|  |  |  |
| --- | --- | --- |
| **2019\_2\_java2\_15** | **학번 : 20135151** | **이름 : 이갑성** |

* **내용 점검**

1. 기초 자료형 입출력

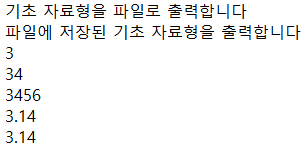
|  |
| --- |
| import java.io.\*;  public class DataStreamTest {  public static void main(String[] args) throws IOException {  DataInputStream in = null; DataOutputStream out = null;  try {  int c;  **//파일 스트림->버퍼 스트림->데이터 스트림**  out = new DataOutputStream(new BufferedOutputStream(  new FileOutputStream("data.bin")));  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")));  **//파일에 저장된 순서대로 읽어 와야 함**  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(); }}}} |
| **[실행 결과]** |

1. 객체 입출력

|  |
| --- |
| **import** java.io.\*; //Serializable 인터페이스를 구현하기 위해 필요  **import** java.util.\*;  **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;  }  **void** disPlay() {  System.***out***.println("Name = " + name + "\n" + "Age = " + age + "\n" + "Weight = " + weight);  }  }  **public** **class** objectFile {  **public** **static** **void** main(String[] args) **throws** IOException, ClassNotFoundException{  ObjectOutputStream out = **null**;  ObjectInputStream in=**null**;  ArrayList<PetRecord> list = **new** ArrayList<>();  **try**{  out = **new** ObjectOutputStream(**new** FileOutputStream("object.dat"));  PetRecord one = **new** PetRecord("해바라기", 7, 10.2);  PetRecord two = **new** PetRecord("금강초롱", 4, 4.2);  list.add(one);  list.add(two);  out.writeObject(list); //ArrayList 객체를 파일에 저장  in = **new** ObjectInputStream(**new** FileInputStream("object.dat"));  list=(ArrayList<PetRecord>)in.readObject(); //파일에서 객체를 읽어 옴  **for**(PetRecord obj : list)  obj.disPlay();  }  **finally** {  **if**(out != **null**) out.close();  **if**(in != **null**) in.close(); }  System.***out***.println("프로그램 종료");  }  } |
| **[실행결과]** |

* **실습 과제**

1. 기본 자료형인 int, short, long, float, double형의 데이터를 출력하는 문장을 작성해보라. 또 반대로 int, short, long, float, double형의 데이터를 입력하는 문장을 작성하고 실행 하시오



|  |
| --- |
| **[프로그램 소스]**  import java.io.\*;  public class fileTest {  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.writeInt(3);  out.writeShort(34);  out.writeLong(3456);  out.writeFloat(3.14f);  out.writeDouble(3.14);  out.flush();  System.***out***.println("기초 자료형을 파일로 출력합니다.");  System.***out***.println("파일에 저장된 기초 자료형을 출력합니다.");  in = new DataInputStream(new BufferedInputStream(new FileInputStream("data.bin")));  // 파일에 저장된 순서대로 읽어 와야 함  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]** |

1. 다음과 같은 객체를 파일에 저장한 후 출력하는 프로그램을 작성하시오.

* 파일에 저장된 ArrayList 객체를 읽어 리스트에 저장된 객체를 출력한다
* 프로그램을 종료하기 전 ArrayList를 파일에 저장한다
* 객체는 ArrayList에 저장
* 내용이 없는 “object.dat” 파일을 메모장에서 만든 후 실행할 것

힌트) 파일에 저장된 데이터가 없을 경우 처리

**try**(ObjectInputStream in = **new** ObjectInputStream(**new** FileInputStream("object.dat"))){

………

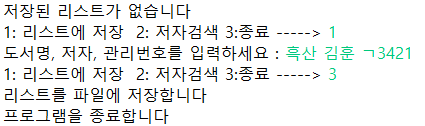
}

**catch**(EOFException e) { **//파일에 저장된 데이터가 없으면 EOFException 예외 발생**

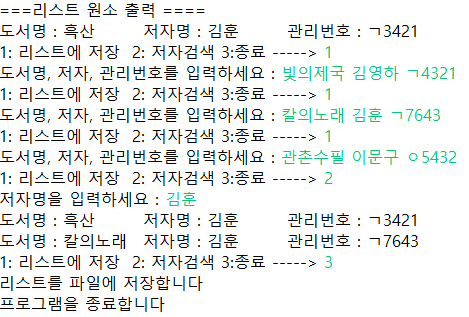
System.***out***.println("저장된 리스트가 없습니다");

}

[프로그램을 최초 실행 했을 때 결과]



[2번째 이후 실행 했을 때 결과]



**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.\*;  import java.util.\*;  public class Test {  public static void main(String[] args) throws IOException, ClassNotFoundException {  Scanner sc = new Scanner(System.***in***);  ObjectOutputStream out = null;  ArrayList<Book> list = new ArrayList<>();  int menu;  boolean flag = true;  String writer;  boolean find = false;  try (ObjectInputStream in = new ObjectInputStream(new FileInputStream("object.dat"))) {  list = (ArrayList<Book>) in.readObject(); // 파일에서 객체를 읽어 옴  for (Book obj : list)  obj.disPlay();  } catch (EOFException e) { // 파일에 저장된 데이터가 없으면 EOFException 예외 발생  System.***out***.println("저장된 리스트가 없습니다");  } finally {  }  try {  while (flag) {  System.***out***.print("1: 리스트에 저장 2: 저자검색 3: 종료 --------->");  menu = sc.nextInt();  switch (menu) {  case 1:  out = new ObjectOutputStream(new FileOutputStream("object.dat"));  System.***out***.print("도서명, 저자, 관리번호를 입력하세요: ");  Book b = new Book(sc.next(), sc.next(), sc.next());  list.add(b);  out.writeObject(list); // ArrayList 객체를 파일에 저장  break;  case 2:  System.***out***.print("저자명을 입력하세요:");  writer = sc.next();  for (Book result : list) {  if (result.author.equals(writer)) {  result.disPlay();  find = true;  }    }  if(find == false) {  System.***out***.println(writer + "에 관한 도서를 찾을 수 없습니다.");  }  break;  case 3:  flag = false;  System.***out***.println("리스트를 파일에 저장합니다.");  break;  }  }  } finally {  if (out != null)  out.close();  System.***out***.println("프로그램 종료");  }  }  }  import java.io.\*; // Serializable 구현을 위해 반드시 필요  public class Book implements Serializable {  String name; // 도서명  String author; // 저자명  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);  }  } |
| **[실행결과].** |