**Mobile Software Programming Homework #5**

1. (50점) 이진탐색을 하는 클래스를 generic과 배열을 사용하여 작성하시오. 이진 탐색의 대상은 20개 크기의 배열로 주어지며 정수와 실수에 대하여 이진탐색이 가능해야 한다. 또한 이진 탐색은 boolean BSearch(T arr[], T target) 메소드를 통해 구현해야 하며 별도의 클래스를 통해 구현해야 한다. 이진 탐색 알고리즘은 다음과 같다. 아래 내용은 탐색하려는 값이 있는 경우 true를 없는 경우 false를 리턴한다.

while(low <= high) {

mid = (low + high) / 2;

if (arr[mid] == target)

return true;

else if (arr[mid] > target)

high = mid - 1;

else

low = mid + 1;

}

return false;

이진 탐색을 위한 데이터는 정수형, 실수형의 2가지 배열로 구성하며 배열의 내용은 다음과 같이 구성한다.

{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}

{1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0}

탐색 데이터는 15, 14.5 로 한다.

Generic의 임의의 자료형에서 비교연산을 수행하기 위해서는 Comparator 인터페이스를 오버라이드하거나 또는 Comparable 클래스를 상속해서 이를 통해 비교연산을 수행하여야 한다. Comaprator 또는 아래의 코드는 Comparable 클래스를 활용하여 비교연산을 수행하는 예이다.

아래의 예에서는 임의의 자료형에서 비교연산을 하기 위한 isGreater(T a, T b)를 구현하였으며, 이를 통해 비교연산(<, >, =)을 구현하고 있다. 아래 코드를 참고하여 BSearch 함수를 구현하도록 한다.

import java.util.\*;

class Box<T> {

T value;

public Box(T v) {

value = v;

}

public T getValue() {

return value;

}

}

public class HelloWorld{

**public static <T extends Comparable<T>> int isGreater(T a,T b) {**

**if(a.compareTo(b) > 0) {**

**return 1;**

**} else if(b.compareTo(a) > 0) {**

**return -1;**

**} else {**

**return 0;**

**}**

**}**

public static void main(String []args){

Box<Integer> t1 = new Box<Integer>(3);

Box<Integer> t2 = new Box<Integer>(2);

System.out.println(**isGreater**(t1.getValue(), t2.getValue()));

Box<Double> t3 = new Box<Double>(3.0);

Box<Double> t4 = new Box<Double>(2.0);

System.out.println(**isGreater**(t3.getValue(), t4.getValue()));

Box<String> t5 = new Box<String>("Kim");

Box<String> t6 = new Box<String>("Woongsup");

System.out.println(**isGreater**(t5.getValue(), t6.getValue()));

}

}

여기서 **<T extends Comparable<T>>**는 T의 자료형을 제한하기 위한 용도로 사용되며, 이를 통해 T는 임의의 모든 자료형에 대해 해당 함수가 적용되는 것이 아닌 비교가 가능한 자료형에 대해서만 T를 적용할 수 있도록 제한한다. 다른 말로 말하면 **<T extends Comparable<T>>** 를 통해서 T에 적용되는 자료형은 <, >, = 연산이 가능한 자료형만 T로 사용할 수 있도록 제한하는 효과를 가지게 된다.

1. (50 점) 이전 숙제에서 구현한 Person, Employee, Student 클래스를 활용하여 생성된 모든 인스턴스들의 이름을 출력하는 루틴을 vector()를 활용하여 구현하라. 각 클래스들의 인스턴스들은 모두 지난 숙제의 예를 그래도 활용하도록 한다. 이 때 vector는 vector<Person>()으로 선언하도록 한다.