Java

겨울방학

과제

10105 김성훈

대덕소프트웨어마이스터고등학교

Chapter 2 problem 7

package com.SHGroup.homework.chapter2.problem7;

import com.SHGroup.homework.HomeworkClass;

import java.util.Scanner;

public class Homework\_2\_7 extends HomeworkClass {

*/\**

*7. 2차원 평면에서 하나의 직사각형은 두 점으로 표현된다.*

*(50, 50)과 (100, 100)의 두 점으로 이루어진 사각형이 있다고 하자.*

*이때 키보드로부터 한 점을 구성하는 정수 x와 y 값을 입력받고*

*점(x, y)가 이 직사각형 안에 있는지를 판별하는 프로그램을 작성하라.*

*\*/*

public final void run() {

int x, y;

Scanner sc = new Scanner(System.in);

System.out.print("점의 X 값을 입력하세요 : ");

x = sc.nextInt();

System.out.print("점의 Y 값을 입력하세요 : ");

y = sc.nextInt();

sc.close();

if((x <= 100 && x >= 50)

&& (y <= 100 && y >= 50)){

System.out.println("해당 점은 그 사각형 안에 있습니다.");

}else{

System.out.println("해당 점은 그 사각형 안에 없습니다.");

}

}

}

Chapter 3 problem 5

package com.SHGroup.homework.chapter3.problem5;

import com.SHGroup.homework.HomeworkClass;

import java.util.Scanner;

public class Homework\_3\_5 extends HomeworkClass {

*/\**

*키보드로부터 정수를 10개 입력받아 배열에 저장하고 이 정수 중에서 3의 배수인 수만 골라 출력하는 프로그램을 작성하라.*

*\*/*

private static final int ARR\_MAX = 10;

public final void run() {

final int[] arr = new int[ARR\_MAX];

int stack\_ptr = 0;

final Scanner sc = new Scanner(System.in);

for(int i = 0 ; i < ARR\_MAX ; i ++){

System.out.print(Integer.toString(i + 1) + "번재 숫자를 입력하세요 : ");

int tmp = sc.nextInt();

if(tmp % 3 == 0){

arr[stack\_ptr++] = tmp;

}

}

if(stack\_ptr == 0){

System.out.println("입력한 수 중에서 3의 배수가 없습니다.");

}else{

System.out.print("입력한 수 중에서 3의 배수는 ");

for(int i = 0 ; i < stack\_ptr ; i ++){

System.out.print((i != 0 ? ", " : "") + Integer.toString(arr[i]));

}

System.out.print("입니다.");

}

sc.close();

}

}

Chapter 3 problem 7

package com.SHGroup.homework.chapter3.problem7;

import com.SHGroup.homework.HomeworkClass;

import java.util.Random;

import java.util.Scanner;

public class Homework\_3\_7 extends HomeworkClass {

*/\**

*4×4의 2차원 배열을 만들고 이곳에 1에서 10까지 범위의 정수를 랜덤하게 생성하여 정수 16개를 배열에 저장하고, 2차원 배열을 화면에 출력하라.*

*\*/*

private static final int ARR\_MAX = 4;

public final void run() {

final Random r = new Random();

final int[][] arr = new int[ARR\_MAX][ARR\_MAX];

for(int i = 0 ; i < ARR\_MAX ; i ++){

for(int j = 0 ; j < ARR\_MAX ; j ++){

arr[i][j] = r.nextInt() % 10 + 1;

System.out.print((j == 0 ? "" : " ") + "[" + arr[i][j] + "]");

}

System.out.println();

