

|  |  |
| --- | --- |
| **State-Account** | |
| **Line** | **Code** |
| **1**  **2**  **3**  **4**  **5**  **6**  **7**  **8**  **9**  **10**  **11**  **12**  **13**  **14**  **15**  **16**  **17**  **18**  **19**  **20**  **21**  **22**  **23**  **24**  **25**  **26**  **27**  **28**  **29**  **30**  **31**  **32**  **33**  **34**  **35**  **36**  **37**  **38**  **39**  **40**  **41**  **42**  **43**  **44**  **45**  **46**  **47**  **48**  **49**  **50**  **51**  **52**  **53**  **54**  **55**  **56**  **57**  **58**  **59**  **60**  **61**  **62**  **63**  **64**  **65**  **66**  **67**  **68**  **69**  **70**  **71**  **72**  **73**  **74**  **75**  **76**  **77**  **78**  **79**  **80**  **81**  **82**  **83**  **84**  **85**  **86**  **87**  **88**  **89**  **90**  **91**  **92**  **93**  **94**  **95**  **96**  **97**  **98**  **99**  **100**  **101**  **102**  **103**  **104**  **105**  **106**  **107**  **108**  **109**  **110**  **111**  **112**  **113**  **114**  **115**  **116**  **117**  **118**  **119**  **120**  **121**  **122**  **123**  **124**  **125**  **126**  **127**  **128**  **129**  **130**  **131**  **132**  **133**  **134** | **using** System;  **namespace** State\_Account {  **class** Account {//Context  **private** AccountState accountState = Empty;  **public static** AccountState MoneyPresent = **new** StateMoneyPresent();  **public static** AccountState Empty = **new** StateEmpty();  **public static** AccountState OverDrawn = **new** StateOverDrawn();  **public static** AccountState OverDrawnLimit = **new** StateOverDrawnLimit();  **public double** Balance;  **public readonly double** ODLimit;  **public** Account(**double** amount, **double** od) {  **if** (amount < 0) **throw new** Exception("amount must >=0");  Balance = amount;  ODLimit = od;  State = (Balance > 0) ? MoneyPresent : Empty;  }  **public** AccountState State {  **get** {  **return** accountState;  }  **set** {  accountState = **value**;  }  }  **public void** withdraw(**double** amount) {  **if** (amount <= 0) **throw new** Exception("Invalid Amount!");  accountState.Withdraw(**this**, amount);  }  **public void** deposit(**double** amount) {  **if** (amount <= 0) **throw new** Exception("Invalid Amount!");  accountState.Deposit(**this**, amount);  }  **public void** showState() {  **if** (accountState != **null**) Console.WriteLine(accountState);  }  }  **abstract class** AccountState {  **abstract public void** Withdraw(Account account, **double** amount);  **abstract public void** Deposit(Account account, **double** amount);  }  **class** StateMoneyPresent : AccountState {  **public override void** Withdraw(Account account, **double** amount) {  **double** newBalance = account.Balance - amount;  AccountState state = **null**;  **if** (newBalance > 0) state = Account.MoneyPresent;  **else** {  **if** (newBalance == 0) state = Account.Empty;  **else** {  **if** (newBalance == -account.ODLimit) state = Account.OverDrawnLimit;  **else** {  **if** (newBalance > -account.ODLimit) state = Account.OverDrawn;  }  }  }  **if** (state != **null**) {  account.Balance = newBalance;  account.State = state; //State Transition happen here….  } **else** Console.WriteLine("Cannot withdraw");  }  **public override void** Deposit(Account account, **double** amount) {  account.Balance += amount;  }  **override public** String ToString() {  **return** "MoneyPresent";  }  }  **class** StateEmpty : AccountState {  **public override void** Withdraw(Account account, **double** amount) {  **double** newBalance = account.Balance - amount;  AccountState state = **null**;  **if** (newBalance == -account.ODLimit) state = Account.OverDrawnLimit;  **else** {  **if** (newBalance > -account.ODLimit) state = Account.OverDrawn;  }  **if** (state != **null**) {  account.Balance = newBalance;  account.State = state;  } **else** Console.WriteLine("Cannot withdraw");  }  **public override void** Deposit(Account account, **double** amount) {  account.Balance += amount;  account.State = Account.MoneyPresent;  }  **override public** String ToString() {  **return** "Empty";  }  }  **class** StateOverDrawn : AccountState {  **public override void** Deposit(Account account, **double** amount) {  //Complete the method here...  }  **public override void** Withdraw(Account account, **double** amount) {  //Complete the method here...  }  **override public** String ToString() {  **return** "OverDrawn";  }  }  **class** StateOverDrawnLimit : AccountState {  **public override void** Deposit(Account account, **double** amount) {  //Complete the method here...  }  **public override void** Withdraw(Account account, **double** amount) {  //Complete the method here...  }  **override public** String ToString() {  **return** "OverDrawnLimit";  }  }  **class** Program {  **static void** Main(**string**[] args) {  // Setup context in a state  Account acc = **new** Account(1000, 500);  // Start transitions  acc.showState(); //Money Present  acc.deposit(100); //Current Balance is $1100  acc.showState(); //Money Present  acc.withdraw(1200); //Current Balance is -$100  acc.showState(); //Overdrawn  acc.withdraw(400); //Current Balance is -$500  acc.showState(); //Overdraft Limit  acc.deposit(500); //Current Balance is $0  acc.showState(); //Empty  acc.withdraw(1000); // Cannot withdraw  Console.ReadKey();  }  }  } |

**Output:**

MoneyPresent

MoneyPresent

OverDrawn

OverDrawnLimit

Empty

Cannot withdraw