا ۷ را انجام دهد و اگر چنین بود آن عدد را به `sum` اضافه کند. اما از آنجا که برای دسترسی به `sum` دچار `race condition` میشویم در پایین تر متد `add value` را ساخته که مقادیر موردنظر را به `sum` اضافه کرده و در آن از کلمه کلیدی `Synchronized` استفاده شده تا `race condition` موجب ایجاد نقطه بحرانی در برنامه نشود.

مشکلی که با آن مواجه شدم این بود که هر `unite test` به طور جداگانه درست ایجاد میشد اما وقتی از ابتدا تا انتها به صورت متوالی اجرا گرفته میشد با خطا مواجه میشدم. متوجه شدم به خاطر این است که هر بار مقدار `sum` حاصل شده به ازای قبلی در `sum` جدید گذاشته میشود و `sum` بجای شروع از صفر از مقدار قبلی خود شروع میکند و همین موجب افزایش مقدار خروجی نسبت به مقدار واقعی میشود. برای اینکار در `getsum` با تعریف دوباره `sum=0` متغیر `sum` هر بار رفرش میشود و این خطا برطرف میشود. برای اجرای عملیات ذکر شده از `newCachedThreadPool` استفاده شده تا برنامه هر تعداد `Thread` که نیاز بود بسازد و سرعت برنامه افزایش پیدا کند. در ادامه نیز مراحل ذکر شده به `Thread` واگذار و `Thread` شروع به کار میکند و در نهایت خاموش میشود. از متد `awaitTerminate` بدین منظور استفاده شده که اگر احيانا بعد از گذشت ده دقیقه همچنان کار تردها تمام نشده بود تردها `terminate` شوند و برنامه به پایان برسد.

### 3.USEINTERRUPTS

در این برنامه دو ترد `sleep thread` و `loop thread` تعریف شده اند که میخواهیم تنها سه ثانیه در حال اجرا باشند و اگر پس از آن خودشان متوقف نشده بودند ما آن ها را به واسطه یک ترد کمکی که آن را `stopper thread` نامیده ایم متوقف کنیم. بدین منظور کلاس `Stopper Thread` که از کلاس `Thread` ارث بری میکند را میسازیم که کاری که انجام میدهد توقف به مدت `sleep time` است که بعدا برای هر دو لوپ با توجه به صورت سوال ۳ ثانیه لحاظ میشود.

نحوه اجرا: در هر دو ترد `sleep thread` و `loop thread` پس از اجرای ترد اصلی یک ترد کمکی از این کلاس `Stopper thread` میسازیم و آن را نیز اجرا میکنیم و با استفاده از متد `join` مطمئن می شویم که ترید کمکی اجرا شده که این به معنای گذشت ۳ ثانیه زمان است حال اگر ترید اصلی همچنان در حال اجرا بود (`isAlive`) با دستور `interrupt` خودمان آن را متوقف می کنیم. در واقع این ترد کمکی ابزاری براس سنجش این است که آیا بعد از ۳ ثانیه ترد موردنظر همچنان در حال اجراست یا خیر، اگر بله پس متوقف شود!

مشکلی که در ابتدا با آن مواجه شدم این بود که اگرچه تردها `interrupt` میشدند اما فعالیت حلقه ها در عمل ادامه داشت و فقط ترد بود که متوقف میشد، برای تصحیح این امر در دستور اجرای هر دو حلقه این نکته را اضافه کردم که اگر ترد `interrupt` شده است حلقه متوقف شود.