(1)

ConcurrentModificationException 가 일어나는 원인은 무엇이고 어떻게 해결하면 좋을지 기술하라

 ConcurrentModificationException가 일어나는 원인은 반복 루프의 횟수에 영향을 끼치는 값이 루프 안에서 변하게 될 때 발생하게 된다. 이를 해결하기 위해서는 루프를 돌면서 지워야될 값을 저장하고 루프가 종료되면 한 번에 지운다. 두 번째 방법은 java의 removeif를 활용하여 지우고자 하는 엘리먼트를 지운다.

(2)

다음 클래스의 특징 및 차이점, 사용법 등을 기술하라

-ObjectAnimator : view의 translationY를 증가시킨다. 뷰에서 간단한 애니메이션을 활용할 때 사용한다.

Repeat를 사용해서 반복 애니메이션이 가능하다.

propertyName에 이름을 설정한다. valueType에는 자료형을 설정한다. valueFrom에는 시작 값을 설정한다. valueTo에는 끝 값을 설정한다. repeatMode에는 반복 설정을 한다. repeatCount에는 반복 횟수를 설정한다. duration에는 동작의 시간을 설정한다.

-ValueAnimator : View의 height를 증가시킨다. 오직 하나의 뷰에서 애니메이션을 쓰고자 할 때 사용한다. 반복 애니메이션이 불가능하다.

duration에는 동작 시간을 설정한다. ofFloat은 변화 범위를 설정한다. start에는 애니메이션의 시작을 설정한다.

(3)

2D Graphics 와 3x3 행렬은 어떤 관계에 있는지, 2차원 변환행렬의 종류에 대해 알아보고 정리하라

<https://m.blog.naver.com/PostView.nhn?blogId=kimjw1218&logNo=70178444277&proxyReferer=https:%2F%2Fwww.google.com%2F>

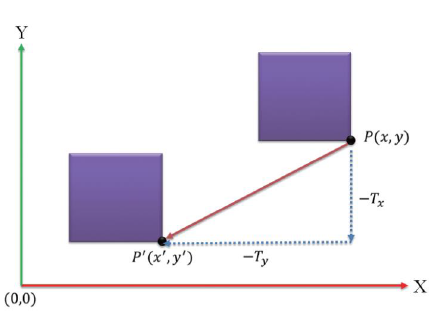
등을 참고해도 좋다

이동, 신축, 회전의 연속된 변환을 동차 좌표계를 사용하여 하나의 행렬로 나타낼 수 있다.

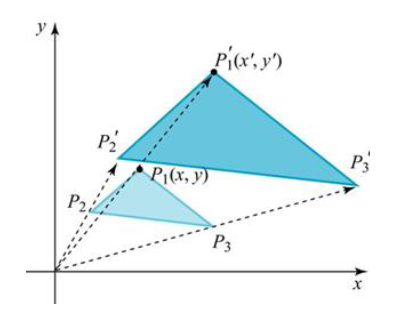
하나의 변환 행렬로 표현한 합성 변환에서는 한 번의 행렬 곱셈만을 필요로 한다.

Translation(이동)변환은 좌표계의 한 곳에서 다른 곳으로 직선 경로를 따라 객체의 위치를 바꾸는 것이다.

객체의 크기나 모양 등은 바뀌지 않는다.



Scalint(신축)변환은 객체의 크기를 확대/축소 시킨다. 객체의 크기뿐 아니라 기준점으로부터의 위치도 배율에 따라 변한다.



Rotation(회전)변환은 xy평면에서 원 경로를 따라 객체를 재배치한다. 객체의 모양 변화는 없이 객체가 놓여있는 방향이 변한다.

