201935266 박영준 JAVA 3차 과제

1번 문제

**package** HW;

//1번째 과제 "제네릭을 사용하여서 똑같은 타입의 객체 두 개를 저장하는 pair 클래스를 작성하여 보자" 입니다.

**import** java.util.Scanner;

**class** Mypair<T>

{

**static** Scanner *input* = **new** Scanner(System.***in***);

**private** T val1;

**private** T val2; // 제네릭 변수

**public** Mypair(T c1,T c2) //생성자

{

**this**.val1 = c1;

**this**.val2 = c2;

}

**public** **void** setval()

{

System.***out***.println("첫번째 변수 입력");

**this**.val1=(T) *input*.next();

System.***out***.println("두번째 변수 입력");

**this**.val2=(T) *input*.next();

}

/\* getval의 경우,parameter로 일반 자료형(\*String,int,char 등)을 넘겨주면 call by value로 구현되어 메인함수에 값이 전달되지 않으므로,

call by reference를 구현하기 위해서는 반드시 class형으로 넘겨주어야 한다,이를 위해 Mypair안에 Stringclass를 별도로 생성해 주었다

\*/

**public** **void** getval(StringClass v1,StringClass v2)

{

v1.str = (String) **this**.val1;

v2.str = (String) **this**.val2;

}

**public** String toString()

{

String returnval="";

returnval = "Val1 = " +(String)**this**.val1 + " Val2 = " +(String)**this**.val2;

**return** returnval;

}

}

//getval의 parameter가 class형이 되도록 하게 위한 클래스

**class** StringClass

{

String str="";

StringClass(String st) //StringClass의 생성자

{

**this**.str=st;

}

}

**public** **class** MypairTest

{

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

{

Mypair<String> fruit = **new** Mypair<String>("사과","포도");

System.***out***.println("fruit 생성 완료!");

System.***out***.println("현재 fruit에 저장된 " + fruit.toString()+"\n");

System.***out***.println("Setval 함수 실행!");

fruit.setval();

System.***out***.println("현재 fruit에 저장된 " + fruit.toString()+"\n");

String empty = "empty";

StringClass value1= **new** StringClass("empty");

StringClass value2= **new** StringClass("empty");

System.***out***.print("현재 value 1에 저장된 값은 "+ value1.str +"\n" + "현재 value 2에 저장된 값은 "+ value2.str+"\n");

fruit.getval(value1,value2);

System.***out***.println("getval 함수 실행!");

System.***out***.println("현재 value 1에 저장된 값은 "+ value1.str +"\n"+ "현재 value 2에 저장된 값은 "+ value2.str+"\n");

System.***out***.println("toString() 함수 실행!");

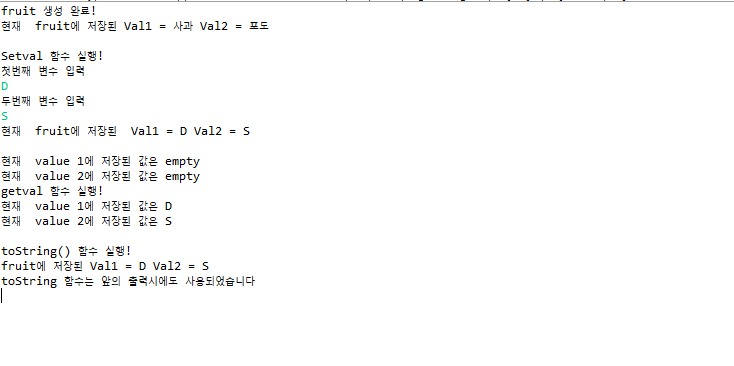
System.***out***.println("fruit에 저장된 "+fruit.toString());

System.***out***.println("toString 함수는 앞의 출력시에도 사용되었습니다");

}

}

실행 결과



2번째 문제

**package** HW;

// 2번째 과제 "클래스 안에서 하나의 메소드만 제네릭으로 만들어보자" 입니다.

