

자료구조 실습 과제 #1 (20171622 박건후)

1. Javadoc

1.1 Javadoc은 무엇이고 왜 사용하는가?

Javadoc은 **Java 코드에서 API 문서를 HTML 형식으로 생성해주는 도구**입니다. HTML 형식이어서 관련 다른 API를 하이퍼 링크를 통해 접근이 가능하다는 장점이 있습니다. 여러 개발자들이 사용하는 라이브러리를 만든다 가정할 때, 자신을 제외한 다른 개발자는 라이브러리의 구성, 클래스의 구성이 어떻게 되는지 모릅니다. 따라서 **다른 개발자들이 자신이 만든 라이브러리를 잘 사용할 수 있도록 API 문서를 제작하는데, 그 문서를 html 파일 형식으로 만들어주는 것이 Javadoc**입니다.

1.2 Javadoc은 어떻게 사용하는가?

```
/**
 * 연습용 클래스 입니다
 * @author parkgeonhu
 */
public class Introduce {
    /**
     * 이름 출력용 메소드 입니다.
     * @param String str 문자
     * @param int x 기본 변수
     * @return String re OK or Failed
     */
    public static void display() {
        System.out.println("parkgeonhu");
    }
}
```

- @author 개발자
- @exception 메소드에서의 예외 확인
- @param 메소드의 매개변수
- @return 메소드의 반환값
- @see 다른 주제에 관한 링크 지정
- @serial 직렬화 필드
- @since 릴리즈 기록
- @throws 메소드에서의 예외
- @version 클래스의 버전

위의 소스 코드를 "Sample14.java"라는 파일명으로 저장하고 저장된 디렉토리에서 다음과 같이 실행합니다.

```
$ javadoc -d doc Introduce.java
```

Class Introduce

java.lang.Object
Introduce

```
public class Introduce
extends java.lang.Object
```

연습용 클래스입니다

Constructor Summary

Constructors

Constructor and Description

Introduce()

Method Summary

Methods

Modifier and Type	Method and Description
static void	display() 이름 출력용 메소드 입니다.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Introduce

```
public Introduce()
```

Method Detail

display

```
public static void display()
```

이름 출력용 메소드 입니다.

Parameters:

String - str 문자

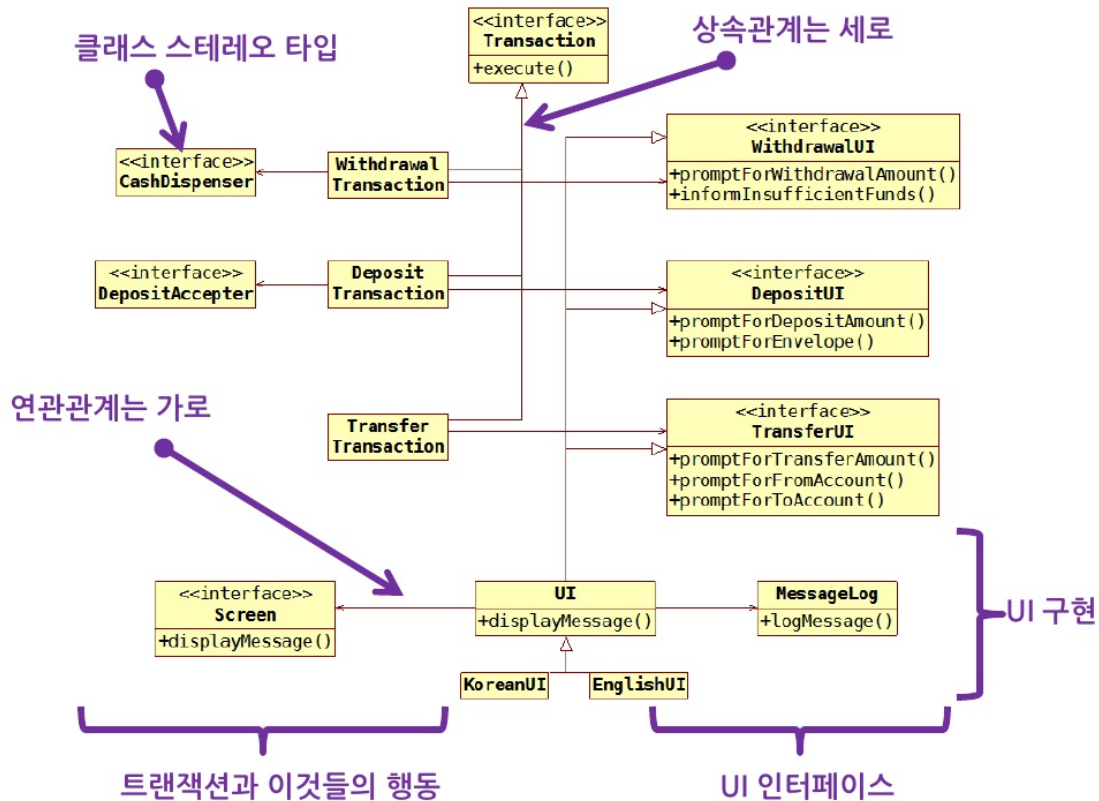
int - x 기본 변수

index.html 을 열어보면 위와 같이 나타납니다. Hierarchy For Package를 확인하여 어떤 구조로 되어있는지 파악이 가능하며, Deprecated API를 통해 만료된(더이상 사용하지 않는) 메소드나 클래스를 알아낼 수 있습니다.

2. UML

2.1 UML은 무엇이고 왜 사용하는가?

UML(Unified Modeling Language)이란 모델을 다이어그램으로 그리기 위해 사용하는 시각적인 표기법입니다. 모델을 만들어야 하는 이유는 실제 물건을 만드는 비용이 모델로 설계를 검사하는 비용보다 높기 때문입니다. 소프트웨어 공학적으로는, 실제 구현을 하는 비용이 모델로 설계를 검사하는 비용보다 높은 것을 의미합니다. 코드를 읽는 것보다 클래스 의존관계를 빨리 파악할 수 있기 때문에 UML을 사용합니다.



