₩ KH 정보교육원 국비대상자 간편조회 | 국비무료교육 온라인상담 국비지원과정 교육원소개 취업지원센터 채용정보 프로젝트 커뮤니티 마이페이지 상담센터 산학연계 김윤수 스마트 콘텐츠 융합 응용SW 엔지니어 양성과정(888H) [강남 M] 2020. 12.02 ~ 2021.07.08 15:30~22:00 김동현 강사 | 진은지 취업담임 남은 시간 11:56:56 수강생 평가 \* 첨부파일의 확장자를 소문자로 등록하세요. - 예시: test.jpg(O), test.JPG(X) [NCS전공교과] 네트워크 프로그래밍 구현 (문제해결시나리오) 1회차 ~ 총점: 100.0 1. 2001020508\_14v2.1 개발환경 분석하기 점수 (10.0점) TCP 소켓을 사용하여 클라이언트에서 서버로 연결처리하는 네트워크 프로그램을 구현하고 구동하였으나 서버와 연결이 되지 않았다 예상 실행결과가 정확히 도출될 수 있도록 문제점이 있는 코드를 찾아 [원인](5점)에 파악된 문제점을 기술하고, 해결 방안을 구상하여 [조치내용](5점)에 해결방법을 작성하여 보고서를 완성하시오. (10점) '-서버 IP: 192.168.20.34, 포트번호: 3000 [클라이언트용 프로그래밍 코드] public void fileUpload() { int port = 3000; String serverIP; serverIP = InetAddress.getLocalHost().getHostAddress(); // 2. 소켓 객체 생성: 생성과 동시에 서버에 연결 요청됨 Socket socket = new Socket(serverIP, port); // 연결 실패는 null 리턴됨 //이하 코드 중간 생략..... [서버용 프로그래밍 코드] public class ServerManager { public ServerManager() {} public void serverStart() { // 1. port 번호 지정 int port = 3000; // 2. 서버용 소켓 객체 생성 ServerSocket server; server = new ServerSocket(port); // 3. 클라이언트가 연결을 요청할 때까지 기다림 while (true) { // 4. 연결을 요청한 클라이언트의 요청 수락함: 해당 클라이언트 정보를 저장 Socket client = server.accept(); // 중간 생략.... Client부분에서 수강생 답 String serverIP부분이 초기화 되어있지않다 (원인) 현재는 그럼 String serverIp = null 이므로 당연히 서버와 연결될수없다. Client부분에서 String serverIP부분이 초기화 되어있지않다 답안첨삭 현재는 그럼 String serverip = null 이므로 당연히 서버와 연결될수없다. 요구된 사항이 맞지 않거나 문법적인 오류에 대해 일부만 파악하엿음 명시된 서버의 IP주소가 사용되지 않았음. 모범답안 Socket socket = new Socket(serverIP, port); Client부분에서 String타입변수 serverIP값을 제대로 초기화 해준다 수강생 답 String serverIP = "192.168.20.34"; (조치내용) Client부분에서 String타입변수 serverIP값을 제대로 초기화 해준다 String serverIP = "192.168.20.34"; 답안첨삭 파악된 오류 원인에 대한 해결 조치를 정확하게 제시하였음 Socket socket = new Socket("195.168.20.34", port); 모범답안 로 수정하면 서버와 연결됨 성취기준 채점기준 10 문법 오류를 조금 이해하고 부분적으로 수정할 수 있으며, 요구사항을 잘 반영할 수 있다. 평가항목 기 작성된 코드를 분석하여 이해하는 능력이 부족하고, 문법오류를 조금 수정할 수 있다. 5~9 기 작성된 코드를 분석하고 이해하는 능력이 많이 부족하고 문법 오류를 수정할 수 없다. 0~4 2. 2001020508\_14v2.2 기능 구현하기 점수 (40.0점) 파일에 객체 정보를 저장하는 프로그램을 구현하려고 한다. 사용되는 Book 클래스를 [원인](10점)에 기술하고, 파일에 데이터로 기록 저장하는 코드를 [조치내용](30점)에 작성하시오. (40점) [클라이언트용 프로그래밍 실행 순서] [원인]: kh.model.vo.Book 클래스를 캡슐화(EnCapsulation) 원칙에 맞춰서 구현하시오. - 직렬화 선언한다. - String타입의 title과 int타입의 price 필드를 선언한다 - 기본생성자와 매개변수 있는 생성자를 작성한다. - 필드에 대한 getter 와 setter 작성한다. - toString() 메소드를 오버라이딩하여 인스턴스변수의 값에 대한 문장을 작성하여 리턴한다. [조치내용]: fileSave() 메소드를 작성하시오.: public void fileSave(String fileName){} - 전달받은 fileName 으로 파일 객체를 생성한다. - 파일 객체와 연결하는 파일 출력스트림을 생성한다. - 파일에 Book 객체를 기록할 수 있는 객체 출력스트림을 추가한다. - 파일에 Book 객체 정보를 기록하고 스트림들을 닫는다. - Book 객체 샘플: "자바를 