}

}

}

Chapter 4 problem 1

package com.SHGroup.homework.chapter4.problem1;

import com.SHGroup.homework.HomeworkClass;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.Vector;

public class Homework\_4\_1 extends HomeworkClass {

*/\**

*1. 노래를 나타내는 Song이라는 클래스를 설계하라. Song 클래스는 다음과 같은 필드를 갖는다.*

*- 노래의 제목을 나타내는 title*

*- 가수를 나타내는 artist*

*- 노래가 속한 앨범 제목을 나타내는 album*

*- 노래의 작곡가를 나타내는 composer, 작곡가는 여러 명 있을 수 있다.*

*- 노래가 발표된 연도를 나타내는 year*

*- 노래가 속한 앨범에서의 트랙 번호를 나타내는 track*

*생성자는 기본 생성자와 모든 필드를 초기화하는 생성자를 작성하고, 노래의 정보를 출력하는 show() 메소드도 작성하라.*

*ABBA의 "Dancing Queen" 노래를 Song 객체로 생성하고 show()를 이용하여 이 노래의 정보를 출력하는 프로그램을 작성하라.*

*\*/*

private static final SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");

public final void run() {

Date date = null;

try{

date = dateFormat.parse("1976-08-15");

}catch(Exception *ex*){

ex.printStackTrace();

}

Song dancingQueen = new Song("Dancing Queen", "ABBA", "Arrival", 1, date, "Benny Andersson", "Bjorn Ulvaeus");

dancingQueen.show();

}

public class Song {

private final String title;

private final String artist;

private final String album;

private final int track;

private final Vector<String> composer = new Vector<>();

private final Date create\_year;

public Song(){

this.title = "제목 없음";

this.artist = "가수 없음";

this.album = "앨범 없음";

this.track = 0;

Date create\_year = null;

try{

create\_year = dateFormat.parse("0000-00-00");

}catch(Exception *ex*){

ex.printStackTrace();

create\_year = null;

}

this.create\_year = create\_year;

}

public Song(final String *title*, final String *artist*, final String *album*, final int *track*, final Date *create\_year*, final String... *composers*){

this.title = title;

this.artist = artist;

this.album = album;

this.track = track;

this.create\_year = create\_year;

if(composers != null){

for(String n : composers){

if(n == null)

continue;

composer.add(n);

}

}

}

public final void show(){

Vector<String> info = new Vector<>();

info.add("=================");

info.add("Track info");

info.add(" Title : " + title);

info.add(" Artist : " + artist);

if(composer.size() > 0){

info.add(" Composer");

for(String n : composer){

info.add(" - " + n);

}

}

info.add(" Album : " + album);

info.add(" Track : " + Integer.toString(track));

info.add(" Create Year : " + dateFormat.format(create\_year));

info.add("=================");

for(String n : info)

System.out.println(n);

}

}

}

Chapter 4 problem 2

package com.SHGroup.homework.chapter4.problem2;

import com.SHGroup.homework.HomeworkClass;

import java.util.Vector;

public class Homework\_4\_2 extends HomeworkClass {

*/\**

*다음과 같은 멤버를 가지는 직사각형을 표현하는 Rectangle 클래스를 작성하라.*

*- int 타입의 x1, y1, x2, y2 필드 : 사각형을 구성하는 두 점의 좌표*

*- 생성자 2개 : 매개 변수 없는 생성자와 x1, y1, x2, y2의 값을 설정하는 생성자*

*- void set(int x1, int y1, int x2, int y2) : x1, y1, x2, y2 좌표 설정*

*- int square() : 사각형 넓이 리턴*

*- void show() : 좌표와 넓이 등 직사각형 정보의 화면 출력*

*- boolean equals(Rectangle r) : 인자로 전달된 객체 r과 현 객체가 동일한 직사각형이면 true 리턴*

*Rectangle을 이용한 main() 메소드는 다음과 같으며 이 main() 메소드가 잘 작동하도록 하라.*

*\*/*

public final void run() {

Rectangle r = new Rectangle(0, 10, 10, 0);

r.show();

Rectangle r2 = new Rectangle(50, 10, 10, 50);

Rectangle r3 = new Rectangle(0, 10, 10, 0);

System.out.println(r.equals(r2) ? "r == r2" : "r != r2");

System.out.println(r.equals(r3) ? "r == r3" : "r != r3");

}

public class Rectangle {

private int x1;

private int y1;

private int x2;

private int y2;

public Rectangle(){

this.x1 = 0;

this.y1 = 0;

this.x2 = 0;

this.y2 = 0;

}

public Rectangle(final int *x1*, final int *y1*, final int *x2*, final int *y2*){

this.x1 = x1;

this.y1 = y1;

this.x2 = x2;

this.y2 = y2;

