

**Screenshots Portfolio Project Module 8: ATM Operations**

Alexander Ricciardi

Colorado State University Global

CSC505: Principles of Programming

Professor: Dr. Joseph Issa

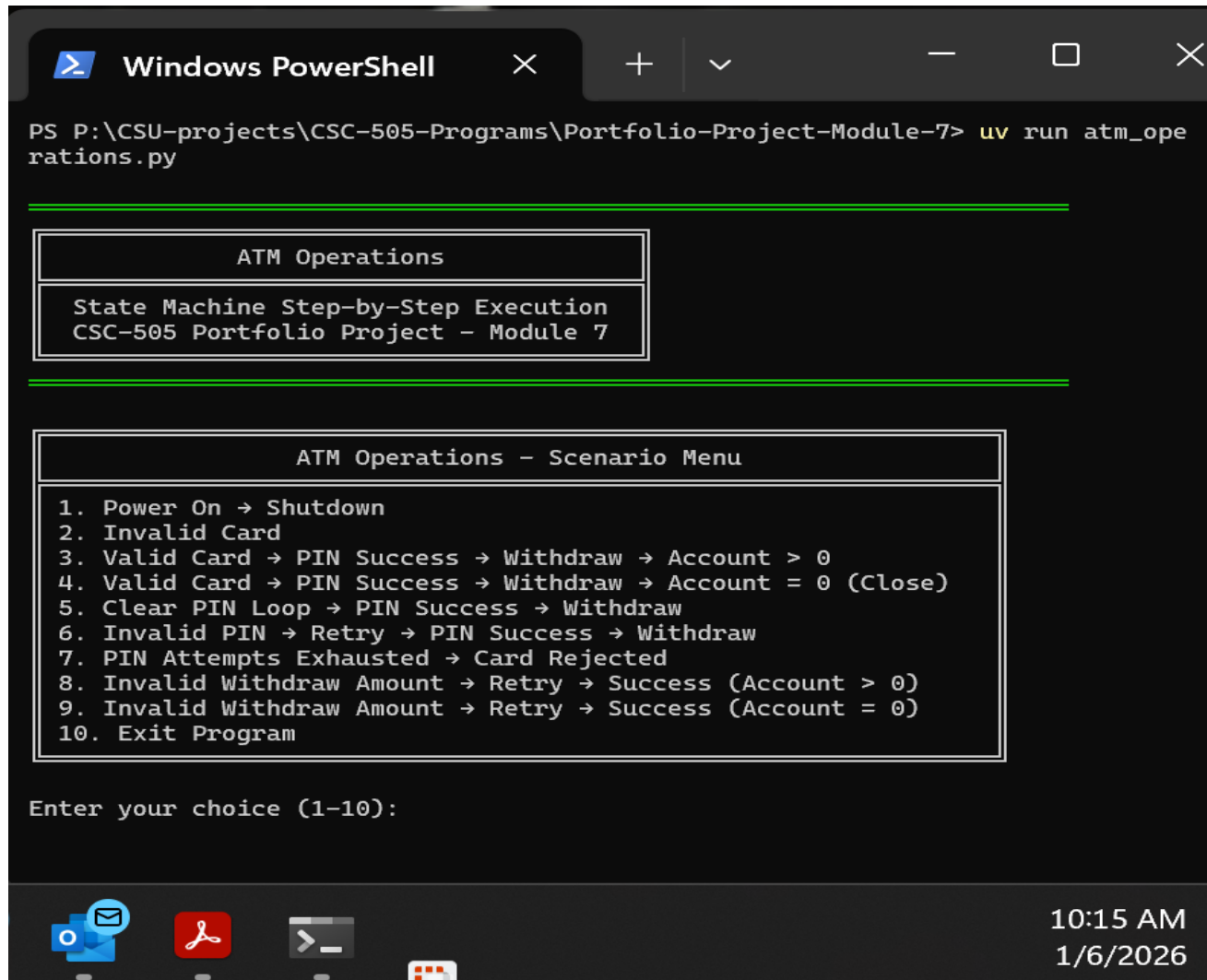
December 11, 2025

### Screenshots Portfolio Project Module 8: ATM Operations

This document contains a screenshot of the ATM Operations script user interactions and outputs

**Figure 1**

Main menu



```
Windows PowerShell
PS P:\CSU-projects\CSC-505-Programs\Portfolio-Project-Module-7> uv run atm_operations.py
```

ATM Operations

State Machine Step-by-Step Execution  
CSC-505 Portfolio Project - Module 7

ATM Operations - Scenario Menu

1. Power On → Shutdown
2. Invalid Card
3. Valid Card → PIN Success → Withdraw → Account > 0
4. Valid Card → PIN Success → Withdraw → Account = 0 (Close)
5. Clear PIN Loop → PIN Success → Withdraw
6. Invalid PIN → Retry → PIN Success → Withdraw
7. PIN Attempts Exhausted → Card Rejected
8. Invalid Withdraw Amount → Retry → Success (Account > 0)
9. Invalid Withdraw Amount → Retry → Success (Account = 0)
10. Exit Program

Enter your choice (1-10):

10:15 AM  
1/6/2026

See next page.

Figure 2

Scenario 1 Power On → Shutdown

```

PS P:\CSU-projects\CSC-505-Programs\Portfolio-Project-Module-7> uv run atm_operations.py

ATM Operations

State Machine Step-by-Step Execution
CSC-505 Portfolio Project - Module 7

ATM Operations - Scenario Menu

1. Power On → Shutdown
2. Invalid Card
3. Valid Card → PIN Success → Withdraw → Account > 0
4. Valid Card → PIN Success → Withdraw → Account = 0 (Close)
5. Clear PIN Loop → PIN Success → Withdraw
6. Invalid PIN → Retry → PIN Success → Withdraw
7. PIN Attempts Exhausted → Card Rejected
8. Invalid Withdraw Amount → Retry → Success (Account > 0)
9. Invalid Withdraw Amount → Retry → Success (Account = 0)
10. Exit Program

Enter your choice (1-10): 1

SCENARIO 1

Scenario: Power On → Shutdown

ATM powers on, displays welcome, then shuts down

Initial Context: account=$100, attempt=0

01 START [*] Initial State
   State machine begins execution from the initial pseudo-state.
02 TRANS PowerOn / showWelcome
   System event: ATM powers on and initializes the welcome display.
03 STATE Idle: entry / showingWelcome
   ATM enters Idle state; system displays welcome screen to attract users.
04 STATE Idle: exit / readCard or shutdown
   User action required: insert card to begin session, or operator shuts down system.
05 TRANS Shutdown / poweringOff
   Operator event: ATM is being shut down for maintenance or end of day.
06 END [*] Final State - System shutdown
   State machine reaches final state. Session outcome: System shutdown.

Scenario Complete

Result: System shutdown
Final Context: account=$100, attempt=0

Press Enter to continue...
  