**class** Test<T>

{

**public** Test(){};

**public** **void** a(T par)

{

System.***out***.println("매개변수의 클래스 이름은 "+ par.getClass().getName()+", 값은 "+par+"\n"+"\n");

}

}

**class** ClassEx

{

**public** ClassEx(){};

};

**public** **class** TestTest

{

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

{

Test k=**new** Test();

System.***out***.println("int i = 10;");

**int** i = 10;

k.a(i);

System.***out***.println("float f = 3.244F;");

**float** f = 3.244F;

k.a(f);

System.***out***.println("double d = 2.1726261;");

**double** d = 2.1726261;

k.a(d);

System.***out***.println("boolean b = true;");

**boolean** b = **true**;

k.a(b);

System.***out***.println("ClassEx c = new ClassEx();");

ClassEx c = **new** ClassEx();

k.a(c);

System.***out***.println("String s =\"String\";");

String s ="String";

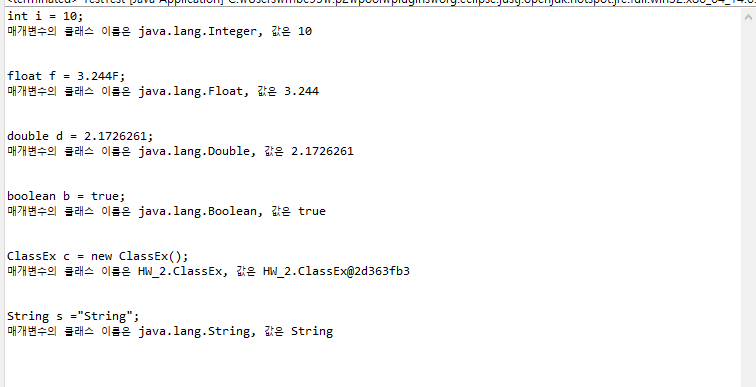
k.a(s);

// **TODO** Auto-generated method stub

}

}

실행 결과



8장 실습문제 10번

**package** HW8\_10;

**import** java.util.\*;

**import** java.io.\*;

**public** **class** HW\_8\_10

{

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

{

HashMap<String,String> PhoneBook = **new** HashMap<String,String>();

**int** human = 0 ; // 사람 숫자

Scanner input = **new** Scanner(System.***in***);

InputStreamReader in = **null**;

InputStreamReader in1 = **null**; // b에 물려줄 inputstream

FileInputStream FR = **null**;

**try**

{

FR = **new** FileInputStream("Phone.txt");

**try**

{

in = **new** InputStreamReader(FR, "MS949");

**int** c=0; // read 처리에 사용되는 변수

**try**

{

**while** (c != -1)

{

**char**[] temp1 = **new** **char**[3]; // 이름을 저장할 3칸짜리 배열 생성

**char**[] temp2 = **new** **char**[14]; // 전화번호를 저장할 13칸짜리 배열 생성

String name = "";

String tel = "";

**for**(**int** i = 0 ; i <3; i++)

{

c=in.read();

temp1[i]=(**char**)c;

}

name = String.*valueOf*(temp1);

c= in.read();

/\*이름과 번호 사이 한 칸 띄어쓴 부분은 temp에 기록하지 않고 그냥 지나친다,이는 Phone.txt가

\* 박영준 010-2523-2512

\* 잘생김 010-2523-2512

\* 의 식으로 줄바꿈으로 사람과 번호를 구분해 놓았기 때문이다.

\*/

**for**(**int** j = 0 ; j <14; j++)

{

c=in.read();

temp2[j]=(**char**)c;

}

tel = String.*valueOf*(temp2);

PhoneBook.put(name, tel);

c= in.read();

human++;

}

}

**catch** (IOException e) // I/O 입력 중 발생한 오류 처리

{

System.***out***.println("I/O Error,please call 컴사랑119");

// **TODO** Auto-generated catch block

}

}

**catch** (UnsupportedEncodingException e) //파일 인코딩중 발생한 오류 처리

{

System.***out***.println("Wrong Encoding Error! please try using different encoding");

// **TODO** Auto-generated catch block

}

} **catch** (FileNotFoundException e)

{

System.***out***.println("Error! file could not be found there.");

// **TODO** Auto-generated catch block

}

System.***out***.println("총 "+human+"개의 전화번호를 읽었습니다.");

String keyboardinput= "";

**while**(!(keyboardinput.equals("그만")))

{

keyboardinput = "";

String SearchResult="";

System.***out***.print("이름 >> ");

keyboardinput = input.next();

SearchResult =(String)PhoneBook.get(keyboardinput);

**boolean** keyexists = PhoneBook.containsKey(keyboardinput); // 키보드로 입력한 값이 해시맵에 포함되었는지 여부를 체크

**if**(keyboardinput.equals("그만")) // 키보드로 입력한 값이 해쉬맵에 존재하는지 찾기 전에,그 값이 "그만"인지 먼저 체크.

{

**break**;

}

**else** **if**(keyexists == **false**)

{

System.***out***.println("찾는 이름이 없습니다.");

}

**else**

{

System.***out***.println(SearchResult);