}

public final void set(final int *x1*, final int *y1*, final int *x2*, final int *y2*){

this.x1 = x1;

this.y1 = y1;

this.x2 = x2;

this.y2 = y2;

}

public final int square(){

return Math.abs(x1 - x2) \* Math.abs(y1 - y2);

}

public final void show(){

Vector<String> info = new Vector<>();

info.add("=================");

info.add(" x1 : " + Integer.toString(x1));

info.add(" y1 : " + Integer.toString(y1));

info.add(" x2 : " + Integer.toString(x2));

info.add(" y2 : " + Integer.toString(y2));

info.add(" square : " + Integer.toString(square()));

info.add("=================");

for(String n : info)

System.out.println(n);

}

public final boolean equals(final Rectangle *r*){

return r != null

&& r.x1 == this.x1

&& r.y1 == this.y1

&& r.x2 == this.x2

&& r.y2 == this.y2;

}

}

}

Chapter 4 problem 5

package com.SHGroup.homework.chapter4.problem5;

import com.SHGroup.homework.HomeworkClass;

import java.util.Vector;

public class Homework\_4\_5 extends HomeworkClass {

*/\**

*다수의 클래스를 정의하고 활용하는 연습을 해보자. 더하기(+), 빼기(-), 곱하기(\*), 나누기(/)를*

*수행하는 각 클래스 Add, Sub, Mul, Div를 만들어라. 이들은 모두 다음 필드와 메소드를 가진다.*

*- int 타입의 a, b 필드 : 연산하고자 하는 피연산자*

*- void setValue(int a, int b) : 피연산자를 객체 내에 설정한다.*

*- int calculate() : 해당 클래스의 목적에 맞는 연산을 실행하고 그 결과를 리턴한다.*

*main() 메소드에서는 다음 실행 사례의 그림과 같이 키보드로부터 두 정수와 계산하고자 하는*

*연산자를 입력받은 후 Add, Sub, Mul, Div 중에서이 연산을 실행할 수 있는 객체를 생성하고*

*setValue()와 calculate()를 호출하여 그 결과 값을 화면에 출력한다.*

*\*/*

public final void run() {

final int a = 10, b = 2;

CalculatorBase base = null;

base = new Add();

base.setValue(a, b);

System.out.println(Integer.toString(a) + " + " + Integer.toString(b) + " = " + Integer.toString(base.calculate()));

base = new Sub();

base.setValue(a, b);

System.out.println(Integer.toString(a) + " - " + Integer.toString(b) + " = " + Integer.toString(base.calculate()));

base = new Mul();

base.setValue(a, b);

System.out.println(Integer.toString(a) + " \* " + Integer.toString(b) + " = " + Integer.toString(base.calculate()));

base = new Div();

base.setValue(a, b);

System.out.println(Integer.toString(a) + " / " + Integer.toString(b) + " = " + Integer.toString(base.calculate()));

}

public abstract class CalculatorBase {

protected int a;

protected int b;

public abstract void setValue(final int *a*, final int *b*);

public abstract int calculate();

}

public class Add extends CalculatorBase {

public Add(){}

public final void setValue(final int *a*, final int *b*){

this.a = a;

this.b = b;

}

public final int calculate(){

return a + b;

}

}

public class Sub extends CalculatorBase {

public Sub(){}

public final void setValue(final int *a*, final int *b*){

this.a = a;

this.b = b;

}

public final int calculate(){

return a - b;

}

}

public class Mul extends CalculatorBase {

public Mul(){}

public final void setValue(final int *a*, final int *b*){

this.a = a;

this.b = b;

}

public final int calculate(){

return a \* b;

}

}

public class Div extends CalculatorBase {

public Div(){}

public final void setValue(final int *a*, final int *b*){

this.a = a;

this.b = b;

}

public final int calculate(){

return a / b;

