AnimationImpl
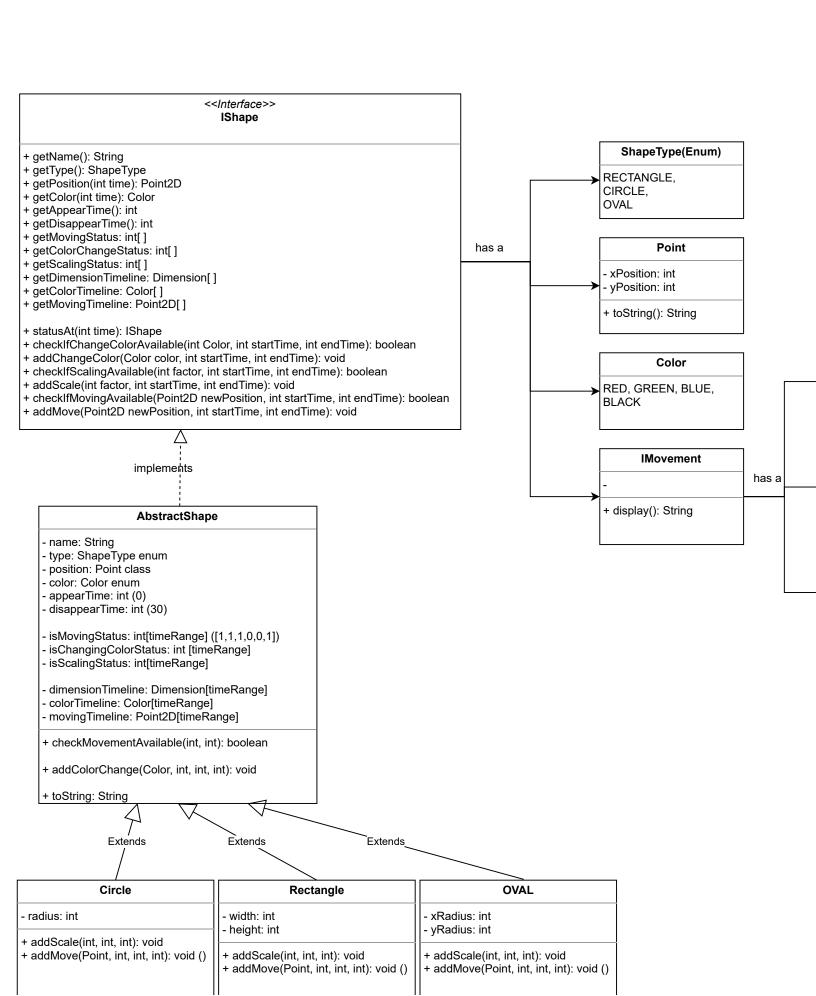
- listOfShapes: LinkedList<IShape>
- statusAtTime: LinkedList<Shape>
-

+ addShape(IShape)

+ changeColor(IShape, Color, int, int): void
+ scale(IShape, int, int, int): void
+ move(IShape, Point2D, int, int): void ()

+ getListOfShapes(): LinkedList<IShape>
+ getAnimations(): LinkedList<Movement>
+ getShapeStatusTimeline(): LinkedList<Shape>[ ]
+ getStatusAtTime(int): LinkedList<Shape>

---- implements ----▷

<<Interface>>
IAnimation

+ addShape

+ changeColor(IShape, Color, int, int): void
+ scale(IShape, int, int, int): void
+ move(IShape, Point2D, int, int): void ()

+ getAnimationTimeline()
+ getStatusTimeline()
+ getStatusAtTime(int)

<<Interface>>
IShape

+ getName(): String
+ getType(): ShapeType
+ getPosition(int time): Point2D
+ getColor(int time): Color
+ getAppearTime(): int
+ getDisappearTime(): int
+ getMovingStatus: int[ ]
+ getColorChangeStatus: int[ ]
+ getScalingStatus: int[ ]
+ getDimensionTimeline: Dimension[ ]
+ getColorTimeline: Color[ ]
+ getMovingTimeline: Point2D[ ]

+ statusAt(int time): IShape
+ checkIfChangeColorAvailable(int Color, int startTime, int endTime): boolean
+ addChangeColor(Color color, int startTime, int endTime): void
+ checkIfScalingAvailable(int factor, int startTime, int endTime): boolean
+ addScale(int factor, int startTime, int endTime): void
+ checkIfMovingAvailable(Point2D newPosition, int startTime, int endTime): boolean
+ addMove(Point2D newPosition, int startTime, int endTime): void

ShapeType(Enum)

RECTANGLE,
CIRCLE,
OVAL

Point

- xPosition: int
- yPosition: int

+ toString(): String

Color

RED, GREEN, BLUE,
BLACK

IMovement

-

+ display(): String

has a

has a

implements

AbstractShape

- name: String
- type: ShapeType enum
- position: Point class
- color: Color enum
- appearTime: int (0)
- disappearTime: int (30)

- isMovingStatus: int[timeRange] ([1,1,1,0,0,1])
- isChangingColorStatus: int [timeRange]
- isScalingStatus: int[timeRange]

- dimensionTimeline: Dimension[timeRange]
- colorTimeline: Color[timeRange]
- movingTimeline: Point2D[timeRange]

+ checkMovementAvailable(int, int): boolean

+ addColorChange(Color, int, int, int): void

+ toString: String

Extends   Extends   Extends

Circle

- radius: int

+ addScale(int, int, int): void
+ addMove(Point, int, int, int): void ()

Rectangle

- width: int
- height: int

+ addScale(int, int, int): void
+ addMove(Point, int, int, int): void ()

OVAL

- xRadius: int
- yRadius: int

+ addScale(int, int, int): void
+ addMove(Point, int, int, int): void ()

Move

- oldPosition: point2D
- newPosition: point2D

+ display(): String

ColorChange

- oldColor: Color
- newPosition: point2D

+ display(): String

Scale

- factor: int

+ display(): String

timeline

t=10   shapes appears   C1   O1

t=20   shapes appears   R1

t = 30   C1 scales starts

t = 34   R1 changes color start

R1   C1

t = 50   R1 changes color done

t= 55   C1 scales done

t = 58   C1 disappears

t = 60   O1 move starts   O1   O1

t = 65   O1 move ends

t = 80   O1 disappears; R1 disappears

Timeline

- listOfShapes: LinkedList<IShape>
- movementOnTimeline: Movement[ ][ ]
- statusOnTime: Shape[ ][ ]

+ changeColor(Shape, Color, int, int): void
+ scale(Shape, int, int, int): void
+ addMove(Shape, Point, int, int): void ()
+ getAnimationTimeline()
+ getStatusTimeline()
+ getStatusAtTime(int)