운영체제 #4,#5

– 4장 쓰레드

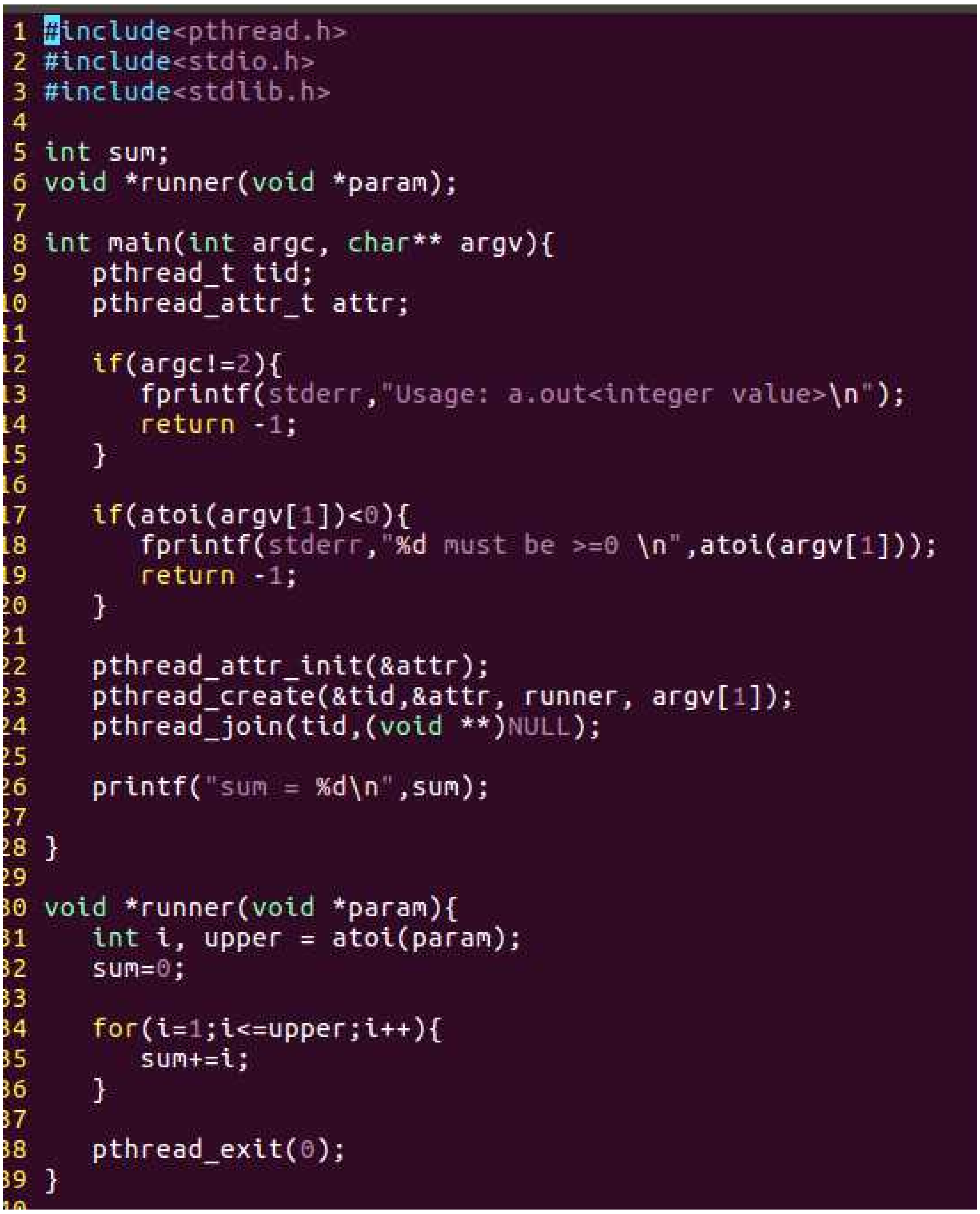
–5장 비동기 실행

제출일	2018-04-08
담당 교수	이건명 교수님
학번	2015041003
이름	구경민

#4장==

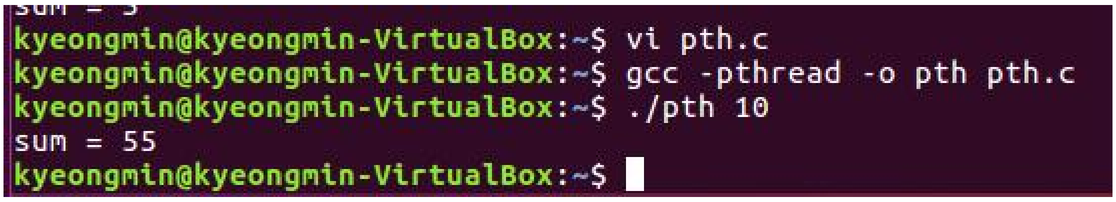
#POSIX Threads

<소스 코드>



```c
#include<pthread.h>
#include<stdio.h>
#include<stdlib.h>

int sum;
void *runner(void *param);

int main(int argc, char** argv){
   pthread_t tid;
   pthread_attr_t attr;

   if(argc!=2){
      fprintf(stderr,"Usage: a.out<integer value>\n");
      return -1;
   }

   if(atoi(argv[1])<0){
      fprintf(stderr,"%d must be >=0 \n",atoi(argv[1]));
      return -1;
   }

   pthread_attr_init(&attr);
   pthread_create(&tid,&attr, runner, argv[1]);
   pthread_join(tid,(void **)NULL);

   printf("sum = %d\n",sum);

}

void *runner(void *param){
   int i, upper = atoi(param);
   sum=0;

   for(i=1;i<=upper;i++){
      sum+=i;
   }

   pthread_exit(0);
}
```

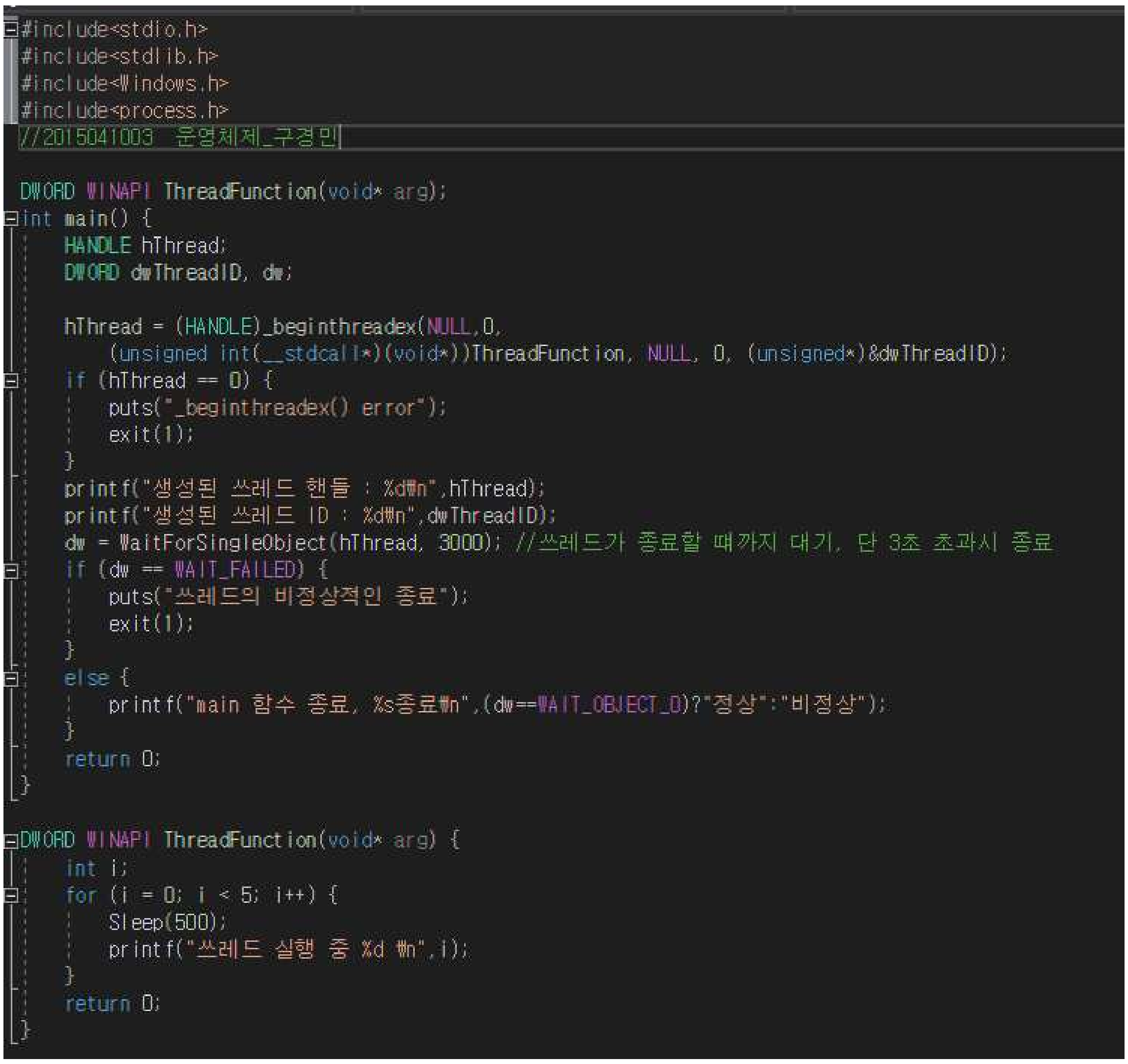
<실행 결과>



```
sum = 5
kyeongmin@kyeongmin-VirtualBox:~$ vi pth.c
kyeongmin@kyeongmin-VirtualBox:~$ gcc -pthread -o pth pth.c
kyeongmin@kyeongmin-VirtualBox:~$ ./pth 10
sum = 55
kyeongmin@kyeongmin-VirtualBox:~$ 
```

