

## Class **Account**

```
-- Name Ann Onymous
-- Does provide a simple savings bank account
-- Thin version for SIGCSE (uses short names)
-- Shows PbC, Programming by Contract in JJ

-- Data attributes
Box bal ofType real is private
Box idno ofClass Str is public

Constructor Account (b, i) is public
  Slot b ofType real -- balance
  Slot i ofClass Str -- identification
-- Does initialize or open the account
  Check (bal >= 0.00) bounce "Neg balance!"
  Set bal = b
  Set idno = i
EndConstructor Account

Routine setBal (amount) is public
  Slot amount ofType real
-- Does set or reset the amount of bal
  Set bal = amount
EndRoutine setBal

Function balance(none) ofType real is public
  Box result ofType real
-- Does return balance in account
  Set result = bal
EndFunction balance

Routine credit (amt) is public
  Slot amt ofType real
-- Does deposit an amount into account
  Check (amt >= 0.00) bounce "Neg deposit!"
  Set bal = bal + amt
EndRoutine credit

Routine debit (amt) is public
  Slot amt ofType real
-- Does withdraw an amount from account
  Check (amt >= 0.00) bounce "Neg debit!"
  Check (amt < bal) bounce "Insufficient!"
  Set bal = bal - amt
EndRoutine debit

Routine show (none) is public
-- Does display the account
  Output "Account " + idno + " has balance "
  Outputln bal
EndRoutine show

Routine compound (prct, durn) is public
  Slot prct ofType real -- percent rate
  Slot durn ofType int -- duration
  Box rate ofType real -- interest rate
-- Does compound balance at a rate for a time
  Set rate = prct / 100.0
  Repeat
  ExitOn (durn == 0)
    Set bal = bal + bal * rate
    Dec durn by 1
  EndRepeat
EndRoutine compound

Routine getFrom (acct, amt) is public
  Slot acct ofClass Account
  Slot amt ofType real
-- Does transfer from one account to this
  Check (acct.bal >= amt) bounce "Insuff!"
  Call acct.debit with (amt)
  Call credit with (amt)
EndRoutine getFrom

Function canCover(amt) ofType bool is public
  Slot amt ofType real
  Box result ofType bool
-- Does check if balance covers amount
  Set result = (bal >= amt)
EndFunction canCover

Routine test (none) is private
  Box his ofClass Account
  Box her ofClass Account
-- Does test Account class
  Start
    New his ofClass Account with (20.0, "ab3")
    New her ofClass Account with ( 0.0, "7yz")

    Call his.debit with (100.00)
    Call his.compound with (10.0, 7)
    Call his.show

    Call her.setBal with (200.00)
    If his.canCover (100.00) then
      Call her.getFrom with (his, 100.00)
    Else
      Outputln "No transfer possible"
    EndIf
    Output "Her balance is "
    Outputln her.balance()
  EndRoutine test

EndClass Account
```