



<<library>>

SafeMath

#mul(a:uint,b:uint): uint const  
#div(a:uint,b:uint): uint const  
#sub(a:uint,b:uint): uint const  
#add(a:uint,b:uint): uint const

Pausable

+paused: bool = false  
<<event>> Pause()  
<<event>> Unpause()  
<<modifier>> whenNotPaused()  
<<modifier>> whenPaused()  
+pause()  
  onlyOwner  
  whenNotPaused  
+unpause()  
  onlyOwner  
  whenPaused

BasicToken

balances: {address->uint}  
totalSupply: uint  
totalSupply(): uint const  
transfer(to:address, value:uint): bool  
balanceOf(\_owner:address): uint const

BurnableToken

<<event>> Burn(burner:address, value:uint)  
+burn(\_value:uint)  
# burn(\_who:address, value:uint)

PausableToken

\_value:uint): bool  
idress, to:address,  
int): bool  
ress, \_value:uint): bool  
nder:address, \_addedValue:uint): bool  
nder:address, \_subtractedValue:uint): bool

0000000e18

"