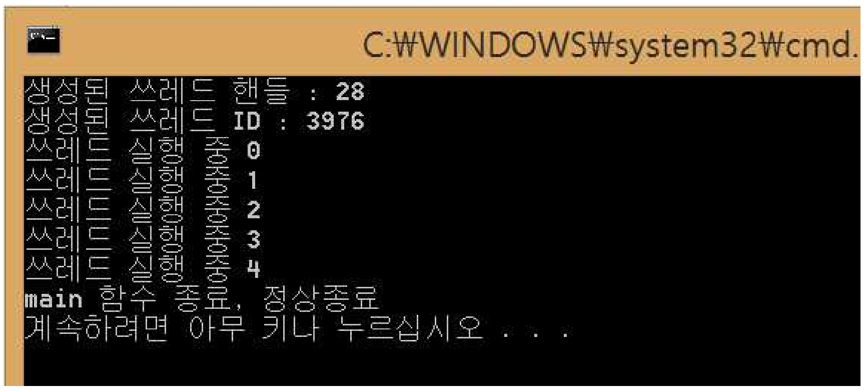
#<Windows 쓰레드>

<소스 코드>



```
#include<stdio.h>
#include<stdlib.h>
#include<Windows.h>
#include<process.h>
//2015041003 운영체제_구경민

DWORD WINAPI ThreadFunction(void* arg);
int main() {
    HANDLE hThread;
    DWORD dwThreadID, dw;

    hThread = (HANDLE)_beginthreadex(NULL,0,
        (unsigned int(__stdcall*)(void*))ThreadFunction, NULL, 0, (unsigned*)&dwThreadID);
    if (hThread == 0) {
        puts("_beginthreadex() error");
        exit(1);
    }
    printf("생성된 쓰레드 핸들 : %d₩n",hThread);
    printf("생성된 쓰레드 ID : %d₩n",dwThreadID);
    dw = WaitForSingleObject(hThread, 3000); //쓰레드가 종료할 때까지 대기, 단 3초 초과시 종료
    if (dw == WAIT_FAILED) {
        puts("쓰레드의 비정상적인 종료");
        exit(1);
    }
    else {
        printf("main 함수 종료, %s종료₩n",(dw==WAIT_OBJECT_0)?"정상":"비정상");
    }
    return 0;
}

DWORD WINAPI ThreadFunction(void* arg) {
    int i;
    for (i = 0; i < 5; i++) {
        Sleep(500);
        printf("쓰레드 실행 중 %d ₩n",i);
    }
    return 0;
}
```

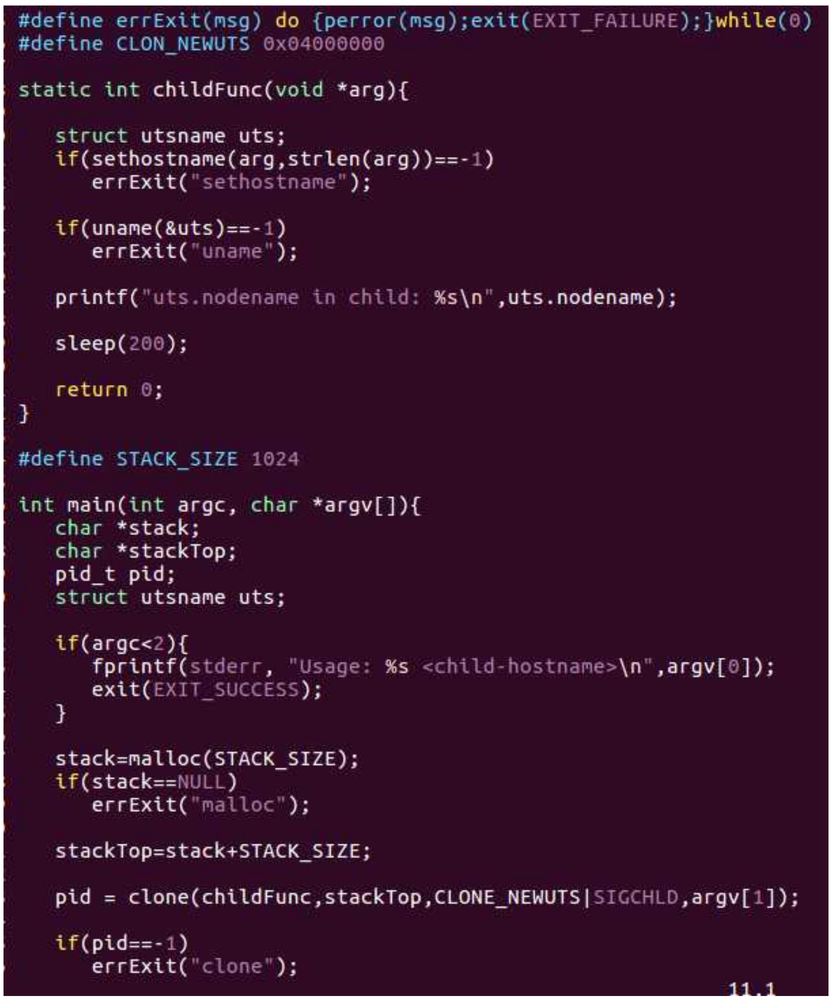
<실행 결과>



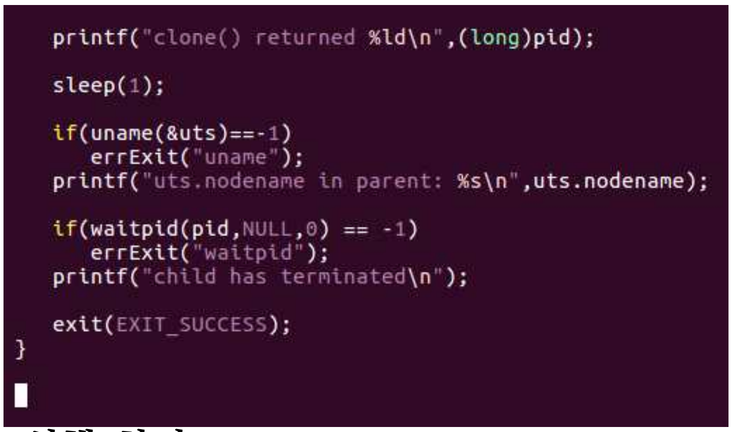
```
C:₩WINDOWS₩system32₩cmd.
생성된 쓰레드 핸들 : 28
생성된 쓰레드 ID : 3976
쓰레드 실행 중 0
쓰레드 실행 중 1
쓰레드 실행 중 2
쓰레드 실행 중 3
쓰레드 실행 중 4
main 함수 종료, 정상종료
계속하려면 아무 키나 누르십시오 . . .
```

#Linux 쓰레드

<소스 코드>



```c
#define errExit(msg) do {perror(msg);exit(EXIT_FAILURE);}while(0)
#define CLON_NEWUTS 0x04000000

static int childFunc(void *arg){

    struct utsname uts;
    if(sethostname(arg,strlen(arg))==-1)
       errExit("sethostname");

    if(uname(&uts)==-1)
       errExit("uname");

    printf("uts.nodename in child: %s\n",uts.nodename);

    sleep(200);

    return 0;
}

#define STACK_SIZE 1024

int main(int argc, char *argv[]){
    char *stack;
    char *stackTop;
    pid_t pid;
    struct utsname uts;

    if(argc<2){
       fprintf(stderr, "Usage: %s <child-hostname>\n",argv[0]);
       exit(EXIT_SUCCESS);
    }

    stack=malloc(STACK_SIZE);
    if(stack==NULL)
       errExit("malloc");

    stackTop=stack+STACK_SIZE;

    pid = clone(childFunc,stackTop,CLONE_NEWUTS|SIGCHLD,argv[1]);

    if(pid==-1)
       errExit("clone");
                                                       11,1
```

