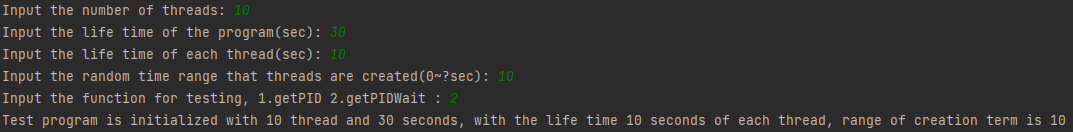
**Programming Assignment #2 - PID Manager**

17101992 Sumin Hong

This program shows the functions of pid manager by its test program.



The program takes 1. The number of threads, 2. The lifetime of the program (by sec), 3. The lifetime of the thread (by sec), 4. The random time range that thread are created, 0 to input value (by sec), 5. function that test in here, we can choose getPID or getPIDWait.

텍스트이(가) 표시된 사진

자동 생성된 설명

If program started, we could show Threads are created and destroyed on the time. At the program runtime is end, all threads are destroyed and program ends.

텍스트이(가) 표시된 사진

자동 생성된 설명

After end of the program, program prints the number of process ids.

This program consists of 3 classes and 1 interface.

Classes: Test (extends PIDManagerCls), PIDManagerCls (implements PIDManager), PIDThread (implements Runnable)

Interface: PIDManager (offered by Support Material)

PIDThread

PIDThread implements Runnable to use thread.

텍스트이(가) 표시된 사진

자동 생성된 설명

It has seq, id variables. Seq is the sequence of thread. It initialized at class creation. Id is process id.

PIDManagerCls object is created for using pid manager functions like getPID or getPIDWait.

텍스트이(가) 표시된 사진

자동 생성된 설명

We override run method; this is the work that thread do. Followed the user’s choice, thread get process id. It print the creation message and sleep while the thread lifetime. After that, pirnt the destroying message. If user choose getPIDwait, pidmanager release process id of this thread.

PIDManagerCls

텍스트이(가) 표시된 사진

자동 생성된 설명

PIDManagerCls has pidThreads arraylist to save pidthread.

Array pidList show whether the process id is used or not. Use index to process id.

텍스트이(가) 표시된 사진

자동 생성된 설명

getPID give process id to thread by turn around the pidList array and find 0(nothing is assigned).

If anything can’t be assigned, it return -1.

Declaration of synchronized do role of monitor in java.

텍스트이(가) 표시된 사진

자동 생성된 설명

getPIDWait give process id to thread by turn around the pidList array and find 0(nothing is assigned).

If anything can’t be assigned, pid become -1 and call wait(), So state of thread is set to blocked until notification come. It is wait any available process id. If pid is assigned, escape while statement and return process id.

releasePID is change the process id state is available and notify to waiting thread that some id can be used.

Test

텍스트이(가) 표시된 사진

자동 생성된 설명

threadNum, programLifeTime, threadLifeTime, randomCreateTime, programStartTime, pidorpidwait variables set static to use globally.

Program take each answer. If answer is invalid, go first command and take user input again.

텍스트이(가) 표시된 사진

자동 생성된 설명

Arraylist threads save each thread instance to stop when program runtime is over.

It saves start time at programStartTime variable.

텍스트이(가) 표시된 사진

자동 생성된 설명

It generates threads in for statement. There’re 4 break points. Second breakpoint breaks the program when value that sleep time that randomly set plus to present runtime is over program lifetime.

Other break points just stop the program when its runtime is over.

Sleeptime variable is the term between each thread creation. The process waits during sleeptime. After waiting, a thread is created by PIDThread instance, pidthread. The thread starts and is added to threads arraylist.

텍스트이(가) 표시된 사진

자동 생성된 설명

If for statement braked, it enters while statement. Only in the case of second breakpoint, it waits until the program runtime is over. Threads that alive at this point are interrupted, stopped by force.

After 0.5 sec for printing result, thread sequence number and process id that saved in pidthread instance is printed sequencly.