```

Figure 3

Scenario 2 Invalid Card

Windows PowerShell

ATM Operations - Scenario Menu

1. Power On → Shutdown
2. Invalid Card
3. Valid Card → PIN Success → Withdraw → Account > 0
4. Valid Card → PIN Success → Withdraw → Account = 0 (Close)
5. Clear PIN Loop → PIN Success → Withdraw
6. Invalid PIN → Retry → PIN Success → Withdraw
7. PIN Attempts Exhausted → Card Rejected
8. Invalid Withdraw Amount → Retry → Success (Account > 0)
9. Invalid Withdraw Amount → Retry → Success (Account = 0)
10. Exit Program

Enter your choice (1-10): 2

---

SCENARIO 2

---

Scenario: Invalid Card

User inserts invalid card → rejected → removes card → session ends

Initial Context: account=\$100, attempt=0

```

01 START [*] Initial State
    State machine begins execution from the initial pseudo-state.
02 TRANS PowerOn / showWelcome
    System event: ATM powers on and initializes the welcome display.
03 STATE Idle: entry / showingWelcome
    ATM enters Idle state; system displays welcome screen to attract users.
04 STATE Idle: exit / readCard or shutdown
    User action required: insert card to begin session, or operator shuts down system.
05 TRANS InsertedCard(validCard) / checkCard
    User action: customer inserts their ATM card into the reader.
06 CHOICE CardCheck: [isValidCard=false] → InvalidCard
    Guard evaluates TRUE: card is invalid. Route to rejection handling.
07 STATE Invalid Card: entry / showingError
    ATM enters Invalid Card state; system shows error message to user.
08 STATE Invalid Card: exit / userAcknowledge && userRemoveCard
    User action required: acknowledge error and remove the rejected card.
09 TRANS UserRemovedCard / endSession
    User removes rejected card; system returns to idle and ends the session.
10 STATE Idle: entry / showingWelcome
    ATM enters Idle state; system displays welcome screen to attract users.
11 STATE Idle: exit / readCard or shutdown
    User action required: insert card to begin session, or operator shuts down system.
12 END [*] Final State - Session ended - invalid card
    State machine reaches final state. Session outcome: Session ended - invalid card.
  
```

Scenario Complete

Result: Session ended - card rejected

Final Context: account=\$100, attempt=0

Press Enter to continue...

10:31 AM  
1/6/2026

Figure 4

Scenario 3 Valid Card → PIN Success → Withdraw → Account &gt; 0

```

Windows PowerShell

ATM Operations - Scenario Menu
1. Power On → Shutdown
2. Invalid Card
3. Valid Card → PIN Success → Withdraw → Account > 0
4. Valid Card → PIN Success → Withdraw → Account = 0 (Close)
5. Clear PIN Loop → PIN Success → Withdraw
6. Invalid PIN → Retry → PIN Success → Withdraw
7. PIN Attempts Exhausted → Card Rejected
8. Invalid Withdraw Amount → Retry → Success (Account > 0)
9. Invalid Withdraw Amount → Retry → Success (Account = 0)
10. Exit Program

Enter your choice (1-10): 3

SCENARIO 3

Scenario: Valid Card → PIN Success → Account > 0
Full successful transaction with remaining balance
Initial Context: account=$100, attempt=0

01 START [*] Initial State
   State machine begins execution from the initial pseudo-state.
02 TRANS PowerOn / showWelcome
   System event: ATM powers on and initializes the welcome display.
03 STATE Idle: entry / showingWelcome
   ATM enters Idle state; system displays welcome screen to attract users.
04 STATE Idle: exit / readCard or shutdown
   User action required: insert card to begin session, or operator shuts down system.
05 TRANS InsertedCard(validCard) / checkCard
   User action: customer inserts their ATM card into the reader.
06 CHOICE CardCheck: [IsValidCard=true && attempt==0] → PromptPin
   Guard evaluates TRUE: card is valid and no prior attempts. Proceed to PIN entry.
07 STATE Prompt User For Pin: entry / promptingForPin
   ATM enters PIN prompt state; system waits for user to enter their PIN.
08 STATE Prompt User For Pin: exit / pinEntered
   User action required: submit PIN entry to proceed with authentication.
09 TRANS PinEntered(pin, attempt) / incrementAttempt
   User submits PIN; system routes to increment the attempt counter.
10 STATE Increment Attempt: entry / incrementAttempt
   ATM enters Increment Attempt state; system begins tracking this PIN try.
11 STATE Increment Attempt: do / attempt = attempt + 1 (attempt: 0 → 1)
   System increments attempt counter; 0 → 1.
12 STATE Increment Attempt: exit / attemptIncrementedSuccessfully
   System completed: attempt counter updated successfully.
13 TRANS PinEntered(pin, attempt) / authenticating
   System forwards PIN and attempt count to authentication module.
14 STATE Authenticating: entry / authenticatingPin
   ATM enters Authenticating state; system verifies PIN against bank records.
15 STATE Authenticating: exit / pinValidation
   System completed: PIN validation result determined (valid or invalid).
16 TRANS PinValidation(IsValidPin, attempt) / check
   System event: PIN validation result is evaluated.
17 CHOICE PinCheck: [IsValidPin=true] → PromptAmt
   Guard evaluates TRUE: PIN is correct. Proceed to withdrawal menu.
18 STATE Prompt User For Amount to Withdraw: entry / promptingForWithdrawAmount
   ATM enters Amount Prompt state; system waits for user to enter withdrawal amount.
19 STATE Prompt User For Amount to Withdraw: exit / userEnteredAmount
   User action required: enter the desired withdrawal amount.
20 TRANS Withdraw($50) / computeWithdraw
   User requests $50 withdrawal; system begins balance computation.
21 STATE Compute Withdraw Amount: entry / processingWithdraw
   ATM enters Compute Amount state; system calculates new balance.
22 STATE Compute Withdraw Amount: do / computedAmount = accountAmount - withdrawAmount (100 - 50 = 50)
   System computes: $100 - $50 = $50.
23 STATE Compute Withdraw Amount: exit / withdrawAmountComputed
   System completed: withdrawal amount computation finished.
24 TRANS ProcessAmount(computedAmount) / check
   System event: computed balance is checked for validity.
25 CHOICE AmtCheck: [computedAmount >= 0] → Dispense
   Guard evaluates TRUE: sufficient funds available. Proceed to dispense cash.
26 STATE Dispense Cash: entry / dispensingCash
   ATM enters Dispense Cash state; system prepares to release cash.
27 STATE Dispense Cash: do / accountAmount = computedAmount (accountAmount + 50)
   System updates account balance to $50 after dispensing cash.
28 STATE Dispense Cash: exit / cashDispensed
   System completed: cash has been dispensed to user.
29 TRANS AccountStatus(accountAmount) / check
   System event: account balance is checked after withdrawal.
30 CHOICE AcctCheck: [accountAmount > 0] → ShowNewAmt
   Guard evaluates TRUE: balance remains positive. Show new balance.
31 STATE Show Account New Amount: entry / showingAccount
   ATM enters Show Balance state; system displays updated account balance to user.
32 STATE Show Account New Amount: exit / userAcknowledge
   User action required: acknowledge the new balance display.
33 TRANS UserRemovedCard / endSession
   User removes card after viewing balance; session ends successfully.
34 STATE Idle: entry / showingWelcome
   ATM enters Idle state; system displays welcome screen to attract users.
35 STATE Idle: exit / readCard or shutdown
   User action required: insert card to begin session, or operator shuts down system.
36 END [*] Final State - Transaction complete - balance: $50
   State machine reaches final state. Session outcome: Transaction complete - balance: $50.

Scenario Complete
Result: Transaction successful - remaining balance: $50
Final Context: account=$50, attempt=1

Press Enter to continue...
10:33 AM
1/6/2026

