|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **[유헬스2\_14주\_실습과제]** | **학번:** |  | **이름:** |  |

* 스레드 – 동기화
* critical section이 실행되는 동안 다른 스레드가 공유 데이터를 사용할 수 없도록 만드는 것
* 공유하는 영역에 synchronized 키워드를 추가

class SharedResource {

**long** nF1 = 1, nF2 = 1, nResult = 0;

//임계영역, ***밑줄 친 키워드를 삭제한 후 결과도 확인하여 제출***

**public** **synchronized** **void** changeShared(String szName) {

nResult = nF1 + nF2;

**try**{

nF2 = nF1;

Thread.*sleep*(1000); //시간 지연

nF1 = nResult;

System.*out*.println(szName + ":" + nResult);

}

**catch**(InterruptedException e){}

}

}

class MyThread extends Thread {

SharedResource sharedResource;

String szTName;

**public** MyThread(String szName, SharedResource res) {

**super**(szName);

szTName = szName;

sharedResource = res;

}

**public** **void** run() {

**for**(**int** i = 0;i < 5;i++)

sharedResource.changeShared(szTName);

}

}

public class NotSynchoTest {

public static void main(String args[]) {

SharedResource res = new SharedResource();

MyThread my\_thread1 = new MyThread("thd1", res);

MyThread my\_thread2 = new MyThread("thd2", res);

MyThread my\_thread3 = new MyThread("thd3", res);

my\_thread1.start();

my\_thread2.start();

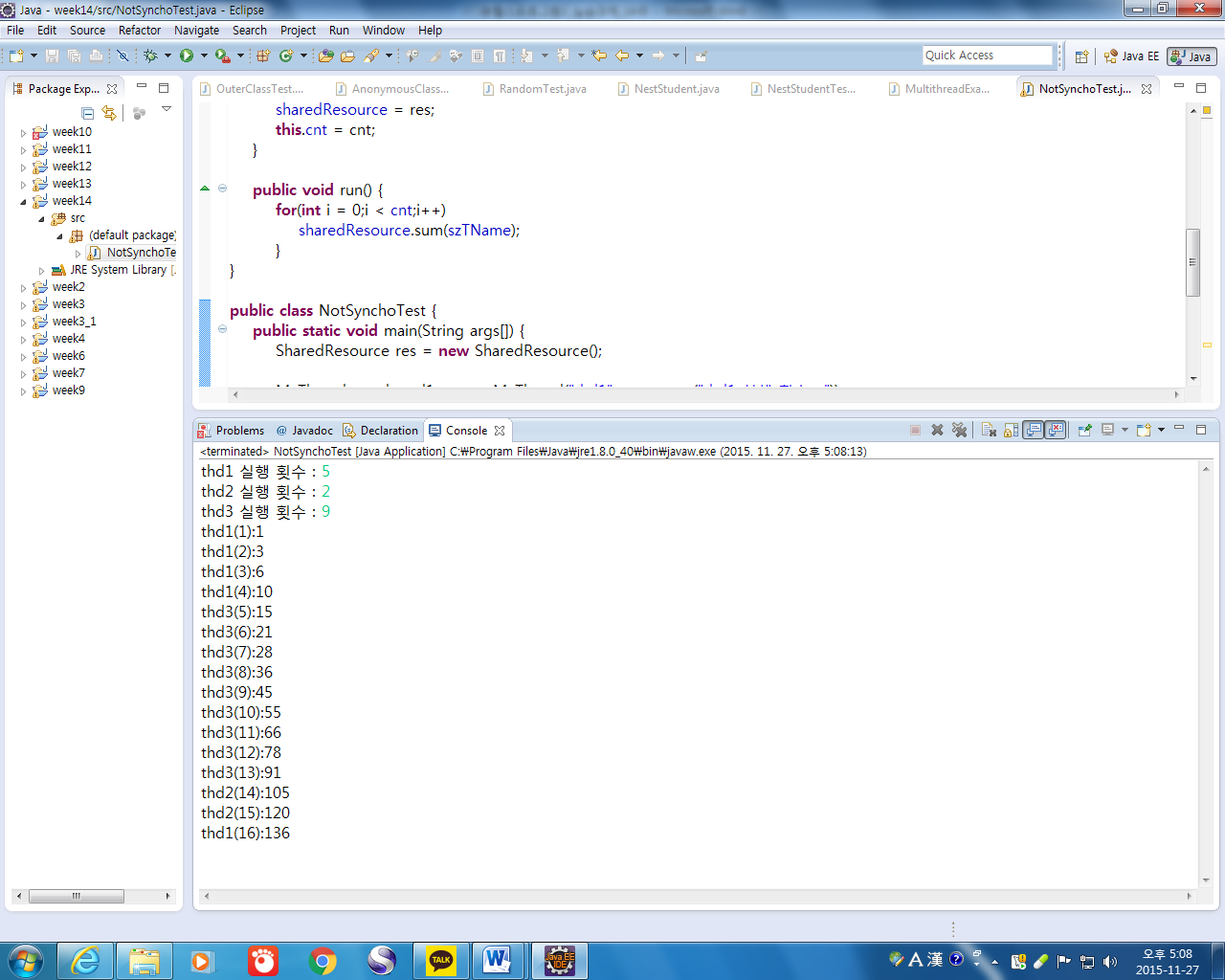
my\_thread3.start();