왜 소프트웨어 모델을 만드는가? UML 다이어그램을 그리는 일은 소프트웨어를 작성하는 일보다 비용이 적긴 하지만, 다른 분야(항공, 건축 등)의 모델처럼 훨씬 적게 드는 것은 아닙니다. 시험해 볼 구체적인 것이 있고, 그것을 코드로 시험해 보는 것보다 UML로 시험해 보는 쪽이 비용이 덜 들 때 UML을 사용합니다.

반드시 코딩을 시작하기에 앞서 포괄적인 설계를 해야 하는가? 계획 없이 어떤 빌딩을 짓는 것보다 미리 계획을 짜는 것이 비용이 '훨씬 적게' 듭니다. 잘못된 청사진을 던져 버리는 일에는 비용이 별로 들지 않지만, 잘못된 빌딩을 부수려면 비용이 '엄청나게' 듭니다. 모델의 경우와 마찬가지로 다른 분야에 비해 소프트웨어 분야에서는 모든것이 이렇게 분명하지 않습니다. 코드를 작성하는 것보다 UML 다이어그램을 그리는 것이 훨씬 비용이 적은지는 명확하지 않습니다. 그러므로 코드를 작성하기에 앞서 포괄적인 UML 설계를 만들면 드는 비용만큼 효과가 있는지 명확하게 알 수 없습니다.

2.2 UML은 어떻게 사용하는가?

사용법

다른 사람들과 의사 소통하기 UML은 소프트웨어 개발자끼리 설계 개념에 대한 의견을 주고 받을 때 굉장히 편리하며, 다른 개발자가 자신이 설계한 도면을 알아볼 때 중요합니다.

로드맵 어떤 클래스가 다른 클래스에 의존하는지 개발자가 빨리 파악할 수 있게 해주고 전체 시스템의 구조에 대한 참조 도표로도 사용됩니다.

객체 다이어그램

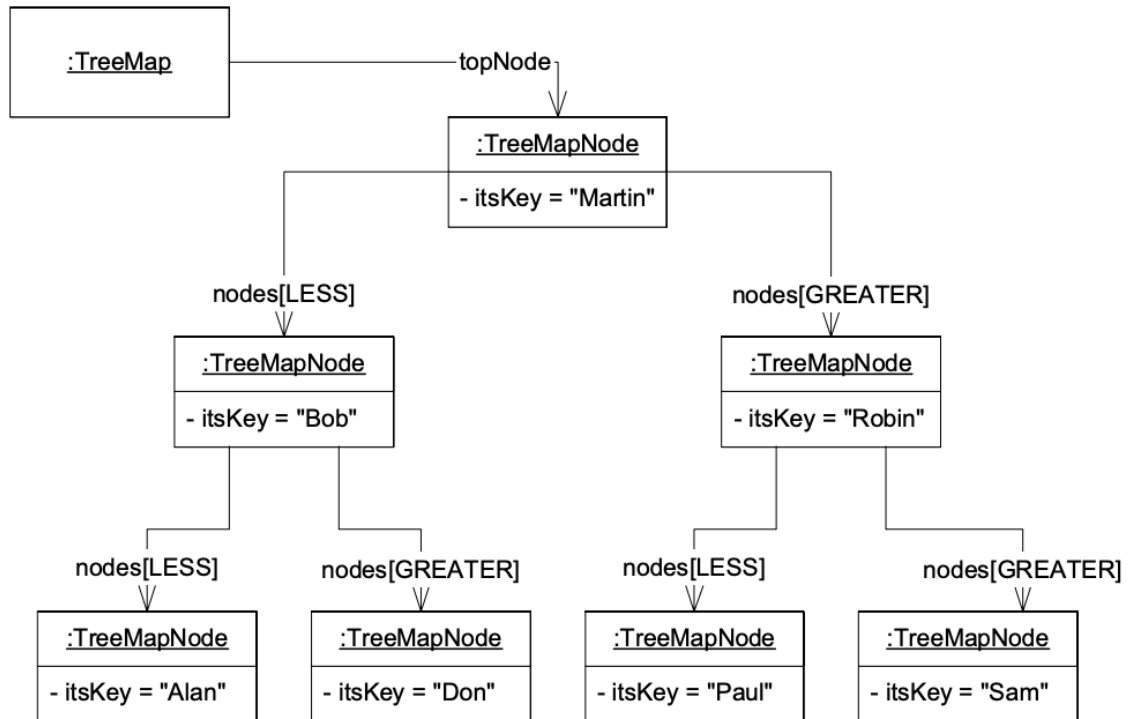


Figure 1-3
TreeMap Object Diagram

- 객체는 사각형과 밑줄로 표현합니다.
- 이 객체가 속하는 클래스의 이름은 콜론(:) 다음에 표현합니다.
- 객체마다 아래 구획에 그 객체의 itsKey 변수의 값이 나와 있습니다.
- 객체 사이의 관계는 연결(link)이라고 합니다.

다이어그램을 그려야 할 경우

- 모두 설계에서 특정한 부분의 구조를 이해해야 할 때 그려야 합니다.
- 두 명 이상이 특정 요소를 어떻게 설계할지 의견을 모을 필요가 있을때 그려야 합니다.
- 누군가에게 코드 일부분의 구조를 설명할 때 그려야 합니다.
- 프로젝트 마지막에 고객이나 다른 사람을 위한 문서에 포함하기 위해 다이어그램을 요구할 때 그려야 합니다.