```

Figure 5

Scenario 4 Valid Card → PIN Success → Withdraw → Account = 0 (Close)

```

ATM Operations - Scenario Menu
1. Power On → Shutdown
2. Invalid Card
3. Valid Card → PIN Success → Withdraw → Account > 0
4. Valid Card → PIN Success → Withdraw → Account = 0 (Close)
5. Clear PIN Loop → PIN Success → Withdraw
6. Invalid PIN → Retry → PIN Success → Withdraw
7. PIN Attempts Exhausted → Card Rejected
8. Invalid Withdraw Amount → Retry → Success (Account > 0)
9. Invalid Withdraw Amount → Retry → Success (Account = 0)
10. Exit Program

Enter your choice (1-10): 4

SCENARIO 4

Scenario: Valid Card → PIN Success → Account = 0
Full withdrawal depletes account, triggering account closure
Initial Context: account=$100, attempt=0

01 START [=] Initial State
   State machine begins execution from the initial pseudo-state.
02 TRANS PowerOn / showWelcome
   System event: ATM powers on and initializes the welcome display.
03 STATE Idle: entry / showingWelcome
   ATM enters Idle state; system displays welcome screen to attract users.
04 STATE Idle: exit / readCard or shutdown
   User action required: insert card to begin session, or operator shuts down system.
05 TRANS InsertedCard(validCard) / checkCard
   User action: customer inserts their ATM card into the reader.
06 CHOICE CardCheck: [isValidCard=true && attempt==0] → PromptPin
   Guard evaluates TRUE: card is valid and no prior attempts. Proceed to PIN entry.
07 STATE Prompt User For Pin: entry / promptingForPin
   ATM enters PIN prompt state; system waits for user to enter their PIN.
08 STATE Prompt User For Pin: exit / pinEntered
   User action required: submit PIN entry to proceed with authentication.
09 TRANS PinEntered(pin, attempt) / incrementAttempt
   User submits PIN; system routes to increment the attempt counter.
10 STATE Increment Attempt: entry / incrementingAttempt
   ATM enters Increment Attempt state; system begins tracking this PIN try.
11 STATE Increment Attempt: do / attempt = attempt + 1 (attempt: 0 → 1)
   System increments attempt counter: 0 → 1.
12 STATE Increment Attempt: exit / attemptIncrementedSuccessfully
   System completed: attempt counter updated successfully.
13 TRANS PinEntered(pin, attempt) / authenticating
   System forwards PIN and attempt count to authentication module.
14 STATE Authenticating: entry / authenticatingPin
   ATM enters Authenticating state; system verifies PIN against bank records.
15 STATE Authenticating: exit / pinValidation
   System completed: PIN validation result determined (valid or invalid).
16 TRANS PinValidation(isValidPin, attempt) / check
   System event: PIN validation result is evaluated.
17 CHOICE PinCheck: [isValidPin=true] → PromptAmt
   Guard evaluates TRUE: PIN is correct. Proceed to withdrawal menu.
18 STATE Prompt User For Amount to Withdraw: entry / promptingForWithdrawAmount
   ATM enters Amount Prompt state; system waits for user to enter withdrawal amount.
19 STATE Prompt User For Amount to Withdraw: exit / userEnteredAmount
   User action required: enter the desired withdrawal amount.
20 TRANS Withdraw(100) / computeWithdraw
   User requests full balance ($100); system begins computation.
21 STATE Compute Withdraw Amount: entry / processingWithdraw
   ATM enters Compute Amount state; system calculates new balance.
22 STATE Compute Withdraw Amount: do / computedAmount = accountAmount - withdrawAmount (100 - 100 = 0)
   System computes: $100 - $100 = $0.
23 STATE Compute Withdraw Amount: exit / withdrawAmountComputed
   System completed: withdrawal amount computation finished.
24 TRANS ProcessAmount(computedAmount) / check
   System event: computed balance is checked for validity.
25 CHOICE AmtCheck: [computedAmount >= 0] → Dispense
   Guard evaluates TRUE: sufficient funds available. Proceed to dispense cash.
26 STATE Dispense Cash: entry / dispensingCash
   ATM enters Dispense Cash state; system prepares to release cash.
27 STATE Dispense Cash: do / accountAmount = computedAmount (accountAmount + 0)
   System updates account balance to $0 after dispensing cash.
28 STATE Dispense Cash: exit / cashDispensed
   System completed: cash has been dispensed to user.
29 TRANS AccountStatus(accountAmount) / check
   System event: account balance is checked after withdrawal.
30 CHOICE AcctCheck: [accountAmount == 0] → CloseAccount
   Guard evaluates TRUE: balance is zero. Trigger account closure.
31 STATE Close Account: entry / showingClosureMessage
   ATM enters Close Account state; system processes account closure (balance = $0).
32 STATE Close Account: exit / userAcknowledge && accountClosed
   User action required: acknowledge closure and remove card.
33 TRANS UserRemovedCard / endSession
   User removes card; session ends with account closure processed.
34 STATE Idle: entry / showingWelcome
   ATM enters Idle state; system displays welcome screen to attract users.
35 STATE Idle: exit / readCard or shutdown
   User action required: insert card to begin session, or operator shuts down system.
36 END [=] Final State - Account closed - zero balance
   State machine reaches final state. Session outcome: Account closed - zero balance.

Scenario Complete
Result: Account closed
Final Context: account=$0, attempt=1

Press Enter to continue...
  