정복하자", 35000 package kh.model.vo; import java.io.Serializable; public class Book implements Serializable{ private String title; private int price; public String getTitle() { return title; public void setTitle(String title) { this.title = title; public int getPrice() { return price; 수강생 답 public void setPrice(int price) { (원인) this.price = price; public Book() { super(); public Book(String title, int price) { super(); this.title = title; this.price = price; @Override public String toString() { return "Book [title=" + title + ", price=" + price + "]"; package kh.model.vo; import java.io.Serializable; public class Book implements Serializable{ private String title; private int price; public String getTitle() { return title; public void setTitle(String title) { this.title = title; public int getPrice() { return price; public void setPrice(int price) { 답안첨삭 this.price = price; public Book() { super(); public Book(String title, int price) { super(); this.title = title; this.price = price; @Override public String to String() { return "Book [title=" + title + ", price=" + price + "]"; 프로그래밍 방법론 및 개발방법 기준에 따라서 네트워크 응용프로그램을 구현할 수 있다. package kh.model.vo; public class Book implements java.io. Serializable{ private String title; private int price; public Book(){} public Book(String title, int price){ this.title = title; this.price = price; public void setTitle(String title){ this.title = title; public String getTitle(){ 모범답안 return title; public void setPrice(int price){ this.price = price; public int getPrice(){ return price; @Override public String toString(){ return title + ", " + price; public void fileSave(String fileName) { File f = new File(fileName); //아래처럼 ObjectOutputStream을 선언해주면 close()를 해주지 않아도 자동반납 try (ObjectOutputStream oos = new ObjectOutputStream( new BufferedOutputStream( new FileOutputStream(f)))) 수강생 답 (조치내용) Book b = new Book("자바를 정복하자", 35000); oos.writeObject(b); } catch (Exception e) { public void fileSave(String fileName) { File f = new File(fileName); //아래처럼 ObjectOutputStream을 선언해주면 close()를 해주지 않아도 자동반납 try (ObjectOutputStream oos = new ObjectOutputStream( new BufferedOutputStream( new FileOutputStream(f)))) 답안첨삭 Book b = new Book("자바를 정복하자", 35000); oos.writeObject(b); } catch (Exception e) { 프로그래밍 방법론 및 개발방법 기준에 따라서 네트워크 응용프로그램을 구현할 수 있다. // 파일에 객체를 저장하기 위한 메소드 public void fileSave(String fileName) { try (ObjectOutputStream objOut = new ObjectOutputStream(new FileOutputString(fileName));){ objOut.writeObject(new Book("자바를 정복하자", 35000)); 모범답안 }catch(Exception e){ e.printStackTrace(); } //fileSave() close 성취기준 채점기준 개발방법 기준 및 프로그래밍 방법론에 따라서 설계내용을 바탕으로 네트워크 프로그래밍 응용프로그램을 잘 구현할 수 있다. 40 35~39 개발방법 기준 및 프로그래밍 방법론에 따라서 설계내용을 바탕으로 네트워크 프로그래밍 응용프로그램을 대부분 구현할 수 있다. 30~34 개발방법 기준 및 프로그래밍 방법론에 따라서 설계내용을 바탕으로 네트워크 프로그래밍 응용프로그램을 구현할 수 있다. 