}

}

}

Chapter 4 problem 6

package com.SHGroup.homework.chapter4.problem6;

import com.SHGroup.homework.HomeworkClass;

import java.util.Scanner;

public class Homework\_4\_6 extends HomeworkClass {

*/\**

*간단한 공연 예약 시스템을 만들어보자.*

*다수의 클래스를 다루고 객체의 배열을 다루기에는 아직 자바로 프로그램 개발이 익숙하지 않은*

*초보자에게 다소 무리가 있을 것이다. 그러나 반드시 넘어야 할 산이다.*

*이 도전 주제를 통해 산을 넘어갈 수 있는 체력을 키워보자. 공연 예약 시스템의 내용은 다음과 같다.*

*- 공연은 하루에 한 번 있다.*

*- 좌석은 S석, A석, B석 타입이 있으며 모두 10석의 좌석이 있다.*

*- 공연 예약 시스템의 메뉴는 "예약", "조회", "취소", "끝내기"가 있다.*

*- 예약은 한 자리만 예약할 수 있고 좌석 타입, 예약자 이름, 좌석 번호를 순서대로 입력받아 예약한다.*

*- 조회는 모든 종류의 좌석을 표시한다.*

*- 취소는 예약자의 이름을 입력하여 취소한다.*

*- 없는 이름, 없는 번호, 없는 메뉴, 잘못된 취소 등에 대해서 오류 메시지를 출력하고 사용자가 다시 시도하도록 한다.*

*\*/*

private static final int ARR\_MAX = 10;

public final void run() {

final Scanner sc = new Scanner(System.in);

final Concert c = new Concert();

while (true) {

System.out.print("예약(1), 조회(2), 취소(3), 끝내기(4) >>");

int choice = sc.nextInt();

if (choice == 1) {

System.out.print("좌석 구분 S, A, B >>");

SeatType st = SeatType.getSeatTypeFromName(sc.next());

if(st == null){

System.out.println("좌석 구분이 올바르지 않습니다.");

continue;

}

c.show(st);

System.out.print("이름 : ");

String name = sc.nextLine();

System.out.print("번호 : ");

int number = sc.nextInt();

boolean b = c.doReservation(st, number, name);

if(b){

System.out.println("예약에 성공하였습니다.");

}else{

System.out.println("예약에 실패하였습니다.");

}

} else if (choice == 2) {

for(SeatType st : SeatType.values())

c.show(st);

System.out.println("<< 조회를 완료하였습니다. >>");

} else if (choice == 3) {

System.out.print("좌석 구분 S, A, B >>");

SeatType st = SeatType.getSeatTypeFromName(sc.next());

if(st == null){

System.out.println("좌석 구분이 올바르지 않습니다.");

continue;

}

c.show(st);

System.out.print("이름 : ");

String name = sc.nextLine();

boolean cancel = c.doCancel(st, name);

if(cancel){

System.out.println("<< 예약을 취소하였습니다. >>");

}else{

System.out.println("<< 예약을 취소하지 못하였습니다. >>");

}

} else if (choice == 4) {

break;

}

}

sc.close();

}

public class Concert {

private final String[] arrS = new String[ARR\_MAX];

private final String[] arrA = new String[ARR\_MAX];

private final String[] arrB = new String[ARR\_MAX];

public Concert(){

for(int i = 0 ; i < ARR\_MAX ; i ++){

arrS[i] = null;

arrA[i] = null;

arrB[i] = null;

}

}

public final boolean hasReservationed(final SeatType *type*, final int *number*){

if(!check(type, number))

return false;

return getArray(type)[number - 1] != null;

}

public final boolean doReservation(final SeatType *type*, final int *number*, final String *name*){

if(!check(type, number))

return false;

if(name == null)

return false;

getArray(type)[number - 1] = name;

return true;

}

public final boolean doCancel(final SeatType *type*, final String *name*){

if(type == null)

return false;

if(name == null)

return false;

String[] arr = getArray(type);

for(int i = 0 ; i < ARR\_MAX ; i ++){

if(arr[i] != null

&& arr[i].equals(name)){

arr[i] = null;

return true;

}

}

return false;

}

public final void show(final SeatType *type*){

if(type == null)

return;

System.out.print(type.name() + ">>");

for(String n : getArray(type)){

if(n == null){

System.out.print(" ---");

