**자료구조(박요한 교수님)**

**전화번호 관리 프로그램\_Version2 입출력**

컴퓨터공학과 유상현

**1.** **프로그램 소스코드**

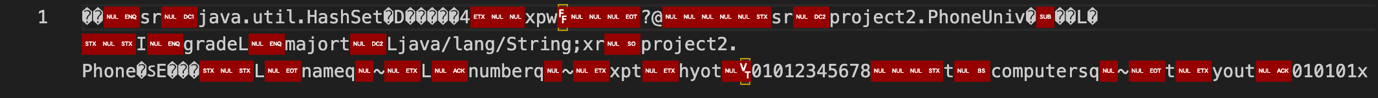
App.java

|  |
| --- |
| package project2;  import java.util.\*;  import java.io.\*;  class Phone implements Serializable{  // members  private String name, number;  // Constructor  public Phone(String name, String number) {  this.name = name;  this.number = number;  }  // methods  public void PrintInfo() {  System.out.printf("[이름: %s] [전화번호: %s]\n", this.name, this.number);  }  public String GetName() {  return this.name;  }  public String GetNumber() {  return this.number;  }  @Override  public boolean equals(Object o){  if (this == o)  return true;  if (!(o instanceof Phone))  return false;  Phone phone = (Phone) o;  if (number == phone.GetNumber() && Objects.equals(number, phone.GetNumber())) {  System.out.printf("ID %d is exist!!\n", number);  return true;  } else  return false;  }  @Override  public int hashCode() {  return Objects.hash(number);  }  }  class PhoneUniv extends Phone {  // members  private String major;  private int grade;  // constructor  public PhoneUniv(String name, String number, String major, int grade) {  super(name, number);  this.major = major;  this.grade = grade;  }  // methods  public void PrintInfo() {  System.out.printf("[이름: %s] [전화번호: %s] [전공: %s] [학년: %d]\n", GetName(), GetNumber(), this.major, this.grade);  }  }  class PhoneComp extends Phone {  // members  private String company;  // constructor  public PhoneComp(String name, String number, String company) {  super(name, number);  this.company = company;  }  // methods  public void PrintInfo() {  System.out.printf("[이름: %s] [전화번호: %s] [회사: %s]\n", GetName(), GetNumber(), this.company);  }  }  class PhoneManager {  // members  private HashSet<Phone> phonelist;  private String filePath = "/Users/sanghyun/Develop/java/homework/homework1/src/project2/object.txt";  Scanner scanner = new Scanner(System.in);  // constructor  PhoneManager(){  phonelist = new HashSet<Phone>();  }  // method  public void InsertData(){  System.out.printf("어떤 사용자인가요?[1 - 일반 / 2 - 대학 / 3 - 회사] : ");  int choice, grade;  String name, company, major, number;  choice = scanner.nextInt();  switch(choice){  case 1:  System.out.println("[정보를 입력하세요]");  System.out.printf("이름 : ");  name = scanner.next();  System.out.printf("전화번호 : ");  number = scanner.next();  phonelist.add(new Phone(name, number));  break;  case 2:  System.out.println("[정보를 입력하세요]");  System.out.printf("이름 : ");  name = scanner.next();  System.out.printf("전화번호 : ");  number = scanner.next();  System.out.printf("전공 : ");  major = scanner.next();  System.out.printf("학년 : ");  grade = scanner.nextInt();  phonelist.add(new PhoneUniv(name, number, major, grade));  break;  case 3:  System.out.println("[정보를 입력하세요]");  System.out.printf("이름 : ");  name = scanner.next();  System.out.printf("전화번호 : ");  number = scanner.next();  System.out.printf("회사명 : ");  company = scanner.next();  phonelist.add(new PhoneComp(name, number, company));  break;  }  }  public void SearchData(){  System.out.printf("찾을 사람의 이름: ");  String name = scanner.next();  Iterator<Phone> iter = phonelist.iterator();  while(iter.hasNext()){  Phone tmp = iter.next();  if(name.equals(tmp.GetName())){  System.out.println("[" + name + "의 정보");  tmp.PrintInfo();  return;  }  }  System.out.println("!!찾는 사람이 없습니다!!");  }  public void DeleteData(){  System.out.printf("삭제할 사람 이름: ");  String name = scanner.next();  Iterator<Phone> iter = phonelist.iterator();  while(iter.hasNext()){  Phone tmp = iter.next();  if(name.equals(tmp.GetName())){  iter.remove();  System.out.println("[" + name + "의 정보를 삭제했습니다.]");  return;  }  }  System.out.println("!!삭제할 사람이 없습니다!!");  }  public void PrintData(){  if(phonelist.isEmpty()){  System.out.printf("[출력할 데이터가 없습니다]\n");  return ;  }  Iterator<Phone> iter = phonelist.iterator();  System.out.println("전체 데이터 출력");  while(iter.hasNext()){  Phone tmp = iter.next();  tmp.PrintInfo();  }  }  public void writeClass() throws IOException, ClassNotFoundException {  ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(filePath));  oos.writeObject(phonelist);  oos.close();  }  public void readClass() throws IOException, ClassNotFoundException {  ObjectInputStream ois = null;  try {  ois = new ObjectInputStream(new FileInputStream(filePath));  Object obj = ois.readObject();  if (obj instanceof HashSet) {  phonelist = (HashSet<Phone>) obj;  } else {  System.out.println("Wrong type Object");  }  ois.close();  } catch (EOFException e) {  System.out.println("가져올 정보가 없습니다");  } finally {  if (ois != null) {  ois.close();  }  }  }  }  public class App {  static void PrintMenu() {  System.out.println("=================");  System.out.println("1. 데이터 입력");  System.out.println("2. 데이터 검색");  System.out.println("3. 데이터 삭제");  System.out.println("4. 전체 데이터 출력");  System.out.println("5. 프로그램 종료");  System.out.println("=================");  System.out.print("메뉴를 선택하세요: ");  }  public static void main(String[] args) throws Exception {  Scanner scanner = new Scanner(System.in);  PhoneManager pm = new PhoneManager();  boolean roof = true;  pm.readClass();  while (roof) {  PrintMenu();  int choice = scanner.nextInt();  switch (choice) {  case 1: // 데이터 입력 구현  pm.InsertData();  break;  case 2: // 데이터 검색 구현  pm.SearchData();  break;  case 3: // 데이터 삭제 구현  pm.DeleteData();  break;  case 4: // 전체 데이터 출력 구현  pm.PrintData();  break;  case 5:  System.out.println("프로그램을 종료합니다.");  pm.writeClass();  roof = false;  break;  default:  System.out.println("올바른 메뉴를 선택하세요.");  }  }  scanner.close();  }  } |

**2. 실행 화면**

텍스트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명



**3. 느낀 점(고찰)**

저번과제에서 추가적으로 HashSet을 사용하여 자료를 관리하니 더 편리하게 자료구조를 구현할 수 있었고 객체로 바이트 스트림을 저장하다 보니 프로그램을 종료하더라도 다시 실행했을 때 이전에 실행했던 데이터가 남아 있는 것이 신기하였고 이를 이용해 더 재밌게 이용할 수 있을 것 같습니다.