Report

Lab 7

제출일: 6/02

과목명: 자바프로그래밍1

담당교수: 박경신 교수님

전공: 컴퓨터공학과

학번: 32191826

이름: 박주희

<ArrayList>

배열은 항상 크기를 선언해주고 사용하여야 한다. 그와 다르게 ArrayList는 크기를 선언하지 않아고 메모리가 허용하는 한 계속해서 추가할 수 있도록 만든 자료형 클래스이다.

데이터의 중복을 허용한다.

<HashMap>

key와 value를 묶어 하나의 entry로 저장한다. 그리고 hashing을 사용하기 때문에 많은양의 데이터를 검색하는데 뛰어난 성능을 보인다.

키는 중복을 허용하지 않으며 값의 중복을 허용한다.

<ArrayList vs HashMap>

|  |  |  |
| --- | --- | --- |
|  | 장점 | 단점 |
| ArrayList | 배열이 자동으로 늘어나게 되어 간편하다. | 원하는 값을 찾을 때 인덱스의 번호를 외워야하는 번거로움이 생긴다. |
| HashMap | ArrayList에서 key값을 추가한 것으로 쉽게 유추하여 데이터를 꺼내올 수 있다.  삽입,삭제 시간이 매우 빠르다.  요소검색이 빠르다. | Key를 직접 입력해야 하는 번거로움이 생긴다. |

<PeriodicTable/PeriodicTableList/PeriodicTableMap>

PeriodicTable의 find(IperiodicElementFinder ifinder) 메소드는 for-each문을 이용하여 조건을 성립하는 원소의 개수를 센 다음 그 크기만큼 pe를 생성하였다. 그리고 pe에 조건을 성립하는 원소들을 차례로 저장하였다. 하지만 PeriodicTableList의 find(IperiodicElementFinder ifinder) 메소드는 배열형이 아닌 List형으로 pe를 생성하였기 때문에 바로 add하면 된다. 즉, PeriodicTable처럼 원소의 개수를 세고 배열을 생성하지 않아도 됨으로써 더 간단해진 것을 알 수 있다.

PeriodicTableMap에서는 Element를 계속 입력해야 해서 pe로 Element를 계속 찾아 입력해야 되는 것이 번거로웠다.

Lab7 main함수에서 pmap add test하는데 어려움이 있었다. Map은 PeriodicTableList와 다르게 Element와 PeriodicElement를 하나의 entry로 저장하므로 Element형 배열 e1을 생성하고 pe3의 원소들의 이름을 Element클래스의 nameOf 메소드에 입력하여 그에 해당하는 Element를 e1에 차례로 저장하려고 하였다. 계속된 오류에 element의 이름을 보니 Lab7 클래스의 elements 배열은 원소 이름이 모두 대문자, Element클래스의 nameOf 메소드에서는 원소이름이 소문자로 저장되어 있었다. 그렇기 때문에 Element.nameOf(pe3[i].getName());가 성립하지 않았던 것이다.

Lab7 #155

String b= a.substring(0,1) +a.substring(1,a.length()).toLowerCase();

첫글자를 제외한 모든 글자를 소문자로 변환하여 b에 저장하여 Element.*nameOf*(b);로 고치니 잘 돌아갔다.

PeriodicTableMap 클래스의 getRandomElement()메소드를 작성하는데에도 어려움이 있었다. HashMap은 key를 입력하여야 하는 것이 좀 헷갈렸다. 그래서 구글에서 찾아보다가 새로운 list를 생성하여 elements의 모든 값을 저장하고 idx번째 원소만 pe에 저장하는 코드를 작성하였다. list는 key값을 따로 입력하지 않아도 되는 점을 이용하여 Element를 다루지 않고 PeriodicElement만 다루어도 되도록 하였다.

PeriodicTableMap #50

PeriodicElement pe=(**new** ArrayList<PeriodicElement>(elements.values())).get(idx);













