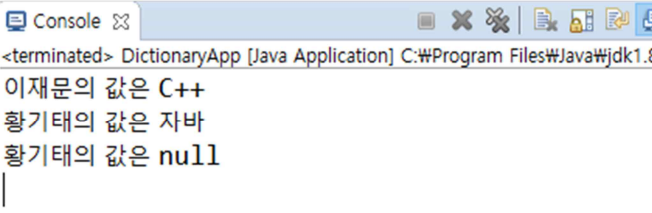


1. 다음은 키와 값을 하나의 아이템으로 저장하고 검색 수정이 가능한 추상 클래스가 있다.

```
abstract class PairMap {  
    protected String keyArray []; // key들을 저장하는 배열  
    protected String valueArray []; // value 들을 저장하는 배열  
    abstract String get(String key); // key 값으로 value를 검색  
    abstract void put(String key, String value); // key와 value를 쌍으로 저장  
    abstract String delete(String key); // key 값을 가진 아이템(value와 함께)을 삭제. 삭제된 value 값 리턴  
    abstract int length(); // 현재 저장된 아이템의 개수 리턴  
}
```

- PairMap을 상속받는 Dictionary 클래스를 구현하고, 이를 다음과 같이 활용하는 main() 메소드를 가진 클래스 DictionaryApp도 작성하라.

```
public static void main(String[] args) {  
    Dictionary dic = new Dictionary(10);  
    dic.put("황기태", "자바");  
    dic.put("이재문", "파이썬");  
    dic.put("이재문", "C++"); // 이재문의 값을 C++로 수정  
    System.out.println("이재문의 값은 " + dic.get("이재문"));  
    System.out.println("황기태의 값은 " + dic.get("황기태"));  
    dic.delete("황기태");  
    System.out.println("황기태의 값은 " + dic.get("황기태"));  
}
```

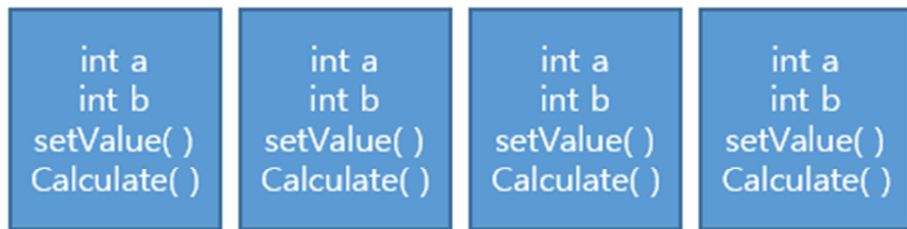


The screenshot shows a console window titled "Console" with the following output:

```
<terminated> DictionaryApp [Java Application] C:\Program Files\Java\jdk1.8  
이재문의 값은 C++  
황기태의 값은 자바  
황기태의 값은 null
```

2. 철수 학생은 다음 3개의 필드와 메소드를 가진 4개의 클래스 Add, Sub, Mul, Div를 작성하려고 한다.

- int 타입의 a, b 필드 : 2개의 피연산자
- void setValue(int a, int b) : 피연산자 값을 객체 내에 저장한다.
- int calculate () : 클래스의 목적에 맞는 연산을 실행하고 결과를 리턴한다.



Add

Sub

Mul

Div

곰곰히 생각해보니, Add, Sub, Mul, Div 클래스에 공통된 필드와 메소드가 존재하므로 새로운 추상 클래스 Calc를 작성하고 Calc를 상속받아 만들면 되겠다고 생각했다. 그리고 main() 메소드에서 다음 실행 사례와 같이 2개의 정수와 연산자를 입력받은 후, Add, Sub, Mul, Div 중에서 이 연산을 처리할 수 있는 객체를 생성하고 setValue()와 Calculate()를 호출하여 그 결과 값을 화면에 출력하면 된다고 생각하였다. 철수처럼 프로그래밍 하라.