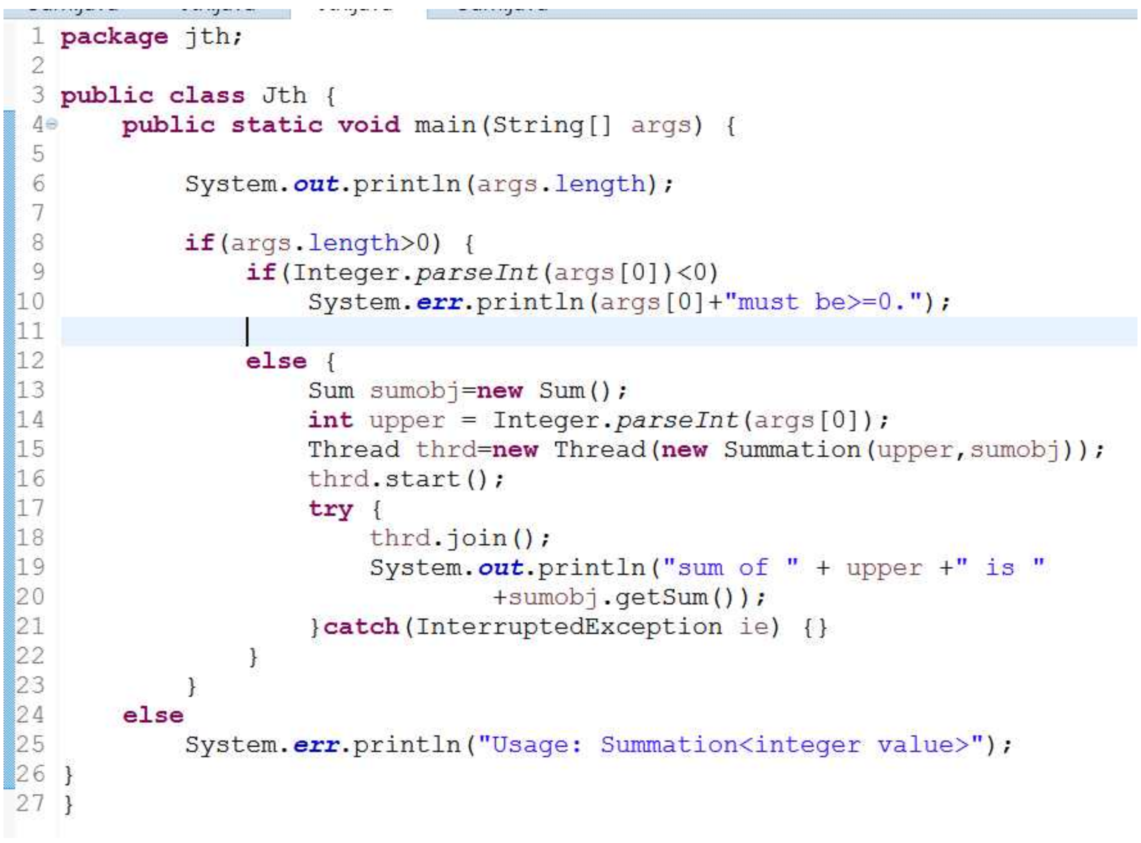


```
    printf("clone() returned %ld\n",(long)pid);

    sleep(1);

    if(uname(&uts)==-1)
       errExit("uname");
    printf("uts.nodename in parent: %s\n",uts.nodename);

    if(waitpid(pid,NULL,0) == -1)
       errExit("waitpid");
    printf("child has terminated\n");

    exit(EXIT_SUCCESS);
}
```

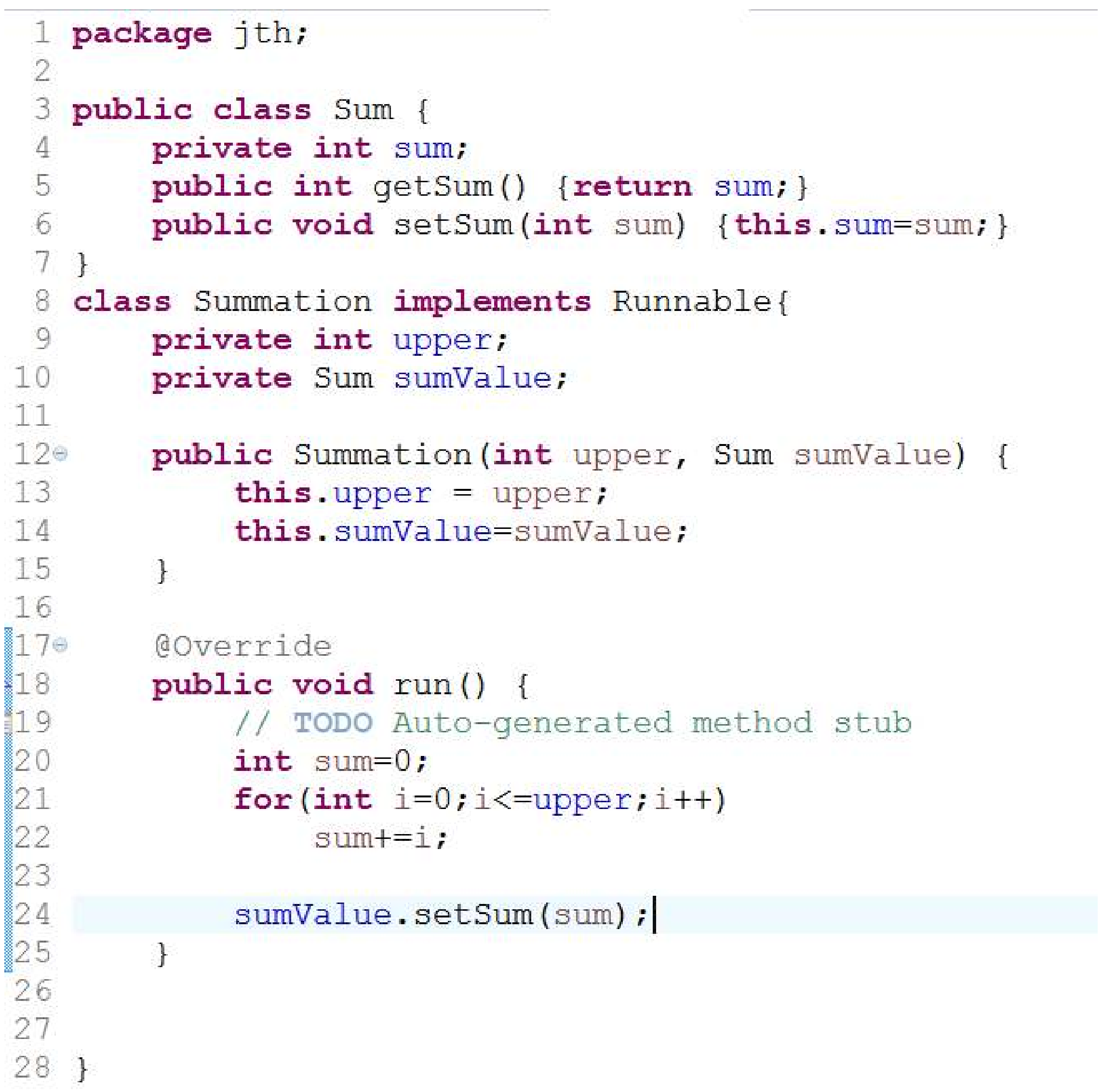
<실행 화면>

#java 쓰레드
처음의 args 값이 없어서 자꾸 Usage: summation오류가 떴다.
해결책
run configurations 에 가서 argument 값을 바꾸어주면 된다.

