(1) ConcurrentModificationException 가 일어나는 원인은 무엇이고 어떻게 해결하면 좋을지 기술하라

원인: 반복문의 반복 횟수에 관여하는 값이 반복문 내에서 변할 때 발생한다.

해결방법:

1. 나중에 하기

반복문을 돌면서 미리 지워야 할 값들을 저장해둔 뒤 반복문이 끝난 후 한꺼번에 지운다.

2. removeIf와 lambda expression을 이용

java에서 제공하는 removeIf 메서드를 이용해서 지우고 싶은 엘리먼트를 지운다.

(2) 다음 클래스의 특징 및 차이점, 사용법 등을 기술하라

-ObjectAnimator

View의 tanslationY 값을 증가시키는 애니메이션이다.

특징: 하나의 뷰에서 간단한 애니메이션을 이용하고자 하지만 반복적으로 동작하기를 바랄 때 사용

장점: Repeat를 사용해 애니메이션을 반복적으로 사용할 수 있다.

XML을 이용하여 애니메이션을 만들 수 있다.

단점: 순차적인 애니메이션 혹은 복합적인 애니메이션이 불가능 하다.

사용법:

propertyName - 이름 설정

valueType - 값 자료형 설정

valueFrom – 시작 값 설정

valueTo – 끝 값 설정

repeatMode – 반복 방식 설정

repeatCount – 반복 횟수 설정

duration – 동작 시간 설정

-ValueAnimator

View의 높이(height)를 증가시키는 애니메이션이다.

특징: 간단한 애니메이션에서 사용, 오직 하나의 뷰에서만 애니메이션이 동작하길 원할 때 사용

장점: 객체를 직접 Animator obj로 인스턴스화 할 필요가 없으므로 간결한 애니메이션을 할 때 편리하다.

단점: 반복적인 Animation을 만들 수 없다.

end value만 설정할 수 있고 start value는 설정할 수 없다.

순차적인 애니메이션 혹은 복합적인 애니메이션이 불가능 하다.

사용법:

duration – 동작 시간 설정

ofFloat – 변화 범위 설정

start – 애니메이션 시작

두 클래스의 차이점:

ObjectAnimator는 View의 tanslationY 값을 ValueAnimator는 View의 높이(height)를 증가시킨다.

ObjectAnimator는 반복적으로 동작할 수 있지만 ValueAnimator는 그럴 수 없다.

ValueAnimator와 다르게 ObjectAnimator는 XML을 이용하여 애니메이션을 만들 수 있다.

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

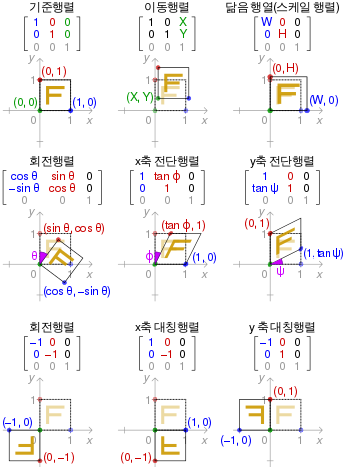
2D Graphics와 3x3 행렬의 관계

기존의 행렬을 사용하지 않고 좌표를 변환했을 때는 같은 값으로 변화를 주더라도 이동하고 회전을 하느냐 회전하고 이동을 하느냐 등 변화의 순서에 따라 결과 값이 달랐다.

이는 연산이 많아질수록 계산양이 크게 증가하는 단점도 존재했는데 이것을 해결하기 위해 동차 좌표계를 이용하는 방법이 나왔다.

동차 좌표계는 2차원 기준으로 x, y 외에 1을 더 추가해 (x, y, 1) 형태로 나타낸다. 이때 1을 추가로 더한다고 해서 값이 변하지 않기 때문에 사용이 가능하며 좌표가 3개가 되었기 때문에 3x3 행렬과의 곱이 가능해져서 이동변환, 신축변환, 회전변환을 3x3행렬에 모두 담을 수 있게 됐다.

변환행렬의 종류



이미지 출처: <https://ko.wikipedia.org/wiki/%EB%B3%80%ED%99%98%ED%96%89%EB%A0%AC#%EC%A2%85%EB%A5%98>