Report

Lab 6

제출일: 5/21

과목명: 자바프로그래밍1

담당교수: 박경신 교수님

전공: 컴퓨터공학과

학번: 32191826

이름: 박주희

<추상클래스 vs 인터페이스>

|  |  |  |
| --- | --- | --- |
| 비교 | 목적 | 구성 |
| 추상클래스 | 추상클래스는 서브 클래스에서 필요로 하는 대부분의 기능을 구현하여 두고 서브 클래스가 상속받아 활용할 수 있도록 하되, 서브 클래스에서 구현할 수 밖에 없는 기능만을 추상 메소드로 선언하여, 서브 클래스에서 구현하도록 하는 목적 | 추상 메소드와 일반 메소드 모두 포함  상수, 변수 필드 모두 포함 |
| 인터페이스 | 인터페이스는 객체의 기능을 모두 공개한 표준화 문서와 같은 것으로, 개발자에게 인터페이스를 상속받는 클래스의 목적에 따라 인터페이스의 모든 추상 메소드를 만들도록 하는 목적 | 변수 필드는 포함하지 않음  상수 추상메소드, 일반 메소드, default 메소드, static 메소드 모두 포함  Protected 접근 지정 선언 불가  다중 상속 지원 |

추상메소드와 인터페이스 모두 객체를 생성할 수 없고 상속을 위한 슈퍼클래스로만 사용되며, 클래스의 다형성을 실현하기 위한 목적이다.

<Finder의 추상 클래스 vs Finder의 인터페이스>

PeriodicElementFinder 추상클래스:

state에 따라 다른 추상메소드 isInstanceOf를 호출하여 state에 상관없이 사용되는 해당 state에 따른 원소들을 반환하는 메소드인 find 메소드를 이용함으로서 state를 판별한다.

IperiodicElementFinder 인터페이스:

추상클래스와는 다르게 모두 추상메소드를 이용하여야 하므로 PeriodicElementFinder 클래스처럼 find 메소드를 구현할 수 없다.

PeriodicTable 클래스에서 대신 그 기능을 구현하였다.

<My code>

위에서 PeriodicElementFinder 추상클래스를 이용하여 gas 상태 원소들을 gasElements에, solid 상태 원소들을 solidElements, liquid 상태 원소들을 liquidElements에, unknown상태 원소들을 unknownElements에 저장하였다. 그것을 이용하여 gas상태인 원소들 중에서만 원자량이 50이상인 원소들과 solid 상태인 원소들 중에서만 원자량이 100 이상인 알 수 있다.

<Lab 6 구현 방법>

PeriodicTable의 find(IperiodicElementFinder ifinder)를 구현하는데 시간이 좀 걸렸다. 처음에 #48-#54까지 for-each문으로 작성하였더니 count변수는 제대로 저장이 되었으나 모두 null로 저장되어 있었다. 그러다가 for-each문을 이용하면

for(PeriodicElement i:pe)

if((i!=null)&&(ifinder.isInstanceOf(elements[i]==true)) 여기서 코드가 잘못되었음을 알게 되었다.

pe배열에는 원소들의 정보가 아직 저장되어있지 않고 null인데 i는 계속 null이므로 조건문과 성립하지 않아 pe에 원소들이 저장될 수 없었던 것이다.

인터페이스와 추상클래스를 어떤 상황에서 어떻게 이용해야 편리한지를 느낄 수 있었다. Lab 6는 state에 관계없이 state에 해당하는 원소를 세는 메소드가 필요했기에 인터페이스보다는 추상클래스가 더 편리했던 것 같다.