}

}

|  |
| --- |
| **[실행결과 – 동기화 이전]**  **[실행결과 – 동기화 이후]** |

* **프로그램 과제 1** : 위의 프로그램을 참고하여 제시된 결과처럼 출력되는 프로그램을 작성하시오.



|  |
| --- |
| **[프로그램 소스]** |
| **[실행결과]** |

* 파일 입출력
* 바이트 스트림(byte stream)
* 8비트의 바이트 단위로 입출력하는 클래스
* 이진 데이터를 읽고 쓰기 위하여 사용
* 클래스 이름에 InputStream(입력)과 OutputStream(출력)이 붙는다.
* 문자 스트림(character stream)
* 문자 단위로 입출력하는 클래스
* 문자 스트림은 유니코드 단위로 입출력한다.
* 클래스 이름에 Reader(입력)와 Writer(출력)가 붙는다

======== 문자 스트림 입출력

**import** java.util.\*;

**import** java.io.\*;

**public** **class** FileTextStream {

**public** **static** **void** main(String[] args) **throws** IOException {

Scanner key=**new** Scanner(System.*in*);

FileWriter output = **null**;

FileReader input = **null**;

**int** var;

**try**{

output = **new** FileWriter("output.txt");

**while**(**true**){

System.*out*.print("숫자 입력(종료:-1)");

var=key.nextInt();

**if**(var<0) //음수가 입력되면 입력 중단

**break**;

output.write(var); //한 문자씩 쓰기

}

output.close(); //"output.txt"을 읽어 오는 작업을 위해 닫아줌

input = **new** FileReader("output.txt"); //읽기 작업을 위해 다시 open

System.*out*.println("======파일에 저장된 데이터 출력");

**while** ((var = input.read() ) != -1) //한 문자씩 읽기, read()는 읽어 온 데이터를 int로 반환

System.*out*.println(var);

}

**finally** {

**if** (output != **null**)

output.close();

**if** (input != **null**)

input.close();

}

}

}

|  |
| --- |
| **[실행결과]** |
| **[파일에 저장 된 결과]** |

* **프로그램 과제 2** : “output.txt”파일의 내용을 “copy.txt”에 복사한 후 두 파일의 내용을 출력하는 프로그램을 작성하시오. 문자스트림을 사용할 것

|  |
| --- |
| **[프로그램 소스]** |
| **[실행결과]** |

======== 바이트 스트림 입출력

**import** java.io.\*;

**public** **class** FileByteStream {

**public** **static** **void** main(String[] args) **throws** IOException {

BufferedOutputStream writer = **null**; //바이트 출력 버퍼스트림

BufferedInputStream reader = **null**; //바이트 입력 버퍼스트림

**int** c;

**try** {

writer = **new** BufferedOutputStream(**new** FileOutputStream("output.txt"));