<소스 코드>

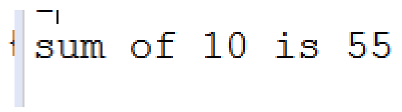


```java
package jth;

public class Jth {
    public static void main(String[] args) {

        System.out.println(args.length);

        if(args.length>0) {
            if(Integer.parseInt(args[0])<0)
                System.err.println(args[0]+"must be>=0.");

            else {
                Sum sumobj=new Sum();
                int upper = Integer.parseInt(args[0]);
                Thread thrd=new Thread(new Summation(upper,sumobj));
                thrd.start();
                try {
                    thrd.join();
                    System.out.println("sum of " + upper +" is "
                            +sumobj.getSum());
                }catch(InterruptedException ie) {}
            }
        }
    else
        System.err.println("Usage: Summation<integer value>");
}
}
```



```java
package jth;

public class Sum {
    private int sum;
    public int getSum() {return sum;}
    public void setSum(int sum) {this.sum=sum;}
}
class Summation implements Runnable{
    private int upper;
    private Sum sumValue;

    public Summation(int upper, Sum sumValue) {
        this.upper = upper;
        this.sumValue=sumValue;
    }

    @Override
    public void run() {
        // TODO Auto-generated method stub
        int sum=0;
        for(int i=0;i<=upper;i++)
            sum+=i;

        sumValue.setSum(sum);
    }


}
```

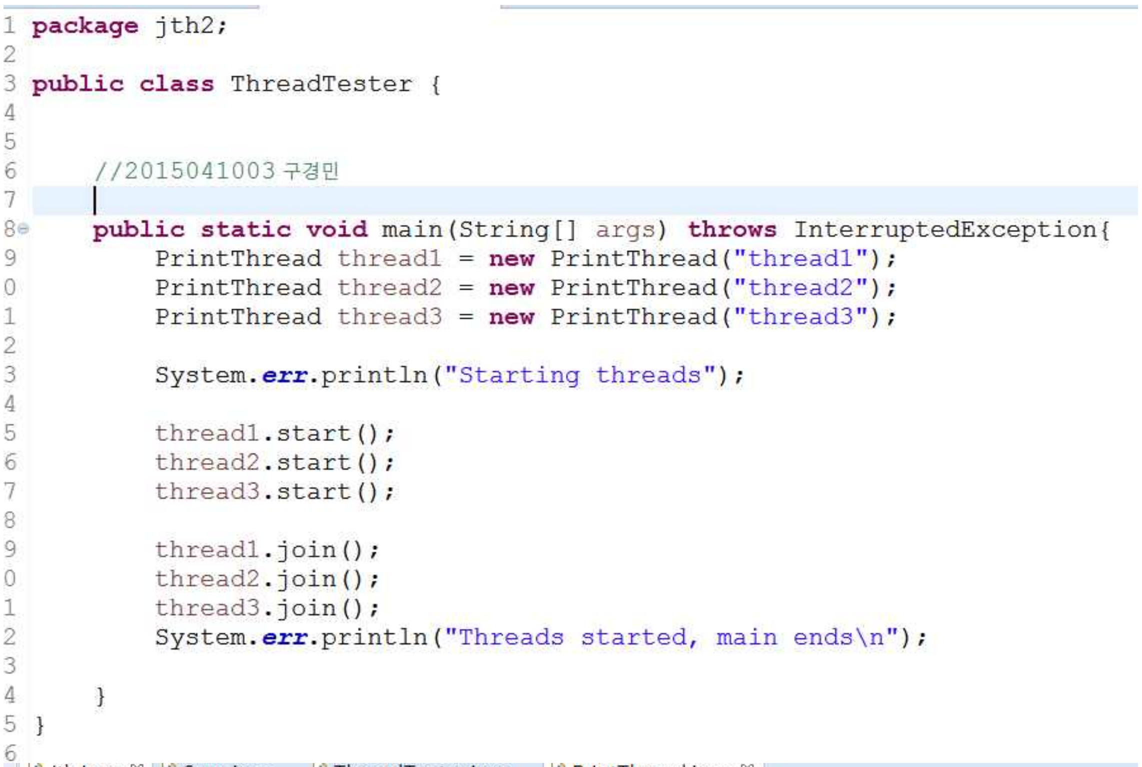
<실행 결과>



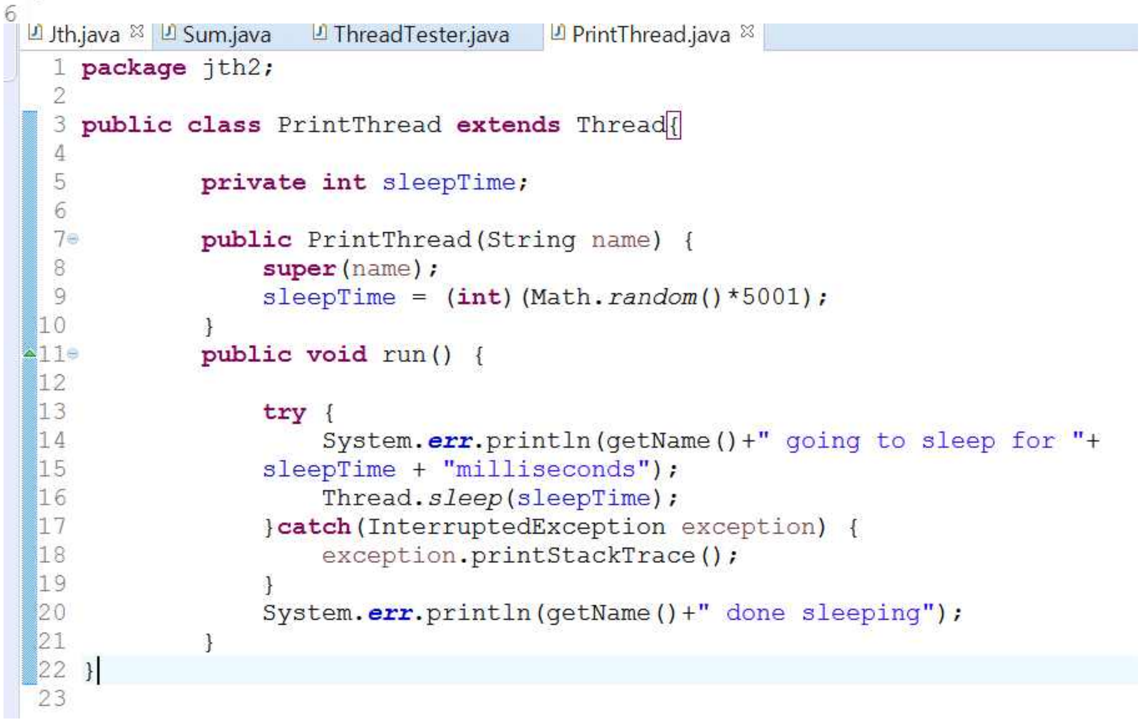
```
sum of 10 is 55
```

#Java 쓰레드 2

<소스 코드>

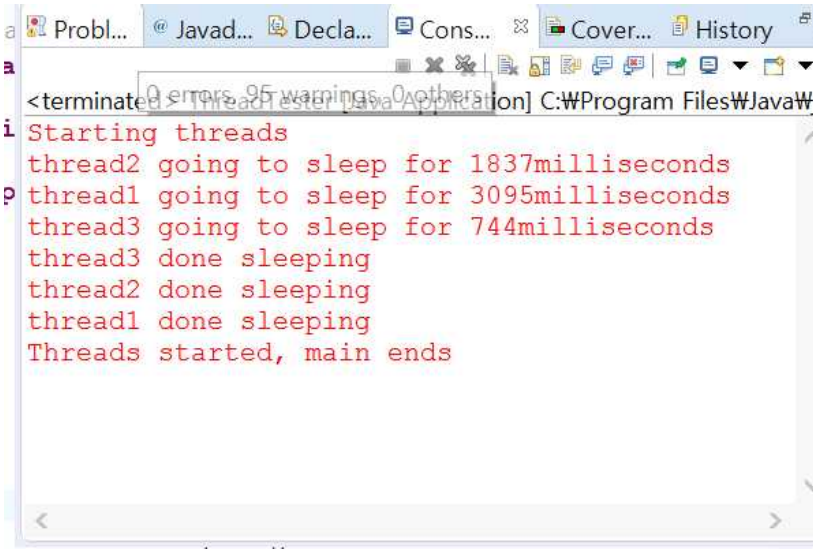


```java
package jth2;

public class ThreadTester {


    //2015041003 구경민

    public static void main(String[] args) throws InterruptedException{
        PrintThread thread1 = new PrintThread("thread1");
        PrintThread thread2 = new PrintThread("thread2");
        PrintThread thread3 = new PrintThread("thread3");

        System.err.println("Starting threads");

        thread1.start();
        thread2.start();
        thread3.start();

        thread1.join();
        thread2.join();
        thread3.join();
        System.err.println("Threads started, main ends\n");

    }
}
```



