

|  |  |
| --- | --- |
| **ChainOfResponsibilityATM (The Client)** | |
| **Line** | **Code** |
| **1**  **2**  **3**  **4**  **5**  **6**  **7**  **8**  **9**  **10**  **11**  **12**  **13**  **14**  **15**  **16**  **17**  **18**  **19**  **20**  **21**  **22** | **using** System;  **namespace** ChainOfResposibilityATM {  **class** Program {  **static void** Main(**string**[] args) {  **while** (**true**) {  Console.Write("Amount>>$");  **string** sAmount = Console.ReadLine().Trim();  **if** (sAmount == "") **break**;  **try** {  **uint** amount = **uint**.Parse(sAmount);  ATM.Withdraw(amount);  ATM.Show();  } **catch** (Exception e) {  Console.WriteLine("Withdraw fail due to:{0}",e.Message);  }  }  Console.WriteLine("Program ended");  Console.ReadKey();  }  }  } |

|  |  |
| --- | --- |
| **ChainOfResponsibilityATM (ATM Class)** | |
| **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** | **using** System;  **namespace** ChainOfResposibilityATM {  **class** ATM {  **static private** MoneySlot firstMoneySlot;  **static** ATM() {  firstMoneySlot = **new** MoneySlot100(20,  **new** MoneySlot50(20,  **new** MoneySlot20(20,  **new** MoneySlot5(20))));  }  **static public void** Withdraw(**uint** amount) {  firstMoneySlot.Withdraw(amount);  }  **static public void** Show() {  Console.WriteLine("Remaining:");  MoneySlot ms = firstMoneySlot;  **while** (ms != **null**) {  ms.Show();  ms = ms.Successor;  }  }  }  } |

|  |  |
| --- | --- |
| **ChainOfResponsibilityATM (MoneySlot Class)** | |
| **Line** | **Code** |
| **1**  **2**  **3**  **4**  **5**  **6**  **7**  **8**  **9**  **10**  **11**  **12**  **13**  **14**  **15**  **16**  **17**  **18**  **19**  **20** | **using** System;  **namespace** ChainOfResposibilityATM {  **abstract class** MoneySlot {  **public readonly ushort** Value;  **public uint** Quantity;  **public** MoneySlot Successor;  **protected** MoneySlot(**ushort value**, **uint** quantity, MoneySlot successor) {  Value = **value**;  Quantity = quantity;  Successor = successor;  }  **public void** Show() {  Console.WriteLine("\t${0}\t: X{1}",Value,Quantity);  }  **abstract public void** Withdraw(**uint** amount);  }  } |

|  |  |
| --- | --- |
| **ChainOfResponsibilityATM (MoneySlot100 Class)** | |
| **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** | **using** System;  **namespace** ChainOfResposibilityATM {  **class** MoneySlot100 : MoneySlot {  **public** MoneySlot100(**ushort** quantity, MoneySlot succesor = **null**) :  **base**(100, quantity, succesor) { }  **public override void** Withdraw(**uint** amount) {  **uint** n = amount / 100;//Capture the no of piece needed  **uint** balance = amount % 100;//The balace need to pass to Successor  **if** (n > Quantity) {  balance += (n - Quantity) \* 100;  n = Quantity;  }  **if** (balance > 0) {  **if** (Successor == **null**) **throw new** Exception("Insufficient Money!");  Successor.Withdraw(balance);  }  **if** (n > 0) {  Console.WriteLine("$100\tX{0}", n);  Quantity = Quantity - n;  }  }  }  } |

|  |  |
| --- | --- |
| **ChainOfResponsibilityATM (MoneySlot50 Class)** | |
| **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** | **using** System;  **namespace** ChainOfResposibilityATM {  **class** MoneySlot50 : MoneySlot {  **public** MoneySlot50(**ushort** quantity, MoneySlot succesor = **null**) :  **base**(50, quantity, succesor) { }  **public override void** Withdraw(**uint** amount) {  **uint** n = amount / 50;//Capture the no of piece needed  **uint** balance = amount % 50;//The balace need to pass to Successor  **if** (n > Quantity) {  balance += (n - Quantity) \* 50;  n = Quantity;  }  **if** (balance > 0) {  **if** (Successor == **null**) **throw new** Exception("Insufficient Money!");  Successor.Withdraw(balance);  }  **if** (n > 0) {  Console.WriteLine("$50\tX{0}", n);  Quantity = Quantity - n;  }  }  }  } |

|  |  |
| --- | --- |
| **ChainOfResponsibilityATM (MoneySlot20 Class)** | |
| **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** | **using** System;  **namespace** ChainOfResposibilityATM {  **class** MoneySlot20 : MoneySlot {  **public** MoneySlot20(**ushort** quantity, MoneySlot succesor = **null**) :  **base**(20, quantity, succesor) { }  **public override void** Withdraw(**uint** amount) {  **uint** n = amount / 20;//Capture the no of piece needed  **uint** balance = amount % 20;//The balace need to pass to Successor  **if** (n > Quantity) {  balance += (n - Quantity) \* 20;  n = Quantity;  }  **if** (balance > 0) {  **if** (Successor == **null**) **throw new** Exception("Insufficient Money!");  Successor.Withdraw(balance);  }  **if** (n > 0) {  Console.WriteLine("$20\tX{0}", n);  Quantity = Quantity - n;  }  }  }  } |

|  |  |
| --- | --- |
| **ChainOfResponsibilityATM (MoneySlot5 Class)** | |
| **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** | **using** System;  **namespace** ChainOfResposibilityATM {  **class** MoneySlot5 : MoneySlot {  **public** MoneySlot5(**ushort** quantity, MoneySlot succesor = **null**) :  **base**(5, quantity, succesor) { }  **public override void** Withdraw(**uint** amount) {  **uint** n = amount / 5;//Capture the no of piece needed  **uint** balance = amount % 5;//The balace need to pass to Successor  **if** (n > Quantity) {  balance += (n - Quantity) \* 5;  n = Quantity;  }  **if** (balance > 0) {  **if** (Successor == **null**) **throw new** Exception("Insufficient Money!");  Successor.Withdraw(balance);  }  **if** (n > 0) {  Console.WriteLine("$5\tX{0}", n);  Quantity = Quantity - n;  }  }  }  } |