개발방법 기준 및 프로그래밍 방법론에 따라서 설계내용을 바탕으로 네트워크 프로그래밍 응용프로그램을 대략적으로 구현할 수 있다. 25~29 평가항목 20~24 개발방법 기준 및 프로그래밍 방법론에 따라서 설계내용을 바탕으로 네트워크 프로그래밍 응용프로그램을 대략적으로 일부 구현할 수 있다. 15~19 개발방법 기준 및 프로그래밍 방법론에 따라서 설계내용을 바탕으로 네트워크 프로그래밍 응용프로그램을 부분적으로 구현할 수 있다. 10~14 개발방법 기준 및 프로그래밍 방법론에 따라서 설계내용을 바탕으로 네트워크 프로그래밍 응용프로그램을 일부 조금 구현할 수 있다. 5~9 개발방법 기준 및 프로그래밍 방법론에 따라서 설계내용을 바탕으로 네트워크 프로그래밍 응용프로그램을 조금 구현할 수 있다.  $0 \sim 4$ 개발방법 기준 및 프로그래밍 방법론에 따라서 설계내용을 바탕으로 네트워크 프로그래밍 응용프로그램을 전혀 구현할 수 없다. 3. 2001020508\_14v2.3 프로그램 디버깅하기 점수 (25.0점) ArrayList 를 사용하여 Book 객체 여러개의 값들을 추가 저장하는 프로그램을 구현하기 위해 다음의 요구하는 기능에 대한 메소드를 작성하시오. 요구된 ArrayList 클래스에 Book 만 저장할수 있도록 제네릭스를 사용하는 객체 생성구문을 [원인](5점)에 기술하고, 요구된 메소드 코드를 [조치내용](20점)에 기술하시오. (25점) [클라이언트용 프로그래밍 실행 순서] + fileSave(): void => 2번 문항의 [원인]에 기술한 Book 객체만 저장할 수 있는 ArrayList bookList 객체를 생성한다. ⇒ bookList 에 아래의 5개의 Book 객체를 리스트에 저장한다. Book("자바를 잡아라", 30000)); Book("오라클 정복", 35000)); Book("웹표준 2.0", 27500)); Book("자바 Servlet/JSP", 28000)); Book("ajax 사용법", 15000)); ⇒ 저장된 bookList 정보 출력 확인한다. 수강생 답 List<Book> list = new ArrayList<>(); (원인) List<Book> list = new ArrayList<>(); 답안첨삭 사용되는 파일, 객체 입출력에 대한 클래스와 설명을 정확하게 기술함 모범답안 ArrayList<Book> bookList = new ArrayList<Book>(); public void fileSave() { List<Book> list = new ArrayList<>(); list.add(new Book("자바를 잡아라", 30000)); list.add(new Book("오라클 정복", 35000)); 수강생 답 list.add(new Book("웹표준 2.0", 27500)); list.add(new Book("자바 Servlet/JSP", 28000)); (조치내용) list.add(new Book("ajax 사용법", 15000)); System.out.println(list); public void fileSave() { List<Book> list = new ArrayList<>(); list.add(new Book("자바를 잡아라", 30000)); list.add(new Book("오라클 정복", 35000)); list.add(new Book("웹표준 2.0", 27500)); list.add(new Book("자바 Servlet/JSP", 28000)); 답안첨삭 list.add(new Book("ajax 사용법", 15000)); System.out.println(list); 구축 계획에 따라, 자원관리를 위하여 데이터 저장관리에 대한 파일관리의 기초 기술을 적용하여 정확 하게 프로그램을 구현할 수 있다. // Book 객체 파일 출력용 public void fileSave() { ArrayList<Book> bookList = new ArrayList<Book>(); bookList.add(new Book("자바를 잡아라", 30000)); bookList.add(new Book("오라클 정복", 35000)); 모범답안 bookList.add(new Book("웹표준 2.0", 27500)); bookList.add(new Book("자바 Servlet/JSP", 28000)); bookList.add(new Book("ajax 사용법", 15000)); System.out.println(bookList); 성취기준 채점기준 25 프로그램 방법론과 구축 계획에 따라 자원관리를 위하여 네트워크 연관 응용프로그램을 잘 구현할 수 있다. 20~24 프로그램 방법론과 구축 계획에 따라 자원관리를 위하여 네트워크 연관 응용프로그램을 대부분 구현할 수 있다. 