}else{

System.out.print(" " + n);

}

}

}

private final boolean check(final SeatType *type*, final int *number*){

if(type == null)

return false;

if(number < 1 || number > ARR\_MAX)

return false;

return true;

}

private final String[] getArray(final SeatType *type*){

if(type == SeatType.S){

return arrS;

}else if(type == SeatType.A){

return arrA;

}else if(type == SeatType.B){

return arrB;

}

return null;

}

}

public enum SeatType{

S, A, B;

public static SeatType getSeatTypeFromName(final String *name*){

for(final SeatType st : values()){

if(st.name().equals(name)){

return st;

}

}

return null;

}

}

}

Chapter 5 problem 2

package com.SHGroup.homework.chapter5.problem2;

import com.SHGroup.homework.HomeworkClass;

import com.SHGroup.homework.chapter4.problem5.Homework\_4\_5;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.Vector;

public class Homework\_5\_2 extends HomeworkClass {

*/\**

*4장 연습문제 중 하나를 수정하여 완성해보자.*

*더하기(+), 빼기(-), 곱하기(\*), 나누기(/)를 수해애하는 각 클래스 Add, Sub, Mul, Div를 만들어라.*

*이들은 모두 다음 필드와 메소드를 가진다.*

*- int 타입의 a, b, 필드 : 연산하고자 하는 피연산자*

*- void setValue(int a, int b) : 피연산자를 객체 내에 설정한다.*

*- int calculate() : 해당 클래스의 목적에 맞는 연산을 실행하고 그 결과를 리턴한다.*

*Add, Sub, Mul, Div 클래스에 공통된 필드와 메소드가 존재하므로 새로운 추상 클래스 Calc를 정의하고*

*이들이 Calc를 상속받게 하라. main() 메소드에서는 다음 실행 사례의 그림과 같이 키보드로부터*

*계산하고자 하는 연산자와 두 정수를 입력받은 후, Add, Sub, Mul, Div 중에서*

*이 연산을 시행할 수 있는 객체를 생성하고 setValue()와 calculate()를 호출하여 그 결과 값을 화면에 출력하라.*

*\*/*

public final void run() {

final int a = 20, b = 2;

CalculatorBase base = null;

base = new Add();

base.setValue(a, b);

System.out.println(Integer.toString(a) + " + " + Integer.toString(b) + " = " + Integer.toString(base.calculate()));

base = new Sub();

base.setValue(a, b);

System.out.println(Integer.toString(a) + " - " + Integer.toString(b) + " = " + Integer.toString(base.calculate()));

base = new Mul();

base.setValue(a, b);

System.out.println(Integer.toString(a) + " \* " + Integer.toString(b) + " = " + Integer.toString(base.calculate()));

base = new Div();

base.setValue(a, b);

System.out.println(Integer.toString(a) + " / " + Integer.toString(b) + " = " + Integer.toString(base.calculate()));

}

public abstract class CalculatorBase {

protected int a;

protected int b;

public abstract void setValue(final int *a*, final int *b*);

public abstract int calculate();

}

public class Add extends CalculatorBase {

public Add(){}

public final void setValue(final int *a*, final int *b*){

this.a = a;

this.b = b;

}

public final int calculate(){

return a + b;

}

}

public class Sub extends CalculatorBase {

public Sub(){}

public final void setValue(final int *a*, final int *b*){

this.a = a;

this.b = b;

}

public final int calculate(){

return a - b;

}

}

public class Mul extends CalculatorBase {

public Mul(){}

public final void setValue(final int *a*, final int *b*){

this.a = a;

this.b = b;

}

public final int calculate(){

return a \* b;

}

}

public class Div extends CalculatorBase {

public Div(){}

public final void setValue(final int *a*, final int *b*){

this.a = a;

this.b = b;

}

public final int calculate(){

return a / b;

}

}

}

Chapter 5 problem 5

package com.SHGroup.homework.chapter5.problem5;

import com.SHGroup.homework.HomeworkClass;

public class Homework\_5\_5 extends HomeworkClass {

*/\**

*추상 클래스의 서브 클래스 만들기에 필요한 추상 메소드 오버라이딩과 super()의 사용에 관한 문제이다. 다음과 같은 MyPoint 추상 클래스가 있다.*

