|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **[유헬스2\_14주\_실습과제]** | **학번:** | **20175105** | **이름:** | **곽영주** |

* 스레드 중간점검

1. 스레드의 몸체를 포함하는 메소드는 무엇인가?
2. **run();**
3. start();
4. stop();
5. main();
6. 다음은 스레드를 생성하여 실행하는 코드이다. 비어 있는 부분에 어떤 코드를 넣어야 할까?

class Test implements Runnable {

public static void main(String args[]) {

/\* 비어있는 부분 \*/ **Thread th = new Thread(new Test());**

}

public void run() {}

}

1. 다음의 메소드는 어떠한 경우에 사용되는가?

(1) sleep() : **현재의 스레드를 지정된 시간만큼 재운다.**

(2) yield() : **현재 스레드를 다른 스레드에 양보하게 만든다.**

(3) start() : **스레드 시작**

1. 다음 프로그램의 출력은?

public class Test implements Runnable{

public static void main(String [] args){

Test t = new Test();

Thread x = new Thread(t);

x.start();

}

public void run(){

for(int i = 0; i < 3; ++i){

System.out.print(i + "..");

}

}

}

**0..1..2..**

1. 다음 프로그램의 출력은?

public class Test {

private int count = 1;

public synchronized void sub() {

for (int i = 0; i < 10; i++)

System.out.println(count++);

}

public static void main(String[] args) {

Test demo = new Test();

Thread a1 = new A(demo);

Thread a2 = new A(demo);

a1.start();

a2.start();

}

}

class A extends Thread {

Test demo;

public A(Test td) {

demo = td;

}

public void run() {

demo.sub();

}

}

**1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20**

1. 질문에 답하시오

(1) 출력을 예상하여 보라.

**class** Job **implements** Runnable { **extends Thread {**

**String name;**

**public Job(String name) {**

**this.name = name;**

**}**

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

**byte** n = 0;

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

System.*out*.println(**name+**""+n++);

}

}

**public** **class** Test {

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

Thread t = **new** Thread(**new** Job());

**Thread t1 = new Job(“Thread1”);**

**Thread t2 = new Job(“Thread2”);**

t.start();

**t1.start();**

**t2.start();**

}

}

**무한루프**

(2) Thread 클래스를 상속받는 형태로 다시 작성하라.

(3) Job 클래스의 생성자를 작성하고 매개 변수로 스레드의 이름을 문자열 형태로 받도록 하라. 스레드가 실행되면서 스레드의 이름을 출력한다.

System.*out*.println(name+n++);

(4) 동일한 스레드를 하나 더 생성하여 동시에 실행하여 보자. 어떤 출력이 나타나는가? 출력을 구분하기 위하여 스레드의 이름을 다르게 하라.

1. 다음 프로그램의 출력을 예상하여 보라.

**class** MyJob **implements** Runnable {

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

**try** {

System.*out*.println("A");

Thread.*sleep*(1000);

System.*out*.println("B");

} **catch** (InterruptedException e) {

System.*out*.println("C");

}

System.*out*.println("D");

}

}

**public** **class** Test {

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

Thread t = **new** Thread(**new** MyJob());

t.start();

t.interrupt();

}

}

**A C D**

* 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]** |

=========== 바이트 스트림 입출력

//이미지 파일 복사하기 – 이미지 파일은 이진 파일

//적당한 JPG 파일을 사용하여 결과를 확인할 것

//소스 입력 후 Ctrl + Shift + O를 눌러 필요한 파일 포함

**public** **class** ByteStreamsLab {

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

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

System.***out***.print("원본 파일 이름을 입력하시오: ");

String inFileName = scan.next();

System.***out***.print("복사 파일 이름을 입력하시오: ");

String outFileName = scan.next();

**try** (InputStream inStream = **new** FileInputStream(inFileName);

OutputStream outStream = **new** FileOutputStream(outFileName)) {

**int** c;

**while** ((c = inStream.read()) != -1) {

outStream.write(c);

}

}

System.***out***.println(inFileName + "을 " + outFileName + "로 복사하였습니다. ");

}

}

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

* 프로그램 과제 1 : 문자 스트림 입출력 예제에서 생성된 “output.txt”파일의 내용을 “copy.txt”에 복사한 후 두 파일의 내용을 출력하는 프로그램을 작성하시오. 문자 스트림을 사용할 것

|  |
| --- |
| **[프로그램 소스]**  **import** java.io.\*;  **public** **class** problem1 {  **public** **static** **void** main(String[] args) **throws** IOException {  String inFileName = "output.txt";  String outFileName = "copy.txt";    **try** (Reader reader = **new** FileReader(inFileName);  Writer writer = **new** FileWriter(outFileName)) {    **int** a;  **while** ((a = reader.read()) != -1) {  writer.write(a);  }  }  System.***out***.println(inFileName + "을 " + outFileName + "로 복사하였습니다.");    System.***out***.println("=== output 내용 출력 ===");  **int** out;  Reader readerOut = **new** FileReader(inFileName);  **while** ((out = readerOut.read() ) != -1)  System.***out***.println(out);    System.***out***.println("=== copy 내용 출력 ===");  **int** copy;  Reader readerCopy = **new** FileReader(outFileName);  **while** ((copy = readerCopy.read() ) != -1)  System.***out***.println(copy);  }  } |
| **[실행결과]** |

