|  |  |  |
| --- | --- | --- |
| **2019\_2\_java2\_12** | **학번 : 20135151** | **이름 : 이갑성** |

* **내용 점검**

1. Map 인터페이스

|  |
| --- |
| import java.util.\*;  class Student {  private int number;  private String name;  public Student(int number, String name) {  this.number = number;  this.name = name;  }  public String toString() {  return name;  }  }  **public** **class** MapTest {  **public** **static** **void** main(String[] args) {  Map<String, Student> st = **new** HashMap<String, Student>();  st.put("20090001", **new** Student(20090001, "구준표"));  st.put("20090002", **new** Student(20090002, "금잔디"));  st.put("20090003", **new** Student(20090003, "윤지후"));  System.***out***.println(st); // 모든 항목을 출력한다.  st.remove("20090002"); // 하나의 항목을 삭제한다.  st.put("20090003", **new** Student(20090003, "소이정")); // 하나의 항목을 대치한다.  System.***out***.println(">>>>>>> 키값에 대응하는 value 출력");  System.***out***.println(st.get("20090003")); // 키 값에 해당하는 value 반환  System.***out***.println("\n >>>>>> 1: 람다식을 이용한 원소 출력 ");  st.forEach((key, value) -> {  System.***out***.println("key = " + key + ", value = " + value);  });  System.***out***.println("\n >>>>>> 2: Map.Entry 인터페이스를 이용한 원소 출력 ");  **for** (Map.Entry<String, Student> s : st.entrySet()) { // 모든 항목을 방문한다  String key = s.getKey();  Student value = s.getValue();  System.***out***.println("key=" + key + ", value=" + value);  }  }  } |
| **[실행 결과]** |

1. Collection 클래스 - 정렬

|  |
| --- |
| import java.util.\*;  class Student implements Comparable<Student>{  int number; String name;  public Student(int number, String name) {  this.number = number; this.name = name; }  public String toString() {  return "학번 : " + number + "\t이름 : " + name + "\n"; }  public int compareTo(Student s ){ //재정의  return this.number - s.number;  }  }  public class SetTest {  public static void main(String args[]) {  Student array[]={new Student(2009002, "김철수"), new Student(2009001, "이철수"),  new Student(2009003, "박철수")};  List<Student> list=Arrays.asList(array); //배열을 리스트로 변환  Collections.sort(list); //크기순으로 정렬  System.out.println(list);  }  } |
| **[실행결과]** |

1. 예외처리

|  |
| --- |
| **class** Account {  **int** balance; // 잔액  @Override  **public** String toString() {  **return** "Account [balance=" + balance + "]";  }  **int** withdraw(**int** amount) **throws** Exception {  **if** (balance < amount)  // 현재 메소드에서 발생하는 예외 종류를 표시하는 throws 절  **throw** **new** Exception("잔액이 부족합니다.");  balance -= amount;  **return** amount;  }  **int** deposit(**int** amount) **throws** Exception{  **if**(amount < 0)  **throw** **new** Exception("입금액이 음수 입니다.");  balance += amount;  **return** amount;  }  }  **public** **class** ExceptionExample {  **public** **static** **void** error(String str) {  System.***out***.println(str + "예외를 정상 처리 하였습니다 ");  }  **public** **static** **void** main(String args[]) {  Account obj = **new** Account();  **try** {  **int** amount = obj.withdraw(100000000); // 예외 발생 메소드  System.***out***.println("인출액:" + amount);  } **catch** (Exception e) { // 발생된 예외 처리  String msg = e.getMessage(); // 오류 메시지 출력  System.***out***.println(msg);  *error*("인출 ");  }  System.***out***.println(obj);  **try** {  **int** amount = obj.deposit(-100); // 예외 발생 메소드  System.***out***.println("입금액:" + amount);  } **catch** (Exception e) { // 발생된 예외 처리  String msg = e.getMessage(); // 오류 메시지 출력  System.***out***.println(msg);  *error*("입금");  }  System.***out***.println(obj);  }  } |
| **[실행결과]** |