평가항목 15~19 프로그램 방법론과 구축 계획에 따라 자원관리를 위하여 네트워크 연관 응용프로그램을 대략적으로 구현할 수 있다. 10~14 프로그램 방법론과 구축 계획에 따라 자원관리를 위하여 네트워크 연관 응용프로그램을 일부만 구현할 수 있다. 5~9 프로그램 방법론과 구축 계획에 따라 자원관리를 위하여 네트워크 연관 응용프로그램을 일부 조금 구현할 수 있다. 프로그램 방법론과 구축 계획에 따라 자원관리를 위하여 네트워크 연관 응용프로그램을 구현할 수 없다.  $0 \sim 4$ 4. 2001020508\_14v2.4 프로그램 최적화하기 점수 (25.0점) ArrayList를 이용해 다수의 Fruit객체를 저장하였다. ([그림1]참고) 값 삽입 후 삭제를 하려고 했을 때 원하는 대로 삭제가 되지 않았다. ([그림2] 참고) 이에 대한 이유를 [원인](10점)에 기술하고, 이에 대한 해결 방법이 담긴 Fruit 객체를 클래스 다이어그램을 참고하여 [조치내용](15점)에 기술하시오. (25점) package com.kh.test.fruit.controller; 3⊖ import java.util.ArrayList; 5 import com.kh.test.fruit.model.vo.Fruit; 7 public class FruitTest { public static void main(String[] args) { ArrayList<Fruit> list = new ArrayList<Fruit>(); 10 list.add(new Fruit("사과", "달콤")); list.add(new Fruit("으렌지", "상큼")); list.add(new Fruit("키위", "상큼")); 11 12 13 14 15 16 17 System.out.println("값 삽입: " + list); list.remove(1); System.out.println("값 삭제1 : " + list); 18 list.remove(new Fruit("사과", "달콤")); System.out.println("값 삭제2 : " + list); 19 20 21 22 } [그림2] 값 삽입 : [Fruit [name=사과, flavor=달콤], Fruit [name=오렌지, flavor=상쿰], Fruit [name=키위, flavor=상쿰]] 값 삭제1 : [Fruit [name=사과, flavor=달콤], Fruit [name=키위, flavor=상콤]] 값 삭제2 : [Fruit [name=사과, flavor=달콤], Fruit [name=키위, flavor=상쿰]] [클래스 다이어그램] com.kh.test.fruit.model.vo.Fruit - name : String - flavor : String + Fruit() + Fruit (name:String, flavor:String) + setter/getter + toString():String 객체를 비교할때는 Object클래스의 equals()와 hashCode()메서드를 재정의 해줘야하는데 수강생 답 현재 Fruit객체는 두개의 메서드가 재정의 되어있지 않아 두 객채의 동등함을 비교하지 못해서 발생한 문제이다 객체를 비교할때는 Object클래스의 equals()와 hashCode()메서드를 재정의 해줘야하는데 현재 Fruit객체는 두개의 메서드가 재정의 되어있지 않아 답안첨삭 두 객채의 동등함을 비교하지 못해서 발생한 문제이다 요구된 사항이 맞지 않거나 문법적인 오류에 대해 정확하게 파악하엿음 모범답안 equals를 오버라이딩 하지 않아 주소 값으로 비교를 하고 있는데 new연산자로 새로운 주소 값을 갖고 있으니 다른 객체로 인식함. Fruit객체 클래스 안에서 equals, hashCode 메서드 Override package com.kh.test.fruit.model.vo; import java.util.Objects; public class Fruit { private String name; private String flavor; public Fruit() { super(); public Fruit(String name, String flavor) { super(); this.name = name; this.flavor = flavor; public String getName() { return name; public void setName(String name) { this.name = name; public String getFlavor() { 수강생 답 return flavor; (조치내용) public void setFlavor(String flavor) { this.flavor = flavor; @Override public String toString() { return "Fruit [name=" + name + ", flavor=" + flavor + "]"; @Override public boolean equals(Object other) { if(!