* 프로그램 과제 2 : 두 개의 텍스트 파일을 비교하여서 일치하는지를 검사하는 프로그램을 작성하시오. 바이트 스트림을 사용할 것

|  |
| --- |
| **[프로그램 소스]**  **import** java.io.\*;  **public** **class** problem2 {  **public** **static** **void** main(String[] args) **throws** IOException {  OutputStream FileOut1 = **new** FileOutputStream("A1.txt");  OutputStream FileOut2 = **new** FileOutputStream("A2.txt");  FileOut1.write(111);  FileOut2.write(222);  FileOut1.close();  FileOut2.close();    **try** (InputStream FileIn1 = **new** FileInputStream("A1.txt");  InputStream FileIn2 = **new** FileInputStream("A2.txt")){    **if**(FileIn1.read() != FileIn2.read()) {  System.***out***.println("두 개의 텍스트 내용이 일치하지 않습니다.");  }  } **finally** {  FileOut1.close();  FileOut2.close();  }  }  } |
| **[실행결과]** |

* 스트림 결합하기 – 버퍼 스트림

===== 버퍼 스트림 출력- 라인단위로 읽기

**public** **class** StreamCon {

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

BufferedReader in = **null**;

PrintWriter out = **null**;

**try** {

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

out.println("변화를 원한다면,");

out.println("제일 먼저 자신이 변화할 수 있다는 것과");

out.println("변화하기까지 포기하지 않고");

out.println("계속해서 노력할 수 있다는 것을 믿어야 한다.");

out.flush(); **//버퍼를 비움**

in = **new** BufferedReader(**new** FileReader("output.txt")); **//버퍼 스트림과 연결**

String line;

**while** ((line = in.readLine()) != **null**) { **// 라인단위로 읽어옴**

System.***out***.println(line);

}

} **finally** {

**if** (in != **null**)

in.close();

**if** (out != **null**)

out.close();

}

}

}

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

* 프로그램 과제 3 : “output.txt” 텍스트 파일을 라인단위로 읽어서 앞에 라인 번호를 붙여서 출력하는 프로그램을 작성하시오

|  |
| --- |
| **[프로그램 소스]**  **import** java.io.\*;  **public** **class** problem3 {  **public** **static** **void** main(String[] args) **throws** IOException {  BufferedReader in = **null**;  **int** lineNum = 1;    **try** {  in = **new** BufferedReader(**new** FileReader("output.txt"));  String line;  **while**((line = in.readLine()) != **null**) {  System.***out***.println(lineNum + "] " + line);  lineNum++;  }  } **finally** {  **if**(in != **null**)  in.close();  }  }  } |
| **[실행결과]** |

* 프로그램 과제 4: 2개의 텍스트 파일을 합하여 하나의 파일로 만드는 프로그램을 작성하시오. 라인단위로 읽어오며 라인단위로 파일에 출력

|  |
| --- |
| **[프로그램 소스]**  **import** java.io.\*;  **public** **class** problem4 {  **public** **static** **void** main(String[] args) **throws** IOException {  FileWriter fw1 = **new** FileWriter("file1.txt");  FileWriter fw2 = **new** FileWriter("file2.txt");  fw1.write("안녕하세요.");  fw2.write("곽영주입니다.");  fw1.close();  fw2.close();    BufferedReader fr1 = **null**;  BufferedReader fr2 = **null**;  BufferedWriter bw = **null**;    **try** {  fr1 = **new** BufferedReader(**new** FileReader("file1.txt"));  fr2 = **new** BufferedReader(**new** FileReader("file2.txt"));  bw = **new** BufferedWriter(**new** FileWriter("hapFile.txt"));    String f1;  **while**((f1 = fr1.readLine()) != **null**)  bw.write(f1);    String f2;  **while**((f2 = fr2.readLine()) != **null**)  bw.write(f2);    bw.flush();  } **finally** {  **if**(fr1 != **null**)  fr1.close();  **if**(fr2 != **null**)  fr2.close();  **if**(bw != **null**)  bw.close();  }  BufferedReader br = **new** BufferedReader(**new** FileReader("hapFile.txt"));  System.***out***.println(br.readLine());  }  } |
| **[실행결과]** |

* 기초 자료형 읽고 쓰기 : 쓸때와 동일한 순서로 읽어야 함

public class DataStreamTest {

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

DataInputStream in = null;