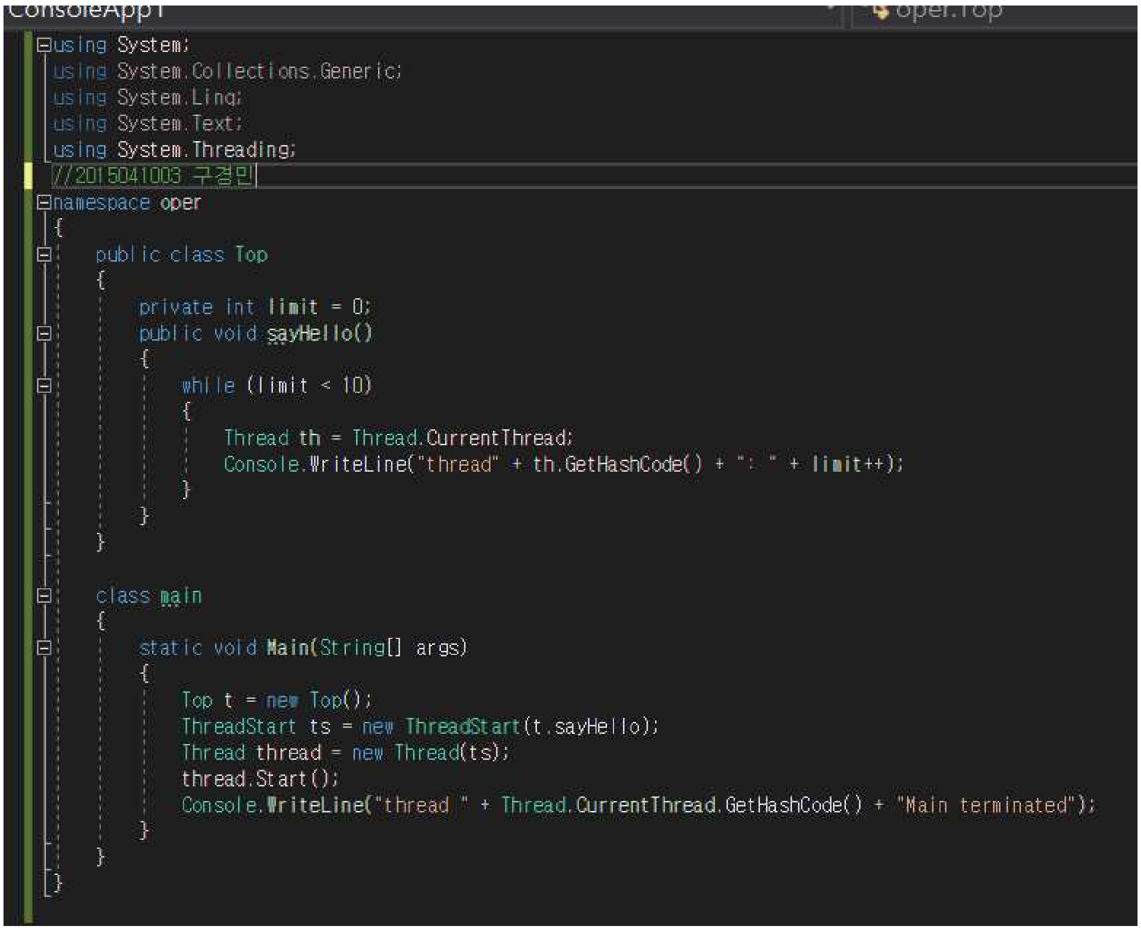
```java
package jth2;

public class PrintThread extends Thread{

        private int sleepTime;

        public PrintThread(String name) {
            super(name);
            sleepTime = (int)(Math.random()*5001);
        }
        public void run() {

            try {
                System.err.println(getName()+" going to sleep for "+
            sleepTime + "milliseconds");
                Thread.sleep(sleepTime);
            }catch(InterruptedException exception) {
                exception.printStackTrace();
            }
            System.err.println(getName()+" done sleeping");
        }
}
```

<실행 화면>



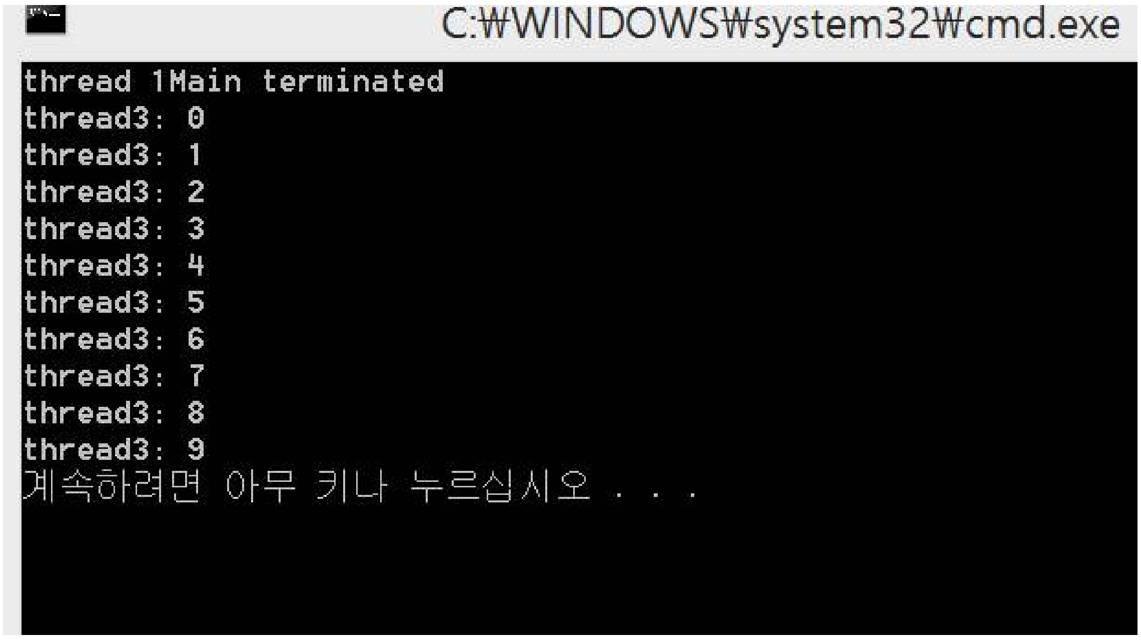
```
<terminated> ThreadTester [Java Application] C:₩Program Files₩Java₩
Starting threads
thread2 going to sleep for 1837milliseconds
thread1 going to sleep for 3095milliseconds
thread3 going to sleep for 744milliseconds
thread3 done sleeping
thread2 done sleeping
thread1 done sleeping
Threads started, main ends
```

#C# 쓰레드



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading;
//2015041003 구경민
namespace oper
{
    public class Top
    {
        private int limit = 0;
        public void sayHello()
        {
            while (limit < 10)
            {
                Thread th = Thread.CurrentThread;
                Console.WriteLine("thread" + th.GetHashCode() + ": " + limit++);
            }
        }
    }

    class main
    {
        static void Main(String[] args)
        {
            Top t = new Top();
            ThreadStart ts = new ThreadStart(t.sayHello);
            Thread thread = new Thread(ts);
            thread.Start();
            Console.WriteLine("thread " + Thread.CurrentThread.GetHashCode() + "Main terminated");
        }
    }
}
```

<실행 화면>

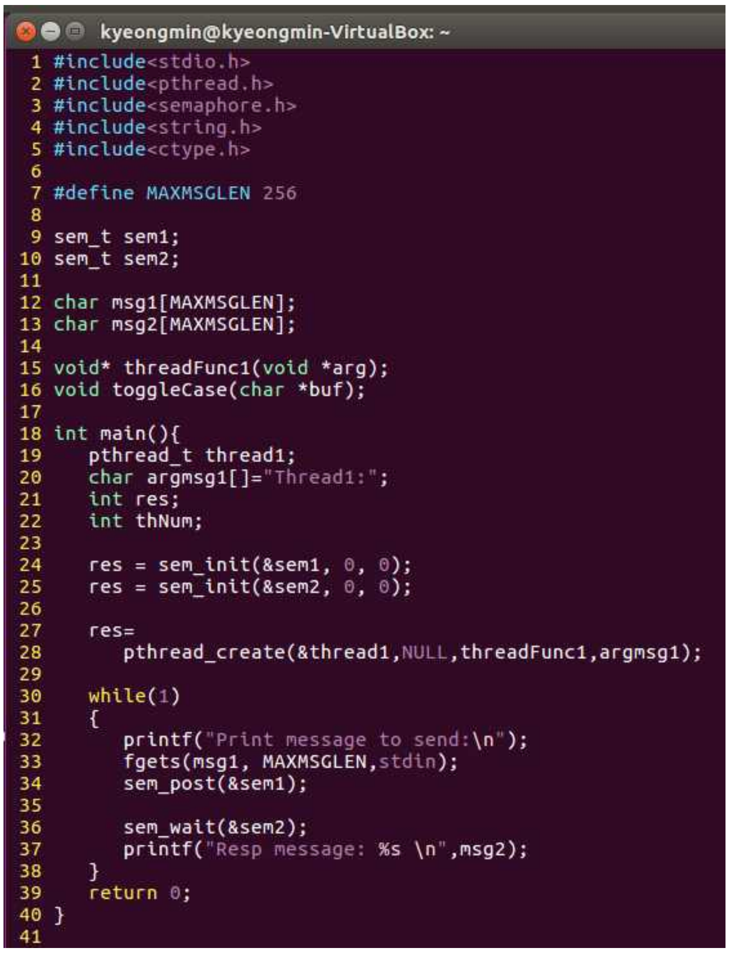


