|  |
| --- |
| [TestMethod]  public void DepositToBankAccount()  {  // #### ARRANGE ####  BankAccount BobsAccount = new BankAccount("Bob", 5000);  // #### ACT ####  BobsAccount.Deposit(5000);  // #### ASSERT ####  Assert.AreEqual(10000, BobsAccount.Balance);  }  [TestMethod]  public void DepositDecimalToBankAccount()  {  // #### ARRANGE ####  BankAccount BobsAccount = new BankAccount("Bob", 5000);  // #### ACT ####  BobsAccount.Deposit(24.57);  // #### ASSERT ####  Assert.AreEqual(5024.57, BobsAccount.Balance);  }  [TestMethod]  public void DepositToLockedBankAccount()  {  // #### ARRANGE ####  BankAccount BobsAccount = new BankAccount("Bob", 5000);  // #### ACT ####  BobsAccount.ChangeLockState();  BobsAccount.Deposit(5000);  // #### ASSERT ####  Assert.AreEqual(5000, BobsAccount.Balance);  }  [TestMethod]  public void WithdrawFromBankAccount()  {  // #### ARRANGE ####  BankAccount BobsAccount = new BankAccount("Bob", 5000);  // #### ACT ####  BobsAccount.Withdraw(5000);  // #### ASSERT ####  Assert.AreEqual(0, BobsAccount.Balance);  }  [TestMethod]  public void WithdrawDecimalFromBankAccount()  {  // #### ARRANGE ####  BankAccount BobsAccount = new BankAccount("Bob", 5000);  // #### ACT ####  BobsAccount.Withdraw(24.57);  // #### ASSERT ####  Assert.AreEqual(4975.43, BobsAccount.Balance);  }  [TestMethod]  public void WithdrawFromLockedBankAccount()  {  // #### ARRANGE ####  BankAccount BobsAccount = new BankAccount("Bob", 5000);  // #### ACT ####  BobsAccount.ChangeLockState();  BobsAccount.Withdraw(5000);  // #### ASSERT ####  Assert.AreEqual(5000, BobsAccount.Balance);  }  [TestMethod]  public void WithdrawTooMuchFromBankAccount()  {  // #### ARRANGE ####  BankAccount BobsAccount = new BankAccount("Bob", 5000);  // #### ACT ####  BobsAccount.Withdraw(10000);  // #### ASSERT ####  Assert.AreEqual(5000, BobsAccount.Balance);  }  [TestMethod]  public void ConstructorAndChangeLockStateBankAccount()  {  // #### ARRANGE ####  BankAccount BobsAccount = new BankAccount("Bob", 5000, true);  // #### ACT ####  BobsAccount.ChangeLockState();  BobsAccount.Deposit(42);  // #### ASSERT ####  Assert.AreEqual(5042, BobsAccount.Balance);  }  [TestMethod]  public void OverrideToStringBankAccount()  {  // #### ARRANGE ####  BankAccount BobsAccount = new BankAccount(5000);  BobsAccount.Name = "Bob";  // #### ACT ####  BobsAccount.ChangeLockState();  BobsAccount.Deposit(100);  BobsAccount.ChangeLockState();  BobsAccount.Deposit(42);  // #### ASSERT ####  Assert.AreEqual("Name: Bob, Balance: 5042", BobsAccount