```



Figure 6

Scenario 5 Clear PIN Loop → PIN Success → Withdraw

```

ATM Operations - Scenario Menu
1. Power On → Shutdown
2. Invalid Card
3. Valid Card → PIN Success → Withdraw → Account > 0
4. Valid Card → PIN Success → Withdraw → Account = 0 (Close)
5. Clear PIN Loop → PIN Success → Withdraw
6. Invalid PIN → Retry → PIN Success → Withdraw
7. PIN Attempts Exhausted → Card Rejected
8. Invalid Withdraw Amount → Retry → Success (Account > 0)
9. Invalid Withdraw Amount → Retry → Success (Account = 0)
10. Exit Program

Enter your choice (1-10): 5

SCENARIO 5

Scenario: Clear PIN Loop → Success
User clears PIN entry once, then enters correct PIN
Initial Context: account=$100, attempt=0

01 START [*] Initial State
   State machine begins execution from the initial pseudo-state.
02 TRANS PowerOn / showWelcome
   System event: ATM powers on and initializes the welcome display.
03 STATE Idle: entry / showingWelcome
   ATM enters Idle state; system displays welcome screen to attract users.
04 STATE Idle: exit / readCard or shutdown
   User action required: insert card to begin session, or operator shuts down system.
05 TRANS InsertedCard(validCard) / checkCard
   User action: customer inserts their ATM card into the reader.
06 CHOICE CardCheck: [isValidCard=true && attempt==0] → PromptPin
   Guard evaluates TRUE: card is valid and no prior attempts. Proceed to PIN entry.
07 STATE Prompt User For Pin: entry / promptingForPin
   ATM enters PIN prompt state; system waits for user to enter their PIN.
08 STATE Prompt User For Pin: exit / pinEntered
   User action required: submit PIN entry to proceed with authentication.
09 TRANS UserClear(pin) / clearPin && rePromptForPin
   User clears PIN (e.g., pressed 'Clear' button); system re-prompts for PIN.
10 STATE Prompt User For Pin: entry / promptingForPin
   ATM enters PIN prompt state; system waits for user to enter their PIN.
11 STATE Prompt User For Pin: exit / pinEntered
   User action required: submit PIN entry to proceed with authentication.
12 TRANS PinEntered(pin, attempt) / incrementAttempt
   User submits PIN (after clearing); system increments attempt counter.
13 STATE Increment Attempt: entry / incrementingAttempt
   ATM enters Increment Attempt state; system begins tracking this PIN try.
14 STATE Increment Attempt: do / attempt = attempt + 1 (attempt: 0 → 1)
   System increments attempt counter: 0 → 1.
15 STATE Increment Attempt: exit / attemptIncrementedSuccessfully
   System completed: attempt counter updated successfully.
16 TRANS PinEntered(pin, attempt) / authenticating
   System forwards PIN and attempt count to authentication module.
17 STATE Authenticating: entry / authenticatingPin
   ATM enters Authenticating state; system verifies PIN against bank records.
18 STATE Authenticating: exit / pinValidation
   System completed: PIN validation result determined (valid or invalid).
19 TRANS PinValidation(isValidPin, attempt) / check
   System event: PIN validation result is evaluated.
20 CHOICE PinCheck: [isValidPin=true] → PromptAmt
   Guard evaluates TRUE: PIN is correct. Proceed to withdrawal menu.
21 STATE Prompt User For Amount to Withdraw: entry / promptingForWithdrawAmount
   ATM enters Amount Prompt state; system waits for user to enter withdrawal amount.
22 STATE Prompt User For Amount to Withdraw: exit / userEnteredAmount
   User action required: enter the desired withdrawal amount.
23 TRANS Withdraw($0) / computeWithdraw
   User requests $30 withdrawal; system begins computation.
24 STATE Compute Withdraw Amount: entry / processingWithdraw
   ATM enters Compute Amount state; system calculates new balance.
25 STATE Compute Withdraw Amount: do / computedAmount = accountAmount - withdrawAmount (100 - 30 = 70)
   System computes: $100 - $30 = $70.
26 STATE Compute Withdraw Amount: exit / withdrawAmountComputed
   System completed: withdrawal amount computation finished.
27 TRANS ProcessAmount(computedAmount) / check
   System event: computed balance is checked for validity.
28 CHOICE AmtCheck: [computedAmount >= 0] → Dispense
   Guard evaluates TRUE: sufficient funds available. Proceed to dispense cash.
29 STATE Dispense Cash: entry / dispensingCash
   ATM enters Dispense Cash state; system prepares to release cash.
30 STATE Dispense Cash: do / accountAmount = computedAmount (accountAmount + 70)
   System updates account balance to $70 after dispensing cash.
31 STATE Dispense Cash: exit / cashDispensed
   System completed: cash has been dispensed to user.
32 TRANS AccountStatus(accountAmount) / check
   System event: account balance is checked after withdrawal.
33 CHOICE AcctCheck: [accountAmount > 0] → ShowNewAmt
   Guard evaluates TRUE: balance remains positive. Show new balance.
34 STATE Show Account New Amount: entry / showingAccount
   ATM enters Show Balance state; system displays updated account balance to user.
35 STATE Show Account New Amount: exit / userAcknowledge
   User action required: acknowledge the new balance display.
36 TRANS UserRemovedCard / endSession
   User removes card after viewing balance; session ends successfully.
37 STATE Idle: entry / showingWelcome
   ATM enters Idle state; system displays welcome screen to attract users.
38 STATE Idle: exit / readCard or shutdown
   User action required: insert card to begin session, or operator shuts down system.
39 END [*] Final State - Transaction complete - balance: $70
   State machine reaches final state. Session outcome: Transaction complete - balance: $70.

Scenario Complete
Result: Transaction successful - remaining balance: $70
Final Context: account=$70, attempt=1

Press Enter to continue...
  