```
thread 1Main terminated
thread3: 0
thread3: 1
thread3: 2
thread3: 3
thread3: 4
thread3: 5
thread3: 6
thread3: 7
thread3: 8
thread3: 9
계속하려면 아무 키나 누르십시오 . . .
```

#5장===

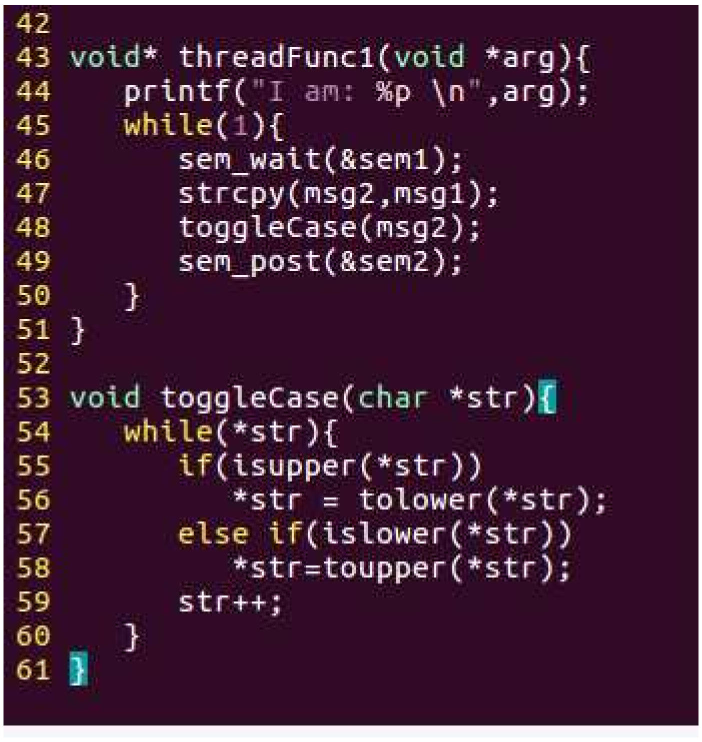
#POSIX 세마포어

<소스 코드>



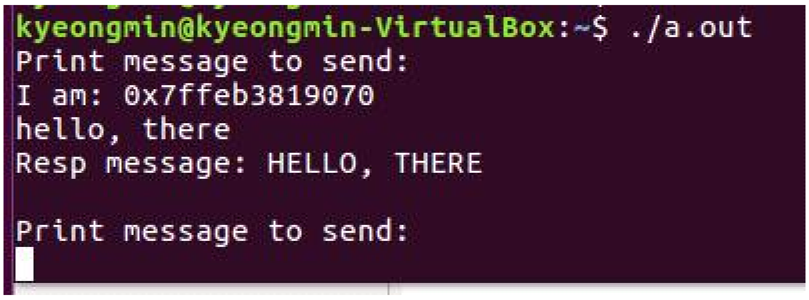
kyeongmin@kyeongmin-VirtualBox: ~

```c
#include<stdio.h>
#include<pthread.h>
#include<semaphore.h>
#include<string.h>
#include<ctype.h>

#define MAXMSGLEN 256

sem_t sem1;
sem_t sem2;

char msg1[MAXMSGLEN];
char msg2[MAXMSGLEN];

void* threadFunc1(void *arg);
void toggleCase(char *buf);

int main(){
    pthread_t thread1;
    char argmsg1[]="Thread1:";
    int res;
    int thNum;

    res = sem_init(&sem1, 0, 0);
    res = sem_init(&sem2, 0, 0);

    res=
       pthread_create(&thread1,NULL,threadFunc1,argmsg1);

    while(1)
    {
       printf("Print message to send:\n");
       fgets(msg1, MAXMSGLEN,stdin);
       sem_post(&sem1);

       sem_wait(&sem2);
       printf("Resp message: %s \n",msg2);
    }
    return 0;
}

```



```c

void* threadFunc1(void *arg){
    printf("I am: %p \n",arg);
    while(1){
        sem_wait(&sem1);
        strcpy(msg2,msg1);
        toggleCase(msg2);
        sem_post(&sem2);
    }
}

void toggleCase(char *str){
    while(*str){
        if(isupper(*str))
            *str = tolower(*str);
        else if(islower(*str))
            *str=toupper(*str);
        str++;
    }
}
```

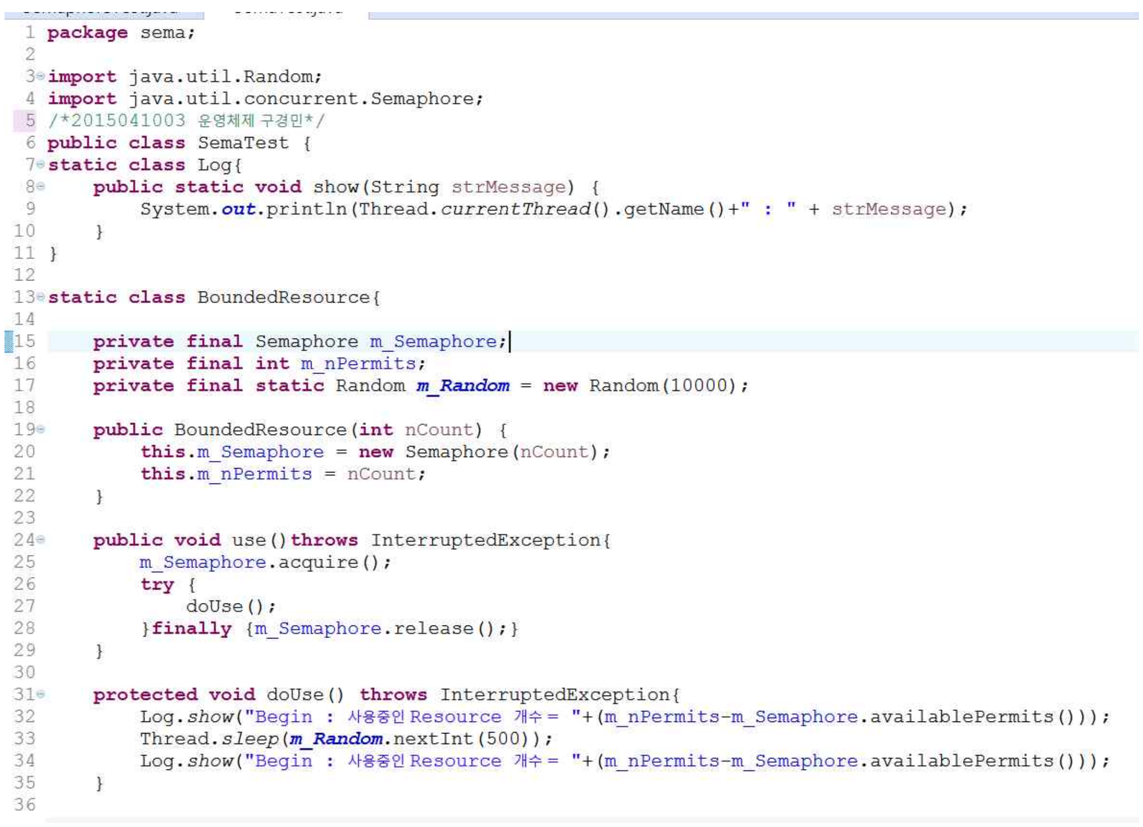
<실행 화면>



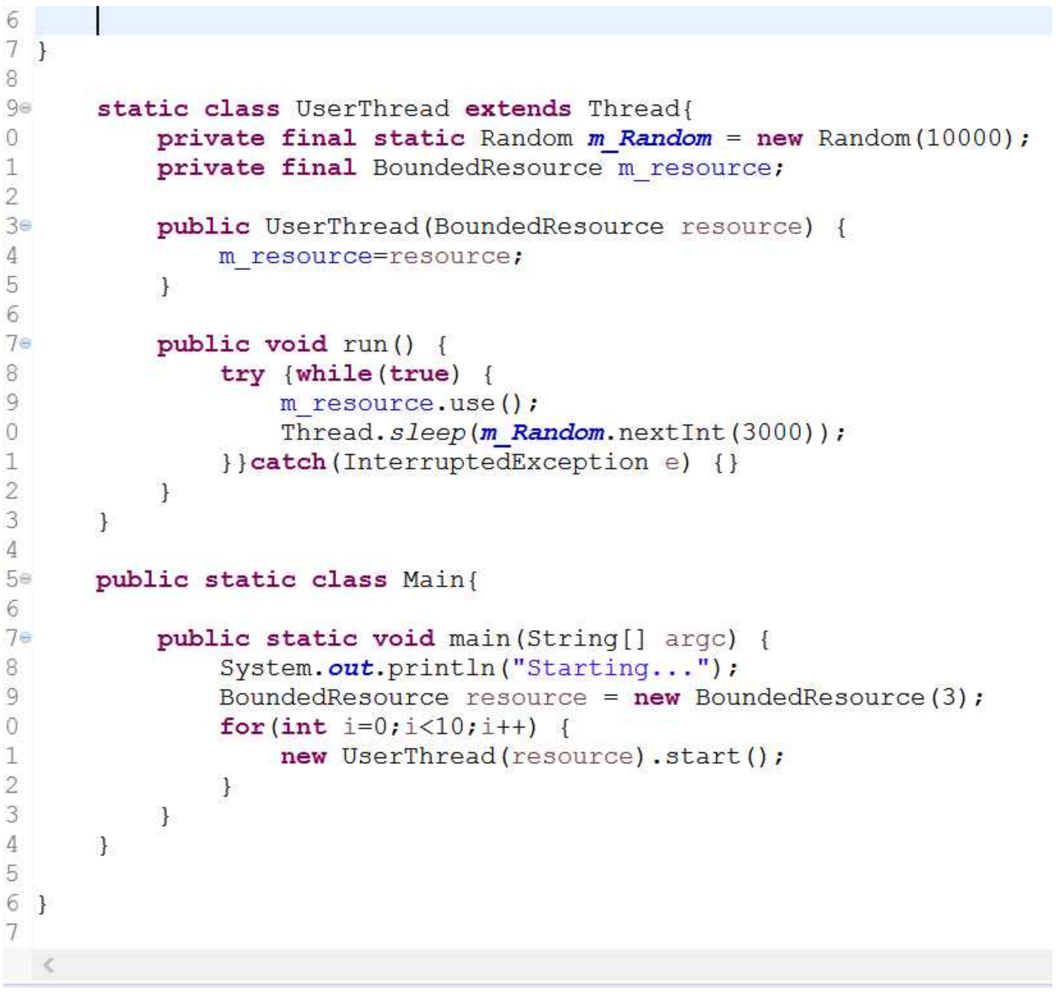
```
kyeongmin@kyeongmin-VirtualBox:~$ ./a.out
Print message to send:
I am: 0x7ffeb3819070
hello, there
Resp message: HELLO, THERE