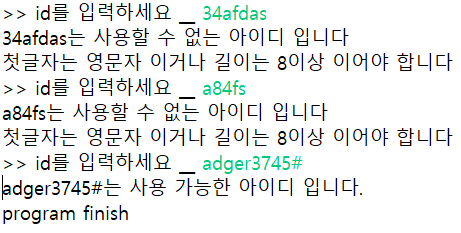
* **실습 과제**

1. HashMap을 사용하여 주사위를 100번 던졌을 때 주사위 눈의 빈도수를 계산하는 프로그램을 작성하시오

|  |  |
| --- | --- |
|  | **public** **class** Answer1 {  **public** **static** **void** main(String args[]) {  Map<Integer, Integer> map;  map=Count.*count*(); //빈도수를 계산한 HashMap을 반환받아 저장)  System.***out***.println("Map에 저장된 빈도수를 출력합니다");  Count.*write*(map); //HashMap에 저장된 원소 출력  }  } |

|  |
| --- |
| **[프로그램 소스]**  import java.util.\*;  public class Count {  public static Map<Integer, Integer> count() {  Random rnd = new Random();  int rndEye;  HashMap<Integer, Integer> hm = new HashMap<>();  for (int i = 0; i < 100; i++) {  rndEye = rnd.nextInt(6) + 1;  Integer cnt = hm.get(rndEye); //랜덤으로 나온 눈의 값(빈도수)를 cnt에 넣어준다.  hm.put(rndEye, hm.get(rndEye) == null ? 1 : cnt+1);  //나온 눈의 값을 key에 넣어주고 그에 해당된 빈도수를 증가시켜준다.  }  return hm;  }  public static void write(Map<Integer, Integer> m) {  m.forEach((key, value) ->  {System.***out***.println("주사위 눈 = " + key + "빈도수 = " + value);}  );  }  }  import java.util.\*;  public class Answer1 {  public static void main(String args[]) {  Map<Integer, Integer> map;  map = Count.*count*(); // 빈도수를 계산한 HashMap을 반환받아 저장)  System.***out***.println("Map에 저장된 빈도수를 출력합니다");  Count.*write*(map); // HashMap에 저장된 원소 출력  }  } |
| **[실행결과]** |

1. 키보드로 입력 받은 문자열에 대하여 길이가 8미만이거나 첫 글자가 영문자가 아니면 예외를 발생시켜 “사용할 수 없는 아이디”를 출력하고, 길이가 8이상이고 첫 글자가 영문자 이면 “사용 가능한 아이디 입니다.” 를 출력하는 프로그램을 클래스를 최소 2개 사용하여 작성하세요.



|  |
| --- |
| **[프로그램 소스]**  import java.util.\*;  public class createID {  String id;    public static void createID(String id) throws Exception{  if(id.length() < 8 || ((id.charAt(0) - '0') >=0 && (id.charAt(0) - '0') <= 9)) {  throw new Exception(id + "는 사용할 수 없는 아이디 입니다.");  }  System.***out***.println(id + "는 사용 가능한 아이디 입니다.");  }  }  import java.util.Scanner;  public class test {  public static void main(String[] args) {  Scanner sc = new Scanner(System.***in***);  String id;  for(int i = 0; i < 3; i++) {  System.***out***.print(">> id를 입력하세요\_\_ ");  id = sc.next();  try {  createID.*createID*(id);  }  catch(Exception e) {  String msg = e.getMessage();  System.***out***.println(msg);  System.***out***.println("첫글자는 영문자 이거나 길이는 8이상 이어야 합니다.");  }  }  System.***out***.println("program finish");  }  } |
| **[실행결과]** |

1. 간단한 주소록을 처리하는 프로그램을 HashMap을 이용하여 작성하시오

- <id, Person>의 key와 value 저장, id는 String

- 임의의 데이터를 저장

- id를 입력하면 해당 Person 데이터 출력.

- HashMap에 저장된 모든 데이터 출력. 단, Map.Entry 인터페이스를 이용한 원소 출력

**class** Person{

**private String** name; //이름

**private String** num; //전화번호

**public** Person(String name, **String** num){

**this**.name=name;

**this**.num=num;

}

**public** String toString(){

**return** " [이름 : " + name + ", 전화번호 : " + num +"]" ;

}

}

|  |
| --- |
| **[프로그램 소스]**  import java.util.\*;  public class test {  public static void main(String[] args) {  Scanner sc = new Scanner(System.***in***);  HashMap<String, Person> hm = new HashMap<>();  String id;    hm.put("sonsational", new Person("손흥민", "010-1259-7752"));  hm.put("goalblocker", new Person("김승규", "010-6348-4523"));  hm.put("youngboy", new Person("이강인", "010-2684-3654"));  hm.put("bullstriker", new Person("황희찬", "010-8552-7486"));    do {  System.***out***.print("찾고자하는 아이디를 입력하세요: ");  id = sc.next();  if(id.equals("q")) {  System.***out***.println("검색 프로그램을 종료합니다");  break;  }  if(hm.get(id) == null)  System.***out***.println("찾고자 하는 아이디에 대한 정보가 없습니다.");  else {  System.***out***.println(hm.get(id));  }    }while(true);    System.***out***.println("=========== 모든 데이터를 출력합니다. ===========");    for(Map.Entry<String, Person> result : hm.entrySet()) {  System.***out***.println(result.getValue());  }    }  } |
| **[실행 결과]** |