(other instanceof Fruit)) return false; if(this.name.equals(((Fruit)other).name) && this.flavor.equals(((Fruit)other).flavor)) return true; else return false; @Override public int hashCode() { return Objects.hash(name,flavor); Fruit객체 클래스 안에서 equals, hashCode 메서드 Override package com.kh.test.fruit.model.vo; import java.util.Objects; public class Fruit { private String name; private String flavor; public Fruit() { super(); public Fruit(String name, String flavor) { super(); this.name = name; this.flavor = flavor; public String getName() { return name; public void setName(String name) { this.name = name; public String getFlavor() { return flavor; 답안첨삭 public void setFlavor(String flavor) { this.flavor = flavor; @Override public String toString() { return "Fruit [name=" + name + ", flavor=" + flavor + "]"; @Override public boolean equals(Object other) { if(!(other instanceof Fruit)) return false; if(this.name.equals(((Fruit)other).name) && this.flavor.equals(((Fruit)other).flavor)) return true; else return false; @Override public int hashCode() { return Objects.hash(name,flavor); 파악된 오류 원인에 대한 해결 조치를 정확하게 제시하였음 package com.kh.test.fruit.model.vo; public class Fruit { private String name; private String flavor; public Fruit() {} public Fruit(String name, String flavor) { super(); this.name = name; this.flavor = flavor; public String getName() { return name; public void setName(String name) { this.name = name; public String getFlavor() { return flavor; public void setFlavor(String flavor) { this.flavor = flavor; @Override public String toString() { return "Fruit [name=" + name + ", flavor=" + flavor + "]"; @Override public int hashCode() { 모범답안 final int prime = 31; int result = 1; result = prime \* result + ((flavor == null) ? 0 : flavor.hashCode()); result = prime \* result + ((name = null) ? 0 : name.hashCode()); return result; @Override public boolean equals(Object obj) { if (this == obj) return true; if (obj = null)return false; if (getClass() != obj.getClass()) return false; Fruit other = (Fruit) obj; if (flavor == null) { if (other.flavor != null) return false; } else if (!flavor.equals(other.flavor)) return false; if (name == null) { if (other.name != null) return false; } else if (!name.equals(other.name)) return false; return true; 성취기준 채점기준 25 프로그램 방법론과 구축 계획에 따라 자원관리를 위하여 네트워크 연관 응용프로그램을 잘 구현할 수 있다. 20~24 프로그램 방법론과 구축 계획에 따라 자원관리를 위하여 네트워크 연관 응용프로그램을 대부분 구현할 수 있다. 평가항목 15~19 프로그램 방법론과 구축 계획에 따라 자원관리를 위하여 네트워크 연관 응용프로그램을 대략적으로 구현할 수 있다. 10~14 프로그램 방법론과 구축 계획에 따라 자원관리를 위하여 네트워크 연관 응용프로그램을 일부만 구현할 수 있다. 5~9 프로그램 방법론과 구축 계획에 따라 자원관리를 위하여 네트워크 연관 응용프로그램을 일부 조금 구현할 수 있다. 0~4 프로그램 방법론과 구축 계획에 따라 자원관리를 위하여 네트워크 연관 응용프로그램을 구현할 수 없다. 평가 메인으로