*abstract class MyPoint {*

*int x;*

*int y;*

*public MyPoint(int x, int y) {*

*this.x = x; this.y = y;*

*}*

*protected abstract void move(int x, int y); // 새로운 x, y 위치로 이동*

*protected abstract void reverse(); // (x, y)에서 (y, x)로 위치 변경*

*protected void show() {*

*System.out.println(x+","+y);*

*}*

*}*

*MyPoint를 상속받는 MyColorPoint 클래스를 작성하라.*

*MyColorPoint의 생성자는 MyColorPoint(int x, int y, String color)로 하라. 그리고 다음과 같은 main() 메소드를 삽입하여 실행되도록 하라.*

*public static void main(String[] args) {*

*MyPoint p = new MyColorPoint(2, 3, "blue");*

*p.move(3,4);*

*p.reverse();*

*p.show();*

*}*

*\*/*

public final void run() {

MyPoint p = new MyColorPoint(2, 3, "blue");

p.move(3,4);

p.reverse();

p.show();

}

public abstract class MyPoint {

protected int x;

protected int y;

public MyPoint(int *x*, int *y*) {

this.x = x; this.y = y;

}

protected abstract void move(final int *x*, final int *y*); *// 새로운 x, y 위치로 이동*

protected abstract void reverse(); *// (x, y)에서 (y, x)로 위치 변경*

protected void show() {

System.out.println(x+","+y);

}

}

public class MyColorPoint extends MyPoint {

private final String color;

public MyColorPoint(final int *x*, final int *y*, final String *color*) {

super(x, y);

this.color = color;

}

@Override

protected void move(final int *x*, final int *y*){

this.x = x;

this.y = y;

}

@Override

protected void reverse(){

x ^= y;

y ^= x;

x ^= y;

}

@Override

protected void show() {

System.out.println(x + "," + y + "," + color);

}

}

}

Chapter 6 problem 1

package com.SHGroup.homework.chapter6.problem1;

import com.SHGroup.homework.HomeworkClass;

public class Homework\_6\_1 extends HomeworkClass {

*/\**

*다음 main() 메소드의 실행 결과 "MyPoint(3, 20)"이 출력되도록 MyPoint 클래스를 작성하라.*

*public static void main(String args[]) {*

*MyPoint a = new MyPoint(3, 20);*

*System.out.println(a);*

*}*

*\*/*

public final void run() {

MyPoint a = new MyPoint(3, 20);

System.out.println(a);

}

public final class MyPoint {

private final int x, y;

public MyPoint(int *x*, int *y*){

this.x = x;

this.y = y;

}

public String toString(){

return "MyPoint(" + x + ", " + y + ")";

}

}

}

Chapter 6 problem 4

package com.SHGroup.homework.chapter6.problem4;

import com.SHGroup.homework.HomeworkClass;

public class Homework\_6\_4 extends HomeworkClass {

*/\**

*int 타입의 x, y, radius 필드를 가지는 Circle 클래스를 작성하라.*

*equals() 메소드를 재정의하여 두 개의 Circle 객체의 반지름이 같으면*

*두 Circle 객체가 동일한 것으로 판별하도록 하라.*

*Circle 클래스의 생성자는 3개의 인자를 가지며 x, y, radius 필드를 인자로 받아 초기화한다.*

*\*/*

public final void run() {

Circle c = new Circle(1, 1, 5);

Circle c2 = new Circle(6, 2, 5);

Circle c3 = new Circle(6, 2, 3);

System.out.println("c == c2 : " + c.equals(c2));

System.out.println("c == c3 : " + c.equals(c3));

System.out.println("c2 == c3 : " + c2.equals(c3));

}

public class Circle {

private final int x, y, radius;

public Circle(int *x*, int *y*, int *radius*){

this.x = x;

this.y = y;

this.radius = radius;

}

@Override

public boolean equals(Object *o*){

if(!(o instanceof Circle)){

return false;

}

Circle c = (Circle)o;

return c.radius == this.radius;