```

10:38 AM  
1/6/2026

Figure 7

Scenario 6 Invalid PIN → Retry → PIN Success → Withdraw

```

Scenario Complete
Result: Transaction successful - remaining balance: $70
Final Context: account=$70, attempt=1

Press Enter to continue...

ATM Operations - Scenario Menu
1. Power On + Shutdown
2. Invalid Card
3. Valid Card + PIN Success + Withdraw + Account > 0
4. Valid Card + PIN Success + Withdraw + Account = 0 (Close)
5. Clear PIN Loop + PIN Success + Withdraw
6. Invalid PIN + Retry + PIN Success + Withdraw
7. PIN Attempts Exhausted + Card Rejected
8. Invalid Withdraw Amount + Retry + Success (Account > 0)
9. Invalid Withdraw Amount + Retry + Success (Account = 0)
10. Exit Program

Enter your choice (1-10): 6

SCENARIO 6

Scenario: Invalid PIN + Retry + Success
First PIN attempt fails, second attempt succeeds
Initial Context: account=$100, attempt=0

01 START [=] Initial State
   State machine begins execution from the initial pseudo-state.
02 TRANS PowerOn / showWelcome
   System event: ATM powers on and initializes the welcome display.
03 STATE Idle: entry / showingWelcome
   ATM enters Idle state; system displays welcome screen to attract users.
04 STATE Idle: exit / readCard or shutdown
   User action required: insert card to begin session, or operator shuts down system.
05 TRANS InsertCard(validCard) / checkCard
   User action: customer inserts their ATM card into the reader.
06 CHOICE CardCheck: [isValidCard=true && attempt=0] → PromptPin
   Guard evaluates TRUE: card is valid and no prior attempts. Proceed to PIN entry.
07 STATE Prompt User For Pin: entry / promptingForPin
   ATM enters PIN prompt state; system waits for user to enter their PIN.
08 STATE Prompt User For Pin: exit / pinEntered
   User action required: submit PIN entry to proceed with authentication.
09 TRANS PinEntered(pin, attempt) / incrementAttempt
   User submits INCORRECT PIN; system increments attempt counter.
10 STATE Increment Attempt: entry / incrementingAttempt
   ATM enters Increment Attempt state; system begins tracking this PIN try.
11 STATE Increment Attempt: do / attempt = attempt + 1 (attempt: 0 + 1)
   System increments attempt counter: 0 + 1.
12 STATE Increment Attempt: exit / attemptIncrementedSuccessfully
   System completed: attempt counter updated successfully.
13 TRANS PinEntered(pin, attempt) / authenticating
   System forwards PIN and attempt count to authentication module.
14 STATE Authenticating: entry / authenticatingPin
   ATM enters Authenticating state; system verifies PIN against bank records.
15 STATE Authenticating: exit / pinValidation
   System completed: PIN validation result determined (valid or invalid).
16 TRANS PinValidation(isValidPin, attempt) / check
   System event: PIN validation result is evaluated.
17 CHOICE PinCheck: [isValidPin=false && attempt<3 <= MaxTries[3]] → InvalidPin
   Guard evaluates TRUE: PIN wrong but retries remain. Allow another attempt.
18 STATE Invalid Pin: entry / showingError
   ATM enters Invalid PIN state; system notifies user of incorrect PIN.
19 STATE Invalid Pin: exit / userAcknowledge
   User action required: acknowledge the error to retry PIN entry.
20 TRANS PromptForPin(attempt) / promptingForPin
   System allows retry (attempt 1 of 3); re-prompts for PIN.
21 STATE Prompt User For Pin: entry / promptingForPin
   ATM enters PIN prompt state; system waits for user to enter their PIN.
22 STATE Prompt User For Pin: exit / pinEntered
   User action required: submit PIN entry to proceed with authentication.
23 TRANS PinEntered(pin, attempt) / incrementAttempt
   User submits CORRECT PIN; system increments attempt counter.
24 STATE Increment Attempt: entry / incrementingAttempt
   ATM enters Increment Attempt state; system begins tracking this PIN try.
25 STATE Increment Attempt: do / attempt = attempt + 1 (attempt: 1 + 2)
   System increments attempt counter: 1 + 2.
26 STATE Increment Attempt: exit / attemptIncrementedSuccessfully
   System completed: attempt counter updated successfully.
27 TRANS PinEntered(pin, attempt) / authenticating
   System forwards PIN and attempt count to authentication module.
28 STATE Authenticating: entry / authenticatingPin
   ATM enters Authenticating state; system verifies PIN against bank records.
29 STATE Authenticating: exit / pinValidation
   System completed: PIN validation result determined (valid or invalid).
30 TRANS PinValidation(isValidPin, attempt) / check
   System event: PIN validation result is evaluated.
31 CHOICE PinCheck: [isValidPin=true] → PromptMenu
   Guard evaluates TRUE: PIN is correct. Proceed to withdrawal menu.
32 STATE Prompt User For Amount to Withdraw: entry / promptingForWithdrawAmount
   ATM enters Amount Prompt state; system waits for user to enter withdrawal amount.
33 STATE Prompt User For Amount to Withdraw: exit / userEnteredAmount
   User action required: enter the desired withdrawal amount.
34 TRANS Withdraw(Cash) / computeWithdraw
   User requests $25 withdrawal; system begins computation.
35 STATE Compute Withdraw Amount: entry / processingWithdraw
   ATM enters Compute Amount state; system calculates new balance.
36 STATE Compute Withdraw Amount: do / computedAmount = accountAmount - withdrawAmount (100 - 25 = 75)
   System computes: 100 - 25 = 75.
37 STATE Compute Withdraw Amount: exit / withdrawAmountComputed
   System completed: withdrawal amount computation finished.
38 TRANS ProcessAmount(computedAmount) / check
   System event: computed balance is checked for validity.
39 CHOICE AmtCheck: [computedAmount >= 0] → Dispense
   Guard evaluates TRUE: sufficient funds available. Proceed to dispense cash.
40 STATE Dispense Cash: entry / dispensingCash
   ATM enters Dispense Cash state; system prepares to release cash.
41 STATE Dispense Cash: do / accountAmount = computedAmount (AccountAmount + 75)
   System updates account balance to $75 after dispensing cash.
42 STATE Dispense Cash: exit / cashDispensed
   System completed: cash has been dispensed to user.
43 TRANS AccountStatus(accountAmount) / check
   System event: account balance is checked after withdrawal.
44 CHOICE AccountCheck: [accountAmount > 0] → ShowNewBal
   Guard evaluates TRUE: balance remains positive. Show new balance.
45 STATE Show Account New Amount: entry / showingAccount
   ATM enters Show Balance state; system displays updated account balance to user.
46 STATE Show Account New Amount: exit / userAcknowledge
   User action required: acknowledge the new balance display.
47 TRANS UserRemovesCard / endSession
   User removes card after viewing balance; session ends successfully.
48 STATE Idle: entry / showingWelcome
   ATM enters Idle state; system displays welcome screen to attract users.
49 STATE Idle: exit / readCard or shutdown
   User action required: insert card to begin session, or operator shuts down system.
50 END [=] Final State - Transaction complete - balance: $75
   State machine reaches final state. Session outcome: Transaction complete - balance: $75.

Scenario Complete
Result: Transaction successful - remaining balance: $75
Final Context: account=$75, attempt=2

Press Enter to continue...

```



**Figure 8**  
*Scenario 7 PIN Attempts Exhausted → Card Rejected*

Windows PowerShell

Scenario Complete

Result: Transaction successful - Remaining balance: \$75  
Final Context: account=\$75, attempt=2

Press Enter to continue...