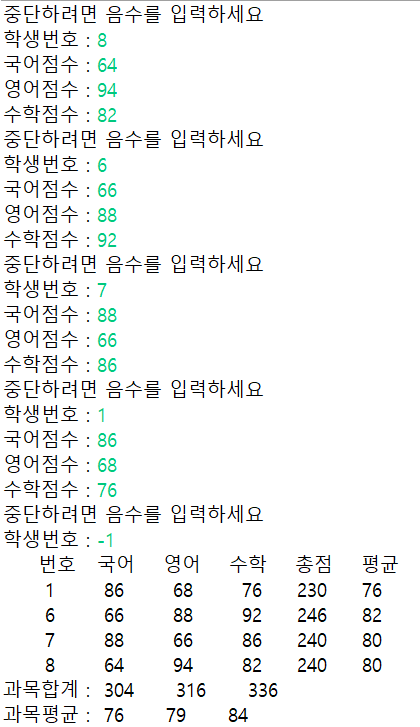
1. 한 학급 학생들의 학생 번호와 국어, 영어, 수학 시험 점수를 입력 받아 학생 번호순으로 정렬하고 학생 개인의 총점과 평균 및 과목별 총점과 평균을 구하여 입력 자료와 함께 출력하는 성적처리 프로그램을 작성하시오.

* 조건 1: ArrayList를 사용할 것

조건 2: 번호 순으로 정렬할 때 Collection 클래스의 sort() 메소드를 사용할 것

(comparable 재정의 해야한다.)

* 조건 3: 2개 이상의 클래스로 나누어 작성할 것



|  |
| --- |
| **[프로그램 소스]**  public class Student implements Comparable<Student>{  int num;  int korean;  int english;  int math;    public Student(int num, int korean, int english, int math) {  this.num = num;  this.korean = korean;  this.english = english;  this.math = math;  }    public int getKorean() {  return korean;  }  public int getEnglish() {  return english;  }  public int getMath() {  return math;  }  *@Override*  public int compareTo(Student s) {  return this.num - s.num;  }    public String toString() {  String result;  int total = korean + english + math;  int avg = total / 3;    result = num + "\t" + korean + "\t" + english + "\t" + math  + "\t" + total + "\t" + avg;    return result;  }    }  import java.util.\*;  public class Test {  public static void main(String[] args) {  Scanner sc = new Scanner(System.***in***);  int num, k, e, m, kTotal = 0, eTotal = 0, mTotal = 0;  ArrayList<Student> st = new ArrayList<>();    do{  System.***out***.println("중단하려면 음수를 입력하세요.");  System.***out***.print("학생번호: ");  num = sc.nextInt();  if(num < 0) {  break;  }  System.***out***.print("국어점수: ");  k = sc.nextInt();  System.***out***.print("영어점수: ");  e = sc.nextInt();  System.***out***.print("수학점수: ");  m = sc.nextInt();    st.add(new Student(num, k, e, m));    }while(true);    Collections.*sort*(st);  System.***out***.println("번호\t국어\t영어\t수학\t총점\t평균");  for(Student result : st) {  System.***out***.println(result);  kTotal += result.getKorean();  eTotal += result.getEnglish();  mTotal += result.getMath();  }    System.***out***.println("과목합계 : " + kTotal + "\t" + eTotal + "\t" + mTotal);  System.***out***.println("과목평균 : " + kTotal/st.size() + "\t" + eTotal/st.size() + "\t" + mTotal/st.size());  }    } |
| **[실행 결과]** |

1. Book 클래스 이용하여 Book 정보를 ArrayList에 저장하고, 저장된 모든 데이터를 출력하는 프로그램을 작성하시오. 다음과 같은 메뉴를 제공하도록 한다.

1: 저장 2:대출 3:검색 4:리스트 출력 5. 삭제 6. 종료

public class Book {

private String number; //관리번호

private String title; //제목

private String author; //저자

private boolean status; // 대출 상태(대출 중-> true, 대출가능 -> false)