Print message to send:
```

#Java 세마포어

<소스 코드>

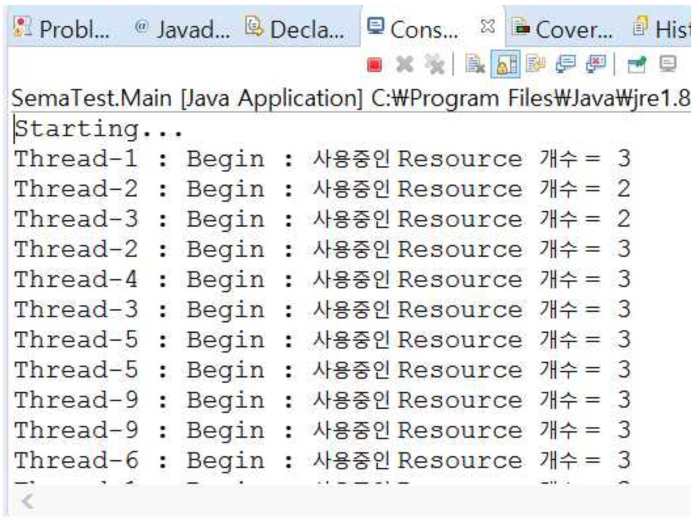


```
package sema;

import java.util.Random;
import java.util.concurrent.Semaphore;
/*2015041003 운영체제 구경민*/
public class SemaTest {
static class Log{
    public static void show(String strMessage) {
        System.out.println(Thread.currentThread().getName()+" : " + strMessage);
    }
}

static class BoundedResource{

    private final Semaphore m_Semaphore;
    private final int m_nPermits;
    private final static Random m_Random = new Random(10000);

    public BoundedResource(int nCount) {
        this.m_Semaphore = new Semaphore(nCount);
        this.m_nPermits = nCount;
    }

    public void use()throws InterruptedException{
        m_Semaphore.acquire();
        try {
            doUse();
        }finally {m_Semaphore.release();}
    }

    protected void doUse() throws InterruptedException{
        Log.show("Begin : 사용중인 Resource 개수= "+(m_nPermits-m_Semaphore.availablePermits()));
        Thread.sleep(m_Random.nextInt(500));
        Log.show("Begin : 사용중인 Resource 개수= "+(m_nPermits-m_Semaphore.availablePermits()));
    }

```



```java

}

    static class UserThread extends Thread{
        private final static Random m_Random = new Random(10000);
        private final BoundedResource m_resource;

        public UserThread(BoundedResource resource) {
            m_resource=resource;
        }

        public void run() {
            try {while(true) {
                m_resource.use();
                Thread.sleep(m_Random.nextInt(3000));
            }}catch(InterruptedException e) {}
        }
    }

    public static class Main{

        public static void main(String[] argc) {
            System.out.println("Starting...");
            BoundedResource resource = new BoundedResource(3);
            for(int i=0;i<10;i++) {
                new UserThread(resource).start();
            }
        }
    }