ATM Operations - Scenario Menu

1. Power On → Shutdown
2. Invalid Card
3. Valid Card → PIN Success → Withdraw → Account > 0
4. Valid Card → PIN Success → Withdraw → Account = 0 (Close)
5. Clear PIN Loop → PIN Success → Withdraw
6. Invalid PIN → Retry → PIN Success → Withdraw
7. PIN Attempts Exhausted → Card Rejected
8. Invalid Withdraw Amount → Retry → Success (Account > 0)
9. Invalid Withdraw Amount → Retry → Success (Account = 0)
10. Exit Program

Enter your choice (1-10): 7

SCENARIO 7

Scenario: PIN Attempts Exhausted

User fails PIN 4 times, card is rejected

Initial Context: account=\$100, attempt=0

```

01 START [*] Initial State
   State machine begins execution from the initial pseudo-state.
02 TRANS PowerOn / showWelcome
   System event: ATM powers on and initializes the welcome display.
03 STATE Idle: entry / showingWelcome
   ATM enters Idle state; system displays welcome screen to attract users.
04 STATE Idle: exit / readCard or shutdown
   User action required: insert card to begin session, or operator shuts down system.
05 TRANS InsertedCard(validCard) / checkCard
   User action: customer inserts their ATM card into the reader.
06 CHOICE CardCheck: [IsValidPin=true && attempt=0] → promptPin
   Guard evaluates TRUE: card is valid and no prior attempts. Proceed to PIN entry.
07 STATE Prompt User For Pin: entry / promptingForPin
   ATM enters PIN prompt state; system waits for user to enter their PIN.
08 STATE Prompt User For Pin: exit / pinEntered
   User action required: submit PIN entry to proceed with authentication.
09 TRANS PinEntered(pin, attempt) / incrementAttempt
   User submits INCORRECT PIN (attempt #1); system increments counter.
10 STATE Increment Attempt: entry / incrementingAttempt
   ATM enters Increment Attempt state; system begins tracking this PIN try.
11 STATE Increment Attempt: do / attempt = attempt + 1 (attempt: 0 → 1)
   System increments attempt counter: 0 → 1.
12 STATE Increment Attempt: exit / attemptIncrementedSuccessfully
   System completed: attempt counter updated successfully.
13 TRANS PinEntered(pin, attempt) / authenticating
   System forwards PIN and attempt count to authentication module.
14 STATE Authenticating: entry / authenticatingPin
   ATM enters Authenticating state; system verifies PIN against bank records.
15 STATE Authenticating: exit / pinValidation
   System completed: PIN validation result determined (valid or invalid).
16 TRANS PinValidation(IsValidPin, attempt) / check
   System event: PIN validation result is evaluated.
17 CHOICE PinCheck: [IsValidPin=false && attempt(1) ≤ Max_tries(3)] → InvalidPin
   Guard evaluates TRUE: PIN wrong but retries remain. Allow another attempt.
18 STATE Invalid Pin: entry / showingError
   ATM enters Invalid PIN state; system notifies user of incorrect PIN.
19 STATE Invalid Pin: exit / userAcknowledge
   User action required: acknowledge the error to retry PIN entry.
20 TRANS PromptForPin(attempt) / promptForPin
   System allows retry (attempt 1 of 3); re-prompts for PIN.
21 STATE Prompt User For Pin: entry / promptingForPin
   ATM enters PIN prompt state; system waits for user to enter their PIN.
22 STATE Prompt User For Pin: exit / pinEntered
   User action required: submit PIN entry to proceed with authentication.
23 TRANS PinEntered(pin, attempt) / incrementAttempt
   User submits INCORRECT PIN (attempt #2); system increments counter.
24 STATE Increment Attempt: entry / incrementingAttempt
   ATM enters Increment Attempt state; system begins tracking this PIN try.
25 STATE Increment Attempt: do / attempt = attempt + 1 (attempt: 1 → 2)
   System increments attempt counter: 1 → 2.
26 STATE Increment Attempt: exit / attemptIncrementedSuccessfully
   System completed: attempt counter updated successfully.
27 TRANS PinEntered(pin, attempt) / authenticating
   System forwards PIN and attempt count to authentication module.
28 STATE Authenticating: entry / authenticatingPin
   ATM enters Authenticating state; system verifies PIN against bank records.
29 STATE Authenticating: exit / pinValidation
   System completed: PIN validation result determined (valid or invalid).
30 TRANS PinValidation(IsValidPin, attempt) / check
   System event: PIN validation result is evaluated.
31 CHOICE PinCheck: [IsValidPin=false && attempt(2) ≤ Max_tries(3)] → InvalidPin
   Guard evaluates TRUE: PIN wrong but retries remain. Allow another attempt.
32 STATE Invalid Pin: entry / showingError
   ATM enters Invalid PIN state; system notifies user of incorrect PIN.
33 STATE Invalid Pin: exit / userAcknowledge
   User action required: acknowledge the error to retry PIN entry.
34 TRANS PromptForPin(attempt) / promptForPin
   System allows retry (attempt 2 of 3); re-prompts for PIN.
35 STATE Prompt User For Pin: entry / promptingForPin
   ATM enters PIN prompt state; system waits for user to enter their PIN.
36 STATE Prompt User For Pin: exit / pinEntered
   User action required: submit PIN entry to proceed with authentication.
37 TRANS PinEntered(pin, attempt) / incrementAttempt
   User submits INCORRECT PIN (attempt #3); system increments counter.
38 STATE Increment Attempt: entry / incrementingAttempt
   ATM enters Increment Attempt state; system begins tracking this PIN try.
39 STATE Increment Attempt: do / attempt = attempt + 1 (attempt: 2 → 3)
   System increments attempt counter: 2 → 3.
40 STATE Increment Attempt: exit / attemptIncrementedSuccessfully
   System completed: attempt counter updated successfully.
41 TRANS PinEntered(pin, attempt) / authenticating
   System forwards PIN and attempt count to authentication module.
42 STATE Authenticating: entry / authenticatingPin
   ATM enters Authenticating state; system verifies PIN against bank records.
43 STATE Authenticating: exit / pinValidation
   System completed: PIN validation result determined (valid or invalid).
44 TRANS PinValidation(IsValidPin, attempt) / check
   System event: PIN validation result is evaluated.
45 CHOICE PinCheck: [IsValidPin=false && attempt(3) ≤ Max_tries(3)] → InvalidPin
   Guard evaluates TRUE: PIN wrong but retries remain. Allow another attempt.
46 STATE Invalid Pin: entry / showingError
   ATM enters Invalid PIN state; system notifies user of incorrect PIN.
47 STATE Invalid Pin: exit / userAcknowledge
   User action required: acknowledge the error to retry PIN entry.
48 TRANS PromptForPin(attempt) / promptForPin
   System allows retry (attempt 3 of 3); re-prompts for PIN.
49 STATE Prompt User For Pin: entry / promptingForPin
   ATM enters PIN prompt state; system waits for user to enter their PIN.
50 STATE Prompt User For Pin: exit / pinEntered
   User action required: submit PIN entry to proceed with authentication.
51 TRANS PinEntered(pin, attempt) / incrementAttempt
   User submits INCORRECT PIN (attempt #4); system increments counter.
52 STATE Increment Attempt: entry / incrementingAttempt
   ATM enters Increment Attempt state; system begins tracking this PIN try.
53 STATE Increment Attempt: do / attempt = attempt + 1 (attempt: 3 → 4)
   System increments attempt counter: 3 → 4.
54 STATE Increment Attempt: exit / attemptIncrementedSuccessfully
   System completed: attempt counter updated successfully.
55 TRANS PinEntered(pin, attempt) / authenticating
   System forwards PIN and attempt count to authentication module.
56 STATE Authenticating: entry / authenticatingPin
   ATM enters Authenticating state; system verifies PIN against bank records.
57 STATE Authenticating: exit / pinValidation
   System completed: PIN validation result determined (valid or invalid).
58 TRANS PinValidation(IsValidPin, attempt) / check
   System event: PIN validation result is evaluated.
59 CHOICE PinCheck: [IsValidPin=false && attempt(4) > Max_tries(3)] → AttemptExhausted
   Guard evaluates TRUE: PIN wrong and no retries left. Reject card.
60 STATE Attempt Exhausted: entry / showingError
   ATM enters Attempt Exhausted state; system rejects card after too many failures.
61 STATE Attempt Exhausted: exit / userAcknowledge && userRemoveCard
   User action required: acknowledge error and remove rejected card.
62 TRANS UserRemovedCard / endSession
   User removes rejected card; session ends with security lockout.
63 STATE Idle: entry / showingWelcome