}

}

}

Chapter 7 problem 7

package com.SHGroup.homework.chapter7.problem7;

import com.SHGroup.homework.HomeworkClass;

import java.util.HashMap;

import java.util.Vector;

public class Homework\_7\_7 extends HomeworkClass {

*/\**

*아래의 HV 클래스는 해쉬맵을 인자로 받아 벡터를 리턴하는 hashToVector() 메소드를 가지고 있다.*

*이 메소드는 해쉬맵 내의 '값(value)' 을 모두 Vector<String>에 삽입하여 리턴한다.*

*hashToVector()를 작성하라.*

*\*/*

public final static Vector<String> hashToVector(final HashMap<String, String> *h*){

return new Vector<String>(h.values());

}

public final void run() {

HashMap<String, String> h = new HashMap<String, String>();

h.put("범죄", "112");

h.put("화재", "119");

h.put("전화번호", "114");

Vector<String> v = Homework\_7\_7.hashToVector(h);

for(int n = 0 ; n < v.size(); n ++){

System.out.println(v.get(n));

}

}

}

Chapter 7 problem 8

package com.SHGroup.homework.chapter7.problem8;

import com.SHGroup.homework.HomeworkClass;

public class Homework\_7\_8 extends HomeworkClass {

*/\**

*다음은 String만 다루는 MyClass 코드이다. MyClass 를 제네릭 클래스 MyClass<E>로 일반화하고,*

*이를 이용하는 main() 메소드를 만들어 프로그램을 완성하라.*

*\*/*

public final void run() {

MyClass<String> mc = new MyClass<>("blabla");

mc.setS("hello");

System.out.println("haha : " + mc.getS());

}

public final class MyClass<T> {

private T s;

public MyClass(T *s*){

this.s = s;

}

public void setS(T *s*){

this.s = s;

}

public T getS(){

return s;

}

}

}

Chapter 7 problem 9

package com.SHGroup.homework.chapter7.problem9;

import com.SHGroup.homework.HomeworkClass;

import java.util.HashMap;

import java.util.Scanner;

public class Homework\_7\_9 extends HomeworkClass {

*/\**

*하나의 학생 정보는 Student 클래스로 표현한다.*

*Student 클래스에는 이름, 학과, 학번, 학점 평균을 나타내는 필드가 있다.*

*여기서 학번을 String 타입으로 선언한다.*

*키보드에서 학생 정보를 5개 입력받아 학번을 '키'로 하여 HashMap<String, Student> 에*

*저장하고 학번으로 학생 정보를 검색하여 출력하는 프로그램을 작성하여라.*

*다음과 같은 해시맵을 이용하라.*

*\*/*

private final HashMap<String, Student> dept = new HashMap<String, Student>();

public final void run() {

Scanner sc = new Scanner(System.in);

for(int i = 0 ; i < 5 ; i ++){

String name, type, number;

int avg;

System.out.println("Input [" + (i+1) + "] student info");

System.out.println("Input name");

name = sc.nextLine();

System.out.println("Input type");

type = sc.nextLine();

System.out.println("Input number");

number = sc.nextLine();

System.out.println("Input avg");

avg = sc.nextInt();

dept.put(type, new Student(name, type, number, avg));

}

while(true){

System.out.println("Input search number");

String input = sc.nextLine();

if(dept.containsKey(input)){

dept.get(input).showInfo();

}else{

System.out.println("Not Found..");

}

}

}

public final class Student {

private final String name, type, number;

private final int avg;

public Student(final String *name*, final String *type*, final String *number*, final int *avg*) {

this.name = name;

this.type = type;

this.number = number;

this.avg = avg;

}

public void showInfo(){

System.out.println("[ " + name + "]");

System.out.println(" 학과 : " + type + "]");

System.out.println(" 학번 : " + number + "]");

System.out.println(" 평균 : " + avg + "]");