}

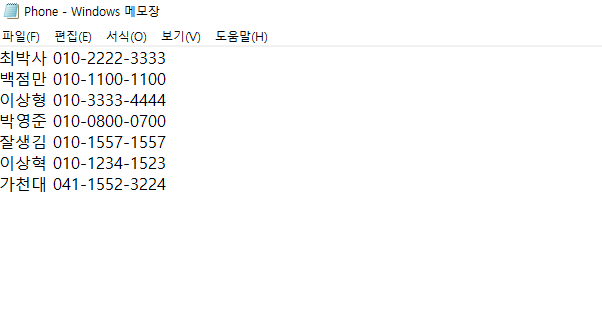
}

// **TODO** Auto-generated method stub

}

}

Phone 파일 내용



실행 결과



9장 실습문제 6번

**package** HW9\_6;

**import** javax.swing.\*;

**import** javax.swing.event.\*;

**import** java.awt.\*;

**import** java.awt.event.\*;

**import** java.math.\*;

**class** MyFrame **extends** JFrame

{

**public** MyFrame()

{

setTitle("Random Labels");

setVisible(**true**);

setSize(300,300);

Container c = getContentPane();

c.setLayout(**null**); // 배치관리자 삭제

**for**(**int** i=0;i<20;i++) //랜덤 위치에 점 배치

{

JLabel L = **new** JLabel();

**int** x = (**int**)(Math.*random*()\*200)+50; // 50~250

**int** y = (**int**)(Math.*random*()\*200)+50; // 50~250

L.setSize(10,10); // 크기를 10으로 설정

L.setBackground(Color.***BLUE***);

L.setOpaque(**true**);

L.setLocation(x,y);

c.add(L);

}

}

}

**public** **class** HW\_9\_6

{

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

{

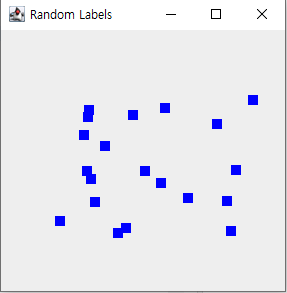
MyFrame FR = **new** MyFrame();

// **TODO** Auto-generated method stub

}

}

실행 결과



9장 8번

**package** HW9\_8;

**import** java.awt.\*;

**import** java.awt.event.\*;

**import** javax.swing.\*;

**import** javax.swing.event.\*;

**import** java.math.\*;

**import** java.awt.BorderLayout;

**import** java.awt.Color;

**import** java.awt.Component;

**import** java.awt.Container;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JPanel;

**import** javax.swing.JTextField;

**class** MyFrame **extends** JFrame

{

**public** MyFrame()

{

setTitle("여러 개의 패널을 가진 프레임"); // 프레임의 타이틀 달기

Container c = getContentPane();

c.setLayout(**new** BorderLayout());

c.add(**new** NorthPanel(),BorderLayout.***NORTH***); // North 패널 추가

c.add(**new** CenterPanel(),BorderLayout.***CENTER***); // Center 패널 추가

c.add(**new** SouthPanel(),BorderLayout.***SOUTH***); // South 패널 추가

setSize(400,400);

setVisible(**true**);

}

}

**class** NorthPanel **extends** JPanel // North에 부착할 패널

{

**public** NorthPanel()

{

setBackground(Color.***GRAY***);

**this**.add(**new** JButton("열기"));

**this**.add(**new** JButton("닫기"));

**this**.add(**new** JButton("나가기"));

}

}

**class** CenterPanel **extends** JPanel // CENTER에 부착할 패널

{

**public** CenterPanel()

{

**this**.setLayout(**null**); // 배치관리자 삭제

**int** x;

**int** y;

JLabel L;

**for**(**int** i=0;i<10;i++)

{

x = (**int**)(Math.*random*()\*300);

y = (**int**)(Math.*random*()\*300);

L = **new** JLabel("\*");

L.setForeground(Color.***red***);

L.setLocation(x, y);

L.setSize(50,10);

**this**.add(L);

}

}

}

**class** SouthPanel **extends** JPanel // South에 부착할 패널

{

**public** SouthPanel()

{

**this**.setLayout(**new** FlowLayout(FlowLayout.***LEFT***)); // Flowlayout 배치관리자 부착

setBackground(Color.***yellow***);

**this**.add(**new** JButton("Word Input"));

**this**.add(**new** JTextField(24));

}

}

**public** **class** HW\_9\_8

{

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

{

MyFrame FR = **new** MyFrame();

// **TODO** Auto-generated method stub

}

}

실행 결과

