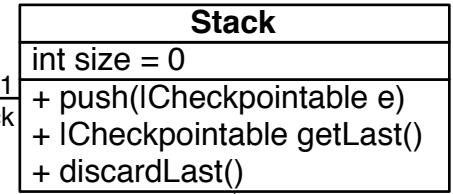
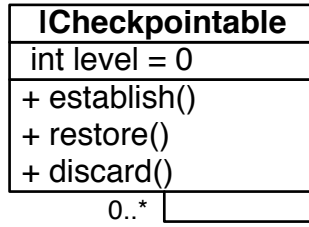


aspect Checkpointable depends on Copyable

structural view depends on Copyable

ICheckpointable

Copyable instantiation
Copyable.ICopyable → ICheckpointable



state view ICheckpointable depends on Copyable

Advice

Binding
Any → *

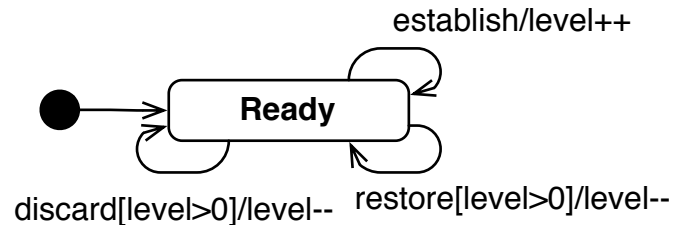
Pointcut

Any

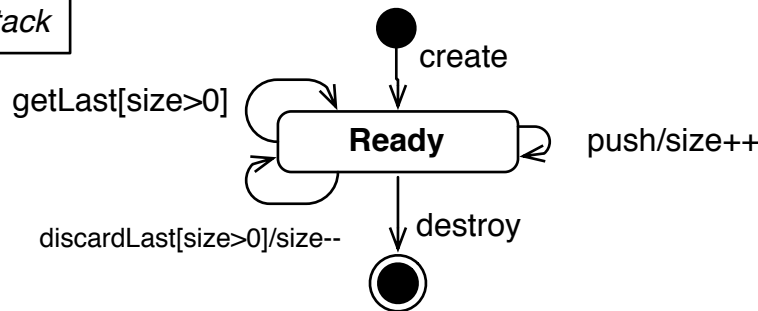
Copyable.ICopyable instantiation
ICopyable.ICloneAllowed
→ Established
ICopyable.IRemoveAllowed
→ Ready

Any

CheckpointingAllowed



state view Stack



message view establish depends on Copyable

Pointcut

caller: Caller

target: ICheckpointable

establish()

Copyable.clone instantiation

clone.original → target (1)
clone.lcopy → newCheckpoint (2)
clone.Caller → ICheckpointable (3)
clone.caller → target (4)

Advice

caller: Caller

target: ICheckpointable

myStack: Stack

establish()

newCheckpoint := clone()
push(newCheckpoint)

Binding

caller → *
Caller → *
target → *

message view restore

Pointcut

lcaller: ICaller

ltarget: ICheckpointable

restore()

Binding

caller → *
Caller → *
target → *

Advice

lcaller: ICaller

ltarget: ICheckpointable

myStack: Stack

restore()

last := getLast()
replaceStateWith(last)

message view discard

Pointcut

lcaller: ICaller

ltarget: ICheckpointable

discard()

Binding

caller → *
Caller → *
target → *

Advice

lcaller: ICaller

ltarget: ICheckpointable

myStack: Stack

discard()

discardLast()