}
```

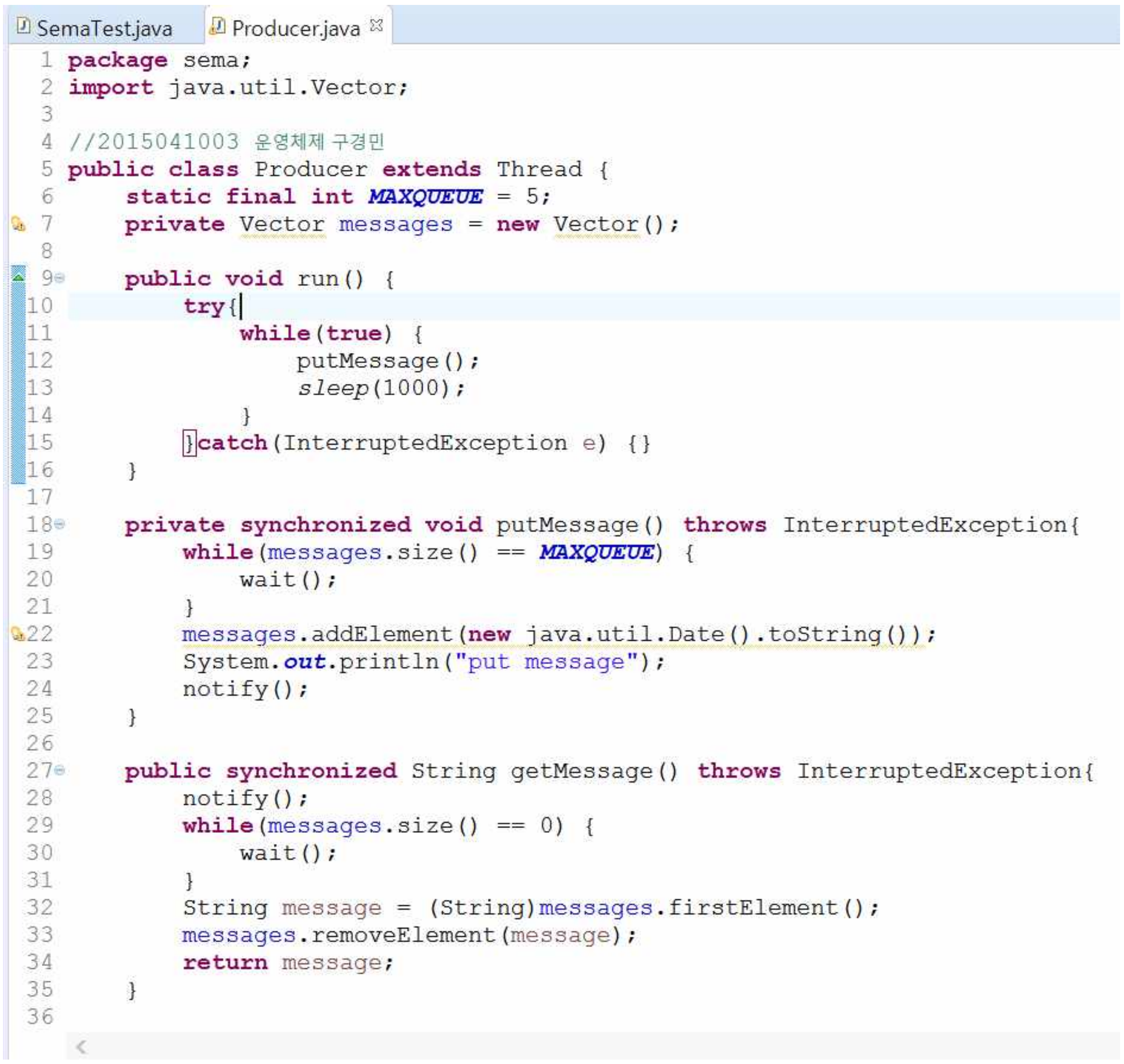
<실행 화면>



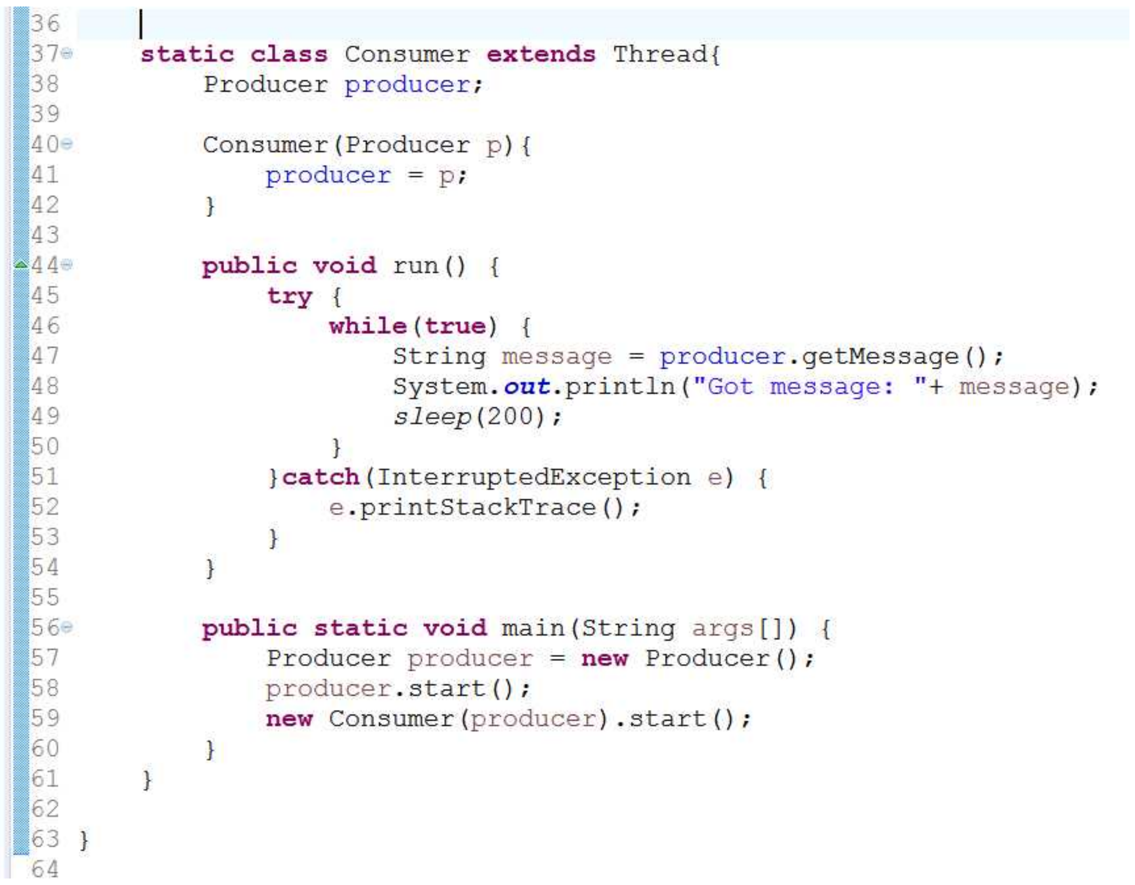
```
SemaTest.Main [Java Application] C:₩Program Files₩Java₩jre1.8
Starting...
Thread-1 : Begin : 사용중인 Resource 개수 = 3
Thread-2 : Begin : 사용중인 Resource 개수 = 2
Thread-3 : Begin : 사용중인 Resource 개수 = 2
Thread-2 : Begin : 사용중인 Resource 개수 = 3
Thread-4 : Begin : 사용중인 Resource 개수 = 3
Thread-3 : Begin : 사용중인 Resource 개수 = 3
Thread-5 : Begin : 사용중인 Resource 개수 = 3
Thread-5 : Begin : 사용중인 Resource 개수 = 3
Thread-9 : Begin : 사용중인 Resource 개수 = 3
Thread-9 : Begin : 사용중인 Resource 개수 = 3
Thread-6 : Begin : 사용중인 Resource 개수 = 3
```

#java 모니터

<소스 코드>

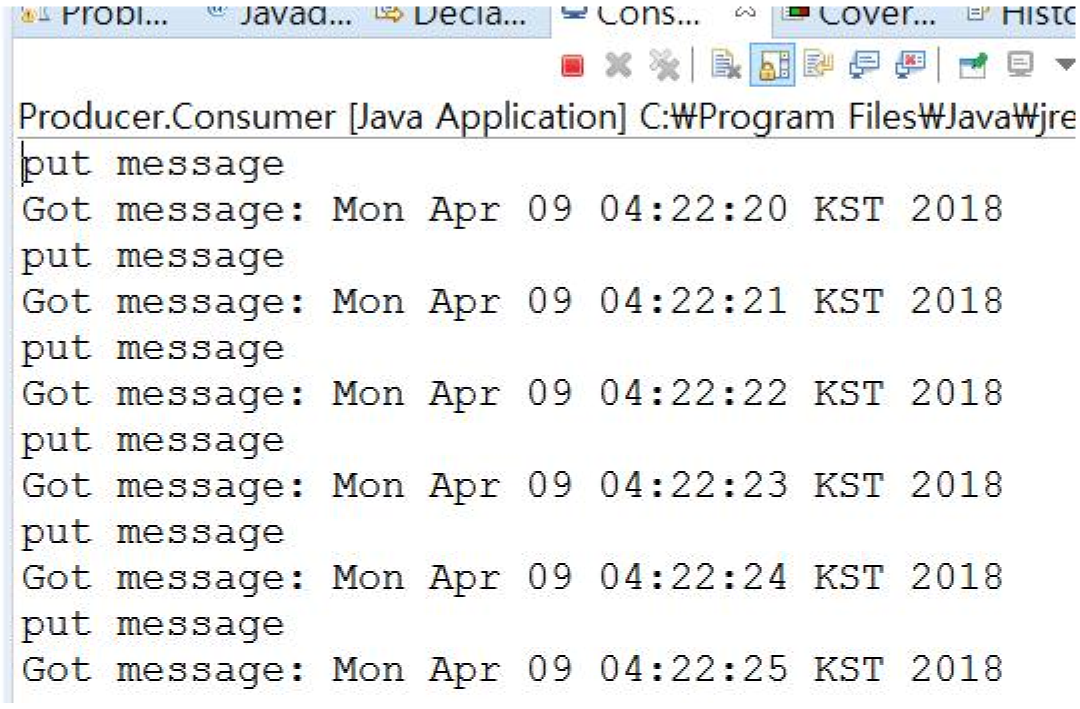


```
package sema;
import java.util.Vector;

//2015041003 운영체제 구경민
public class Producer extends Thread {
    static final int MAXQUEUE = 5;
    private Vector messages = new Vector();

    public void run() {
        try{
            while(true) {
                putMessage();
                sleep(1000);
            }
        }catch(InterruptedException e) {}
    }

    private synchronized void putMessage() throws InterruptedException{
        while(messages.size() == MAXQUEUE) {
            wait();
        }
        messages.addElement(new java.util.Date().toString());
        System.out.println("put message");
        notify();
    }

    public synchronized String getMessage() throws InterruptedException{
        notify();
        while(messages.size() == 0) {
            wait();
        }
        String message = (String)messages.firstElement();
        messages.removeElement(message);
        return message;
    }

```



```java

    static class Consumer extends Thread{
        Producer producer;

        Consumer(Producer p){
            producer = p;
        }

        public void run() {
            try {
                while(true) {
                    String message = producer.getMessage();
                    System.out.println("Got message: "+ message);
                    sleep(200);
                }
            }catch(InterruptedException e) {
                e.printStackTrace();
            }
        }

        public static void main(String args[]) {
            Producer producer = new Producer();
            producer.start();
            new Consumer(producer).start();
        }
    }

}

```

<실행 결과>



Producer.Consumer [Java Application] C:₩Program Files₩Java₩jre

```
put message
Got message: Mon Apr 09 04:22:20 KST 2018
put message
Got message: Mon Apr 09 04:22:21 KST 2018
put message
Got message: Mon Apr 09 04:22:22 KST 2018
put message
Got message: Mon Apr 09 04:22:23 KST 2018
put message
Got message: Mon Apr 09 04:22:24 KST 2018
put message
Got message: Mon Apr 09 04:22:25 KST 2018
```