```

Scenario Complete

Result: Card rejected - too many PIN attempts  
Final Context: account=\$100, attempt=4

Press Enter to continue...

10:43 AM  
1/6/2026

Figure 9

Scenario 8 Invalid Withdraw Amount → Retry → Success (Account > 0)

```

Windows PowerShell

Scenario Complete
Result: Card rejected - too many PIN attempts
Final Context: account=$100, attempt=4

Press Enter to continue...

ATM Operations - Scenario Menu
1. Power On + Shutdown
2. Invalid Card
3. Valid Card + PIN Success + Withdraw + Account > 0
4. Valid Card + PIN Success + Withdraw + Account = 0 (Close)
5. Clear PIN loop + PIN Success + Withdraw
6. Invalid PIN + Retry + PIN Success + Withdraw
7. PIN Attempts Exhausted + Card Rejected
8. Invalid Withdraw Amount + Retry + Success (Account > 0)
9. Invalid Withdraw Amount + Retry + Success (Account = 0)
10. Exit Program

Enter your choice (1-10): 8

SCENARIO 8

Scenario: Invalid Amount → Success (Account > 0)
User enters amount exceeding balance, retries with valid amount
Initial Context: account=$100, attempt=0

01 START [*] Initial State
  State machine begins execution from the initial pseudo-state.
02 TRANS PowerOn / showWelcome
  System event: ATM powers on and initializes the welcome display.
03 STATE Idle: entry / showingWelcome
  ATM enters Idle state; system displays welcome screen to attract users.
04 STATE Idle: exit / shutdown or shutdown
  User action required: insert card to begin session, or operator shuts down system.
05 TRANS InsertedCard(validCard) / checkCard
  User action: customer inserts their ATM card into the reader.
06 CHOICE CardCheck: [isValidCard=true & attempt=0] → PromptPin
  Guard evaluates TRUE: card is valid and no prior attempts. Proceed to PIN entry.
07 STATE Prompt User For Pin: entry / promptingForPin
  ATM enters PIN prompt state; system waits for user to enter their PIN.
08 STATE Prompt User For Pin: exit / pinEntered
  User action required: submit PIN entry to proceed with authentication.
09 TRANS PinEntered(pin, attempt) / incrementAttempt
  User submits PIN; system increments attempt counter.
10 STATE Increment Attempt: entry / incrementingAttempt
  ATM enters Increment Attempt state; system begins tracking this PIN try.
11 STATE Increment Attempt: do / attempt = attempt + 1 (attempt: 0 + 1)
  System increments attempt counter: 0 + 1.
12 STATE Increment Attempt: exit / attemptIncrementedSuccessfully
  System completed: attempt counter updated successfully.
13 TRANS PinEntered(pin, attempt) / authenticating
  System forwards PIN and attempt count to authentication module.
14 STATE Authenticating: entry / authenticatingPin
  ATM enters Authenticating state; system verifies PIN against bank records.
15 STATE Authenticating: exit / pinValidation
  System collected: PIN validation result determined (valid or invalid).
16 TRANS PinValidation(isValidPin, attempt) / check
  System event: PIN validation result is evaluated.
17 CHOICE PinCheck: [isValidPin=true] → PromptAmt
  Guard evaluates TRUE: PIN is correct. Proceed to withdrawal menu.
18 STATE Prompt User For Amount to Withdraw: entry / promptingForWithdrawAmount
  ATM enters Amount Prompt state; system waits for user to enter withdrawal amount.
19 STATE Prompt User For Amount to Withdraw: exit / userEnteredAmount
  User action required: enter the desired withdrawal amount.
20 TRANS Withdraw(150) / computeWithdraw
  User requests $150 (EXCEEDS $100 balance); computation begins.
21 STATE Compute Withdraw Amount: entry / processingWithdraw
  ATM enters Compute Amount state; system calculates new balance.
22 STATE Compute Withdraw Amount: do / computedAmount = accountAmount - withdrawAmount (100 - 150 = -50)
  System computes: $100 - $150 = -$50.
23 STATE Compute Withdraw Amount: exit / withdrawAmountComputed
  System completed: withdrawal amount computation finished.
24 TRANS ProcessAmount(computedAmount) / check
  System event: computed balance is checked for validity.
25 CHOICE AmtCheck: [computedAmount <= 0] → InvalidAmount
  Guard evaluates TRUE: insufficient funds. Route to error handling.
26 STATE Invalid Amount: entry / showingError
  ATM enters Invalid Amount state; system shows insufficient funds error.
27 STATE Invalid Amount: exit / userAcknowledge
  User action required: acknowledge error to enter a new amount.
28 TRANS InvalidAmount(withdrawAmount) / promptUserForNewAmount
  Insufficient funds detected; system prompts user to enter a smaller amount.
29 STATE Prompt User For Amount to Withdraw: entry / promptingForWithdrawAmount
  ATM enters Amount Prompt state; system waits for user to enter withdrawal amount.
30 STATE Prompt User For Amount to Withdraw: exit / userEnteredAmount
  User action required: enter the desired withdrawal amount.
31 TRANS Withdraw(40) / computeWithdraw
  User requests $40 (valid amount); system begins computation.
32 STATE Compute Withdraw Amount: entry / processingWithdraw
  ATM enters Compute Amount state; system calculates new balance.
33 STATE Compute Withdraw Amount: do / computedAmount = accountAmount - withdrawAmount (100 - 40 = 60)
  System computes: $100 - $40 = $60.
34 STATE Compute Withdraw Amount: exit / withdrawAmountComputed
  System completed: withdrawal amount computation finished.
35 TRANS ProcessAmount(computedAmount) / check
  System event: computed balance is checked for validity.
36 CHOICE AmtCheck: [computedAmount > 0] → Dispense
  Guard evaluates TRUE: sufficient funds available. Proceed to dispense cash.
37 STATE Dispense Cash: entry / dispensingCash
  ATM enters Dispense Cash state; system prepares to release cash.
38 STATE Dispense Cash: do / accountAmount = computedAmount (accountAmount + 60)
  System updates account balance to $60 after dispensing cash.
39 STATE Dispense Cash: exit / cashDispensed
  System completed: cash has been dispensed to user.
40 TRANS AccountStatus(accountAmount) / check
  System event: account balance is checked after withdrawal.
41 CHOICE AccCheck: [accountAmount > 0] → ShowNewAmt
  Guard evaluates TRUE: balance remains positive. Show new balance.
42 STATE Show Account New Amount: entry / showingAccount
  ATM enters Show Balance state; system displays updated account balance to user.
43 STATE Show Account New Amount: exit / userAcknowledge
  User action required: acknowledge the new balance display.
44 TRANS UserRemovedCard / endSession
  User removes card after viewing balance; session ends successfully.
45 STATE Idle: entry / showingWelcome
  ATM enters Idle state; system displays welcome screen to attract users.
46 STATE Idle: exit / shutdown or shutdown
  User action required: insert card to begin session, or operator shuts down system.
47 END [*] Final State - Transaction complete - balance: $60.
  State machine reaches final state. Session outcome: Transaction complete - balance: $60.

Scenario Complete
Result: Transaction successful - remaining balance: $60
Final Context: account=$60, attempt=1

Press Enter to continue...

