|  |  |  |  |
| --- | --- | --- | --- |
| 과목명 | Advanced Java Programming | 시행일자 | 2022.03.31 (수행시간 2시간) |
| 점수 |  | 제출자 |  |
| 아래 요구 사항에 맞는 제품을 제작하여 제출하시오   |  | | --- | | [요구사항]  성적관리 프로그램을 제작 의뢰합니다  각과목은 기초문법, 백엔드, 프런트엔드로 3과목입니다  학생수는 20~30내외 입니다  학생의 정보는 학번과 이름, 각 과목의 성적, 평균점수(소수점 두자리)를 출력해주세요  수정과 삭제가 가능하도록 기능해야 합니다  사용자가 편리한 인터페이스 지원을 요구합니다  단, 콘솔 실행입니다  영구저장은 되지 않습니다  학번은 최초 부여받고 변동 없습니다 학번서식은 1번부터 순서대로 부여됩니다 |  |  |  |  |  | | --- | --- | --- | --- | | 배점표 | | | | | 1 | 일정시한 엄수 | 20 |  | | 2 | 사용자 편의성(UI) | 10 |  | | 3 | 자료구조의 선택 | 10 |  | | 4 | 코드 가독성 | 5 |  | | 5 | 리펙토링 | 5 |  | | 6 | 자원의 효율성 | 5 |  | | 7 | 확장성 | 5 |  | | 8 | 제약조건 | 5 |  | | 9 | 문법 활용 능력 | 5 |  | | 10 | 문서화 | 5 |  | | 11 | 제출자 신상정보 | 10 |  | | 12 | 과제 | 10 |  | | 13 | 기타 | 5 |  | | 계 | | 100 |  | | | | |
| 제품어필 | | | |
|  | | | |
| 출력결과 | | | |
| <스크린샷> | | | |
| 소소코드 | | | |
| <첨부>  package com.bit.day12;  import java.io.\*;  import java.lang.reflect.Array;  import java.util.\*;  import javax.security.auth.Subject;  // 성적관리 프로그램을 제작 의뢰합니다  // 각 과목은 기초문법,백엔드,프런트엔드로 3과목입니다.  // 학생수는 20~30내외 입니다.  // 학생의 정보는 학번과 이름, 각 과목의 성적, 평균점수를 출력해주세요  // 수정과 삭제가 가능하도록 기능해야 합니다.  // 사용자가 편리한 인터페이스 지원을 요구합니다  // 단 콘솔 실행 입니다  // 영구저장은 되지 않습니다.  // 학번은 자동증가, 학번은 최초 부여받고 변동 없습니다.  // 학번 서식은 1번부터 순서대로  // 평균점수 소수점 둘째자리까지  // 제품어필  //  // 출력결과  // 스크린샷  // 소스코드  // 첨부  class Student {  // 이름  String name;  // 학번  int stunum = 1;  // 과목성적  int basic;  int backend;  int frontend;  public String getName() {  return name;  }  public void setName(String name) {  this.name = name;  }  public int getStunum() {  return stunum;  }  public void setStunum(int stunum) {  this.stunum = stunum;  }  public int getBasic() {  return basic;  }  public void setBasic(int basic) {  this.basic = basic;  }  public int getBackend() {  return backend;  }  public void setBackend(int backend) {  this.backend = backend;  }  public int getFrontend() {  return frontend;  }  public void setFrontend(int frontend) {  this.frontend = frontend;  }  }  public class Test {  static final int STUDENT\_MIN\_NUM = 20;  static final int STUDENT\_MAX\_NUM = 30;  public static void main(String[] args) {  BufferedReader br = new BufferedReader(new InputStreamReader(System.in));  String[] subject = { "기초문법", "백엔드", "프런트엔드" };  ArrayList<String> stname = new ArrayList<String>();  ArrayList<Integer> stnum = new ArrayList<Integer>();  ArrayList<Integer> basic = new ArrayList<Integer>();  ArrayList<Integer> backend = new ArrayList<Integer>();  ArrayList<Integer> frontend = new ArrayList<Integer>();  Student stu = new Student();  int menu = 0;  int totalstu = 0;  int cnt = 1;  while (true) {  System.out.println("성적관리 프로그램 (ver 0.1.0)");  System.out.print("1. 입력 2. 출력 3. 수정 4. 삭제 0. 종료 >> ");  try {  menu = Integer.parseInt(br.readLine());  } catch (NumberFormatException e) {  System.out.println("번호를 정확히 입력해주세요");  continue;  } catch (IOException e) {  System.out.println("번호를 정확히 입력해주세요");  continue;  }  if (menu == 1) {  System.out.print("학생 인원> ");  try {  totalstu = Integer.parseInt(br.readLine());  } catch (NumberFormatException e1) {  // **TODO** Auto-generated catch block  e1.printStackTrace();  } catch (IOException e1) {  // **TODO** Auto-generated catch block  e1.printStackTrace();  }  if (totalstu < 0) {  System.out.println("최소 인원은 20명입니다");  continue;  } else if (totalstu > 30) {  System.out.println("최대 인원은 30명입니다");  } else {  for (int i = 0; i < totalstu; i++) {  System.out.print("이름> ");  try {  stu.setName(br.readLine());  } catch (IOException e) {  System.out.println("이름을 제대로 입력해 주세요");  continue;  }  // 이름하고 학번 추가  stname.add(stu.name);  stnum.add(stu.stunum);  // 학번 증가  stu.stunum++;  // 성적 입력  try {  // 기초문법  System.out.print(subject[0] + "> ");  stu.setBasic(Integer.parseInt(br.readLine()));  // 백엔드  System.out.print(subject[1] + "> ");  stu.setBackend(Integer.parseInt(br.readLine()));  // 프론트엔드  System.out.print(subject[2] + "> ");  stu.setFrontend(Integer.parseInt(br.readLine()));  } catch (NumberFormatException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  } catch (IOException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  }  // 점수 추가  basic.add(stu.basic);  backend.add(stu.backend);  frontend.add(stu.frontend);  }  cnt++;  }  } else if (menu == 2) {  System.out.println("성적 출력");  for (int i = 0; i < stname.size(); i++) {  System.out.print("이름: " + stname.get(i) + "\n");  System.out.print("학번: " + stnum.get(i) + "\n");  System.out.print("성적: " + "기초문법> " + basic.get(i) + " 백엔드> " + backend.get(i) + " 프론트엔드> "  + frontend.get(i) + "\n\n");  }  } else if (menu == 3) {  // System.out.print("수정할 학번을 입력하세요> ");  // try {  // int input = Integer.parseInt(br.readLine());  //  // if (input <= stnum.size()) {  // for (int i = 0; i < stnum.size(); i++) {  // stname.  // }  // stnum.set(input - 1, null);  // }  // } catch (NumberFormatException e) {  // // **TODO** Auto-generated catch block  // e.printStackTrace();  // } catch (IOException e) {  // // **TODO** Auto-generated catch block  // e.printStackTrace();  // }  } else if (menu == 4) {  System.out.print("삭제할 학번> ");  int input;  try {  input = Integer.parseInt(br.readLine());  stname.remove(input - 1);  stnum.remove(input - 1);  basic.remove(input - 1);  backend.remove(input - 1);  frontend.remove(input - 1);  stu.stunum--;  } catch (NumberFormatException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  } catch (IOException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  }  } else if (menu == 0) {  System.out.println("프로그램을 종료하겠습니다");  break;  } else {  System.out.println("번호를 다시 입력해 주세요");  continue;  }  }  }  } | | | |