String str="First Java Program";

**for** (**int** cnt = 0; cnt < str.length(); cnt++)

writer.write(str.charAt(cnt));

}

**finally** {

**if**(writer != **null**)

writer.close();

}

**try**{

reader =**new** BufferedInputStream (**new** FileInputStream("output.txt"));

System.***out***.println("======파일에 저장된 데이터 출력");

**while** ((c =reader.read() ) != -1) //read()는 읽어 온 데이터를 정수로 반환

System.***out***.println((**char**)c); //문자출력을 위해서는 형 변환 필요

}

**finally** {

**if** (reader != **null**) { reader.close(); }

}

}

}

|  |
| --- |
| **[실행결과]** |
| **[파일에 저장 된 결과]** |

* **프로그램 과제 3** : 사용자로부터 한글 및 영문자를 입력 받아 “char.txt” 파일에 저장한 후, 파일에 저장된 내용을 표준 출력장치로 출력하는 프로그램을 작성하시오. 단, 버퍼와 문자스트림를 사용할 것

|  |
| --- |
| **[프로그램 소스]** |
| **[실행결과]** |

* 객체 입출력

import java.io.\*; //Serializable 인터페이스를 구현하기 위해 필요

public class PetRecord implements Serializable{

String name; //이름

int age; //나이

double weight; //몸무게

public PetRecord(String name, int age, double weight){

this.name = name;

this.age = age;

this.weight = weight;}

public String toString(){

return "Name = " + name + "\n" + "Age = " + age + "\n" + "Weight = " + weight;

}

}

}

public class ClassIODemo {

public static void main(String[] args) throws IOException, ClassNotFoundException{

ObjectOutputStream out = null; //객체를 파일로 저장

ObjectInputStream in=null; //파일에 저장된 객체 read

try{

out = new ObjectOutputStream(new FileOutputStream("object.dat"));

PetRecord one = new PetRecord("해바라기", 7, 10.2);

PetRecord two = new PetRecord("금강초롱", 4, 4.2);

out.writeObject(one);

out.writeObject(two); //파일에 객체 저장

in = new ObjectInputStream(new FileInputStream("object.dat"));

while(true){

PetRecord obj=(PetRecord)in.readObject(); //파일에서 객체를 읽어 옴

if(obj == null )

break; //파일에서 마지막 데이터를 읽으면 반복문 종료

System.out.println(obj); } //파일에서 읽어 온 객체 출력

}

catch(Exception e){ } //EOFException 예외 처리를 위해 필요

finally {

if(out != null) out.close();

if(in != null) in.close();

}

System.out.println("프로그램 종료");

}

}

|  |
| --- |
| **[실행결과]** |
| **[파일에 저장 된 결과]** |

* **프로그램 과제 4** : 다음과 같은 객체를 ArrayList에 저장한 후, ArrayList 객체를 파일에 저장하는 프로그램을 작성하시오. 단, ArrayList는 Serializable 인터페이스를 구현한다.

**public** class Person {

**private** String name; //이름

**private** String tel; //전화번호

**public** Person(String name, String tel){

this.name=name;

this.tel=tel;

}

@Override

**public** String toString() {

**return** "Person [이름 = " + name + ", 전화번호 = " + tel + "]";

}

**}**

* **프로그램 과제 5** : 과제4의 파일에 저장된 ArrayList를 읽어와 ArrayList에 저장된 원소를 출력하는 프로그램을 작성하시오.

힌트) ArrayList<Book> list=new ArrayList<Book>();

list = (ArrayList<Book>)input.readObject();