//생성자 – 본인작성

- 형식매개변수로 필드 초기화, 필드 status는 false로 할 것

// toString() 메소드 – 본인작성

- 객체 상태를 문자열로 반환(관리번호, 제목, 저자, 대출상태

//void rentBook() 메소드 – 본인작성

- status 필드값이 false이면 true로 변경한 후 “대출완료” 출력

- ture 이면 “대출 불가(대출 중)” 출력

//모든 필드에 대한 접근자 메소드 – 본인작성

}

import java.util.\*;

public BookTest{

public static void main(String[] args){

//메뉴를 선택

//1번 : ArrayList에 저장 – 필요한 자료는 입력을 받는다

//2번 : 대출하고자 하는 책 제목을 입력 받은 후 rentBook()메소드로 처리

//3번 : 검색하고자 하는 책 제목을 입력 받은 후 객체 상태 출력

//4번 : ArrayList에 저장 된 모든 원소 출력

//5번 : 입력된 관리번호와 일치하는 책을 ArrayList 에서 삭제

//6번 : 반복문 종료

}

}

|  |
| --- |
| **[프로그램 소스]**  public class Book {  private String number; // 관리번호  private String title; // 제목  private String author; // 저자  private boolean status; // 대출 상태(대출 중-> true, 대출가능 -> false)  public Book(String n, String t, String a) {// 생성자 - 본인작성, 형식매개변수로 필드 초기화, 필드 status는 false로 할 것  this.number = n;  this.title = t;  this.author = a;  this.status = false;  }  // toString() 메소드 – 본인작성, 객체 상태를 문자열로 반환(관리번호, 제목, 저자, 대출상태)  public String toString() {  String result;  String rent;  if(status) {  rent = "대출 중";  }  else {  rent = "대출 가능";  }  result = "관리번호: " + number + " | 제목: " + title + " | 저자: " + author + " | 대출상태:" + rent;  return result;  }  // void rentBook() 메소드 – 본인작성  // status 필드값이 false이면 true로 변경한 후 “대출완료” 출력  // ture 이면 “대출 불가(대출 중)” 출력  public void rentBook() {  if(status) {  System.***out***.println("대출 불가(대출 중)");  }  else {  status = true;  System.***out***.println("대출완료");  }  }    // 모든 필드에 대한 접근자 메소드 – 본인작성  public String getNumber() {  return number;  }  public String getTitle() {  return title;  }  public String getAuthor() {  return author;  }  public boolean isStatus() {  return status;  }  }  import java.util.\*;  public class BookTest {  public static void main(String[] args) {  Scanner sc = new Scanner(System.***in***);  boolean flag = true, find = false;  int menu, index;  String fbTitle, fn;  ArrayList<Book> list = new ArrayList<>();    System.***out***.println("도서관 시스템에 접속을 환영합니다.");  do {  System.***out***.println("이용하고자 하는 서비스를 입력하세요.");  System.***out***.print("1.저장 2.대출 3.검색 4.리스트 출력 5.삭제 6.종료 >>> ");  menu = sc.nextInt();  switch(menu) {  case 1: // 1번 : ArrayList에 저장 – 필요한 자료는 입력을 받는다  System.***out***.println("등록하고자 하는 책의 관리번호, 제목, 저자 순으로 입력해주세요.");  list.add(new Book(sc.next(), sc.next(), sc.next()));  break;  case 2: // 2번 : 대출하고자 하는 책 제목을 입력 받은 후 rentBook()메소드로 처리  System.***out***.print("대출하고자 하는 책 제목을 입력하세요: ");  fbTitle = sc.next();  for(Book b : list) {  if(b.getTitle().equals(fbTitle)) {  b.rentBook();  find = true;  break;  }  }  if(find == false) {  System.***out***.println("해당 책은 저희 도서관에 존재하지 않습니다.");  }  break;  case 3: // 3번 : 검색하고자 하는 책 제목을 입력 받은 후 객체 상태 출력  System.***out***.print("대출하고자 하는 책 제목을 입력하세요: ");  fbTitle = sc.next();  for(Book b : list) {  if(b.getTitle().equals(fbTitle)) {  System.***out***.println(b);  find = true;  break;  }  }  if(find == false) {  System.***out***.println("해당 책은 저희 도서관에 존재하지 않습니다.");  }  break;  case 4: // 4번 : ArrayList에 저장 된 모든 원소 출력  System.***out***.println("============== 도서관의 모든 책 정보 출력 ==============");  for(Book b : list) {  System.***out***.println(b);  }  break;  case 5: // 5번 : 입력된 관리번호와 일치하는 책을 ArrayList 에서 삭제  System.***out***.print("삭제하고자 하는 책의 관리번호를 입력하세요: ");  fn = sc.next();  for(Book b : list) {  if(b.getNumber().equals(fn)) {  list.remove(b);  flag = true;  break;  }  }  if(find == false) {  System.***out***.println("해당 책은 저희 도서관에 존재하지 않습니다.");  }  break;  case 6: // 6번 : 반복문 종료  flag = false;  break;  }  }while(flag);  System.***out***.println("프로그램을 종료합니다.");  }  } |
| **[실행결과]** |