```

10:47 AM  
1/6/2026

Figure 10

Scenario 9 Invalid Withdraw Amount → Retry → Success (Account = 0)

```

Scenario Complete
Result: Transaction successful - remaining balance: $60
Final Context: account=$60, attempt=1

Press Enter to continue...

ATM Operations - Scenario Menu
1. Power On → Shutdown
2. Invalid Card
3. Valid Card → PIN Success → Withdraw → Account > 0
4. Valid Card → PIN Success → Withdraw → Account = 0 (Close)
5. Clear PIN Loop → PIN Success → Withdraw
6. Invalid PIN → Retry → PIN Success → Withdraw
7. PIN Attempts Exhausted → Card Rejected
8. Invalid Withdraw Amount → Retry → Success (Account > 0)
9. Invalid Withdraw Amount → Retry → Success (Account = 0)
10. Exit Program

Enter your choice (1-10): 9

SCENARIO 9

Scenario: Invalid Amount → Success (Account = 0)
User retries with full balance withdrawal, triggering account closure
Initial Context: account=$100, attempt=0

01 START [x] Initial State
02 State machine begins execution from the initial pseudo-state.
03 TRANS PowerOn / showWelcome
04 STATE Idle: entry / showingWelcome
05 STATE Idle: exit / readCard or shutdown
06 TRANS InsertedCard(validCard) / checkCard
07 STATE Prompt User For Pin: entry / promptingForPin
08 STATE Prompt User For Pin: exit / pinEntered
09 TRANS PinEntered(pin, attempt) / incrementAttempt
10 STATE Increment Attempt: entry / incrementingAttempt
11 STATE Increment Attempt: do / attempt = attempt + 1 (attempt: 0 → 1)
12 STATE Increment Attempt: exit / attemptIncrementedSuccessfully
13 TRANS PinEntered(pin, attempt) / authenticating
14 STATE Authenticating: entry / authenticatingPin
15 STATE Authenticating: exit / pinValidation
16 TRANS PinValidation(isValidPin, attempt) / check
17 CHOICE PinCheck: [isValidPin == true] → proceed
18 STATE Prompt User For Amount to Withdraw: entry / promptingForWithdrawAmount
19 STATE Prompt User For Amount to Withdraw: exit / userEnteredAmount
20 TRANS Withdrawal(200) / computeWithdrawal
21 STATE Compute Withdraw Amount: entry / processingWithdrawal
22 STATE Compute Withdraw Amount: do / computedAmount = accountAmount - withdrawalAmount (100 - 200 = -100)
23 STATE Compute Withdraw Amount: exit / withdrawalAmountComputed
24 TRANS ProcessAmount(computedAmount) / check
25 CHOICE AuthCheck: [computedAmount(-100) < 0] → InvalidAmount
26 STATE Invalid Amount: entry / showingError
27 STATE Invalid Amount: exit / userAcknowledge
28 TRANS InvalidAmount(withdrawAmount) / promptUserForNewAmount
29 STATE Prompt User For Amount to Withdraw: entry / promptingForWithdrawAmount
30 STATE Prompt User For Amount to Withdraw: exit / userEnteredAmount
31 TRANS Withdrawal(100) / computeWithdrawal
32 STATE Compute Withdraw Amount: entry / processingWithdrawal
33 STATE Compute Withdraw Amount: do / computedAmount = accountAmount - withdrawalAmount (100 - 100 = 0)
34 STATE Compute Withdraw Amount: exit / withdrawalAmountComputed
35 TRANS ProcessAmount(computedAmount) / check
36 CHOICE AuthCheck: [computedAmount == 0] → Dispense
37 STATE Dispense Cash: entry / dispensingCash
38 STATE Dispense Cash: do / accountAmount = computedAmount (accountAmount + 0)
39 STATE Dispense Cash: exit / cashDispensed
40 TRANS AccountStatus(accountAmount) / check
41 CHOICE AuthCheck: [accountAmount == 0] → CloseAccount
42 STATE Close Account: entry / showingClosureMessage
43 STATE Close Account: exit / userAcknowledges accountClosed
44 TRANS UserRemovedCard / endSession
45 STATE Idle: entry / showingWelcome
46 STATE Idle: exit / readCard or shutdown
47 END [x] Final State - Account closed - zero balance
State machine reaches final state. Session outcome: Account closed - zero balance.

Scenario Complete
Result: Account closed
Final Context: account=$0, attempt=1

Press Enter to continue...

```

**Figure 11**  
*Exit Program*