}

}

}

Chapter 7 problem 10

package com.SHGroup.homework.chapter7.problem10;

import com.SHGroup.homework.HomeworkClass;

import java.util.HashMap;

import java.util.Scanner;

public class Homework\_7\_10 extends HomeworkClass {

*/\**

*지명과 위치 정보를 저장하고 관리하고자 한다. 지명은 String 타입으로,*

*위치 정보를 위도와 경도를 가진 Location 이라는 클래스로 선언하고,*

*지명을 '키'로 하여 HashMap<String, Location>에 키보드로 입력받아 5개의 요소를 저장하고,*

*지명으로 검색하는 프로그램을 작성하라*

*\*/*

private final HashMap<String, Location> locs = new HashMap<String, Location>();

public final void run() {

Scanner sc = new Scanner(System.in);

for(int i = 0 ; i < 5 ; i ++){

String name;

double lat, lon;

System.out.println("Input [" + (i+1) + "] place info");

System.out.println("Input name");

name = sc.nextLine();

System.out.println("Input latitude");

lat = sc.nextDouble();

System.out.println("Input longitude");

lon = sc.nextDouble();

locs.put(name, new Location(name, lat, lon));

}

while(true){

System.out.println("Input place name");

String input = sc.nextLine();

if(locs.containsKey(input)){

locs.get(input).showInfo();

}else{

System.out.println("Not Found..");

}

}

}

public final class Location {

private final String name;

private final double lat, lon;

public Location(final String *name*, final double *lat*, final double *lon*){

this.name = name;

this.lat = lat;

this.lon = lon;

}

public void showInfo(){

System.out.println("-- [ " + name + " ] --");

System.out.println(" 위도 : " + lat);

System.out.println(" 경도 : " + lon);

}

}

}

Chapter 8 problem 4

package com.SHGroup.homework.chapter8.problem4;

import com.SHGroup.homework.HomeworkClass;

import java.io.\*;

import java.util.Scanner;

public class Homework\_8\_4 extends HomeworkClass {

*/\**

*사용자로부터 두 개의 텍스트 파일 이름을 입력받고 첫 번째 파일 뒤에*

*두 번째 파일을 덧붙여 새로운 파일을 생성하는 프로그램을 작성하라.*

*\*/*

public final void run() {

Scanner sc = new Scanner(System.in);

System.out.println("First filename");

String first = sc.nextLine();

System.out.println("Second filename");

String second = sc.nextLine();

try{

File f1 = new File(first);

File f2 = new File(second);

String all = "";

all += readFile(f1);

all += readFile(f2);

writeFile(new File("out.txt"), all);

}catch(Exception *ex*){

ex.printStackTrace();

}

}

public String readFile(File *f*) throws IOException {

BufferedReader br = new BufferedReader(new FileReader(f));

String all = "";

String n;

while((n = br.readLine()) != null){

all += (all.isEmpty()? n : "\n" + n);

}

br.close();

return all;

}

public void writeFile(File *f*, String *n*) throws IOException{

BufferedWriter br = new BufferedWriter(new FileWriter(f));

br.write(n);

br.flush();

br.close();

}

}

Chapter 8 problem 6

package com.SHGroup.homework.chapter8.problem6;

import com.SHGroup.homework.HomeworkClass;

import java.io.\*;

import java.util.Scanner;

public class Homework\_8\_6 extends HomeworkClass {

*/\**

*자바 소스 파일을 읽어들여 맨 앞에 행 번호를 붙여서 화면에 출력하는 프로그램을 작성하라.*

*\*/*

public final void run() {

Scanner sc = new Scanner(System.in);

System.out.println("Input filename");

String source = sc.nextLine();

try {

File output = new File(source);

String out = readFileWithLine(output);

writeFile(output, out);

}catch(Exception *ex*){

ex.printStackTrace();

}

}

public String readFileWithLine(File *f*) throws IOException {

BufferedReader br = new BufferedReader(new FileReader(f));

String all = "";

int line = 0;

String n;

while((n = br.readLine()) != null){

all += (line == 0 ? "" : "\n") + (line + 1) + " " + n;

line ++;

}

br.close();

return all;

}

public void writeFile(File *f*, String *n*) throws IOException{

BufferedWriter br = new BufferedWriter(new FileWriter(f));

br.write(n);

br.flush();

br.close();

}

}