DataOutputStream out = null;

try {

int c;

//파일 스트림->버퍼 스트림->데이터 스트림

out = new DataOutputStream(new BufferedOutputStream(

new FileOutputStream("data.bin")));

out.writeDouble(3.14); out.writeInt(100);

out.writeUTF("자신의 생각을 바꿀 수 없다.");

out.flush();

in = new DataInputStream(new BufferedInputStream(

new FileInputStream("data.bin")));

System.out.println( in.readDouble() ) ;

System.out.println( in.readInt() ); System.out.println( in.readUTF() );

} finally {

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

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

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

* 프로그램 과제 5: 기본 자료형인 int, short, long, float, double형의 데이터를 출력하는 문장을 작성해보라. 또 반대로 int, short, long, float, double형의 데이터를 입력하는 문장을 작성하고 실행하시오

|  |
| --- |
| **[프로그램 소스]**  **import** java.io.\*;  **public** **class** problem5 {  **public** **static** **void** main(String[] args) **throws** IOException {  DataInputStream in = **null**;  DataOutputStream out = **null**;    **try** {  out = **new** DataOutputStream(**new** BufferedOutputStream(**new** FileOutputStream("problem5.bin")));  out.writeInt(123);  out.writeShort(13);  out.writeLong(367L);  out.writeFloat(12.3f);  out.writeDouble(2.3);  out.flush();    in = **new** DataInputStream(**new** BufferedInputStream(**new** FileInputStream("problem5.bin")));  System.***out***.println(in.readInt());  System.***out***.println(in.readShort());  System.***out***.println(in.readLong());  System.***out***.println(in.readFloat());  System.***out***.println(in.readDouble());  } **finally** {  **if**(in != **null**)  in.close();  **if**(out != **null**)  out.close();  }  }  } |
| **[실행결과]** |

* 스캐너 클래스 기초자료형 읽기

import java.io.\*;

import java.util.\*;

public class ScanTest2 {

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

Scanner s = null; PrintWriter out = null;

double sum = 0;

//PrintWriter 클래스 : 문자열이나 각종 수치값을 텍스트 형식으로 출력

out = new PrintWriter(new FileWriter("output.txt"));

out.println("9.5"); // "output.txt”파일에 “9.5” 저장;

out.println("567,000"); out.println("string"); out.println("4");

out.flush(); //버퍼를 수동으로 비움

s = new Scanner(new BufferedReader(new FileReader("output.txt")));

while (s.hasNext()) {

if (s.hasNextDouble()) {

sum += s.nextDouble(); //토큰을 기초 자료형으로 변환

} else { s.next(); }}

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

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

System.out.println(sum);}}

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

* 객체 입출력

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

public class PetRecord implements Serializable{ //Serializable 인터페이스를 구현해야 직렬화 가능

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); //또는 out.writeObject(new PetRecord(“해바라기”, 7, 10.2);

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 6: 다음과 같은 객체를 파일에 저장한 후 출력하는 프로그램을 작성하시오.

**import** java.io.\*; // Serializable 구현을 위해 반드시 필요

**public** **class** Book **implements** Serializable {

**private** String name; // 도서명

**private** String author; // 저자명

**private** String number; //관리번호

**public** Book(String name, String author, String number){

**this**.name=name;

**this**.author=author;

**this**.number=number;

}

**public** **void** disPlay(){

System.*out*.print("도서명 : " + name);

System.*out*.print("\t저자명 : " + author);

System.*out*.println("\t관리번호 : " + number);

}

}

|  |
| --- |
| **[프로그램 소스]**  **import** java.io.\*;  **public** **class** Book **implements** Serializable {  **private** String name;  **private** String author;  **private** String number;    **public** Book(String name, String author, String number) {  **this**.name = name;  **this**.author = author;  **this**.number = number;  }    **public** **void** disPlay() {  System.***out***.println("도서명 : " + name);  System.***out***.println("\t저자명 : " + author);  System.***out***.println("\t관리번호 : " + number);  }  }  **import** java.io.\*;  **public** **class** BookTest {  **public** **static** **void** main(String[] args) **throws** IOException, ClassNotFoundException {  ObjectOutputStream out = **null**;  ObjectInputStream in = **null**;  **try** {  out = **new** ObjectOutputStream(**new** FileOutputStream("Book.dat"));  Book b1 = **new** Book("자서전","곽영주","20175105");  Book b2 = **new** Book("책","김재환","20175123");  out.writeObject(b1);  out.writeObject(b2);    in = **new** ObjectInputStream(**new** FileInputStream("Book.dat"));  **while**(**true**) {  Book obj = (Book)in.readObject();  **if**(obj == **null**)  **break**;  obj.disPlay();  }  } **catch**(Exception e) {    } **finally** {  **if**(out != **null**)  out.close();  **if**(in != **null**)  in.close();  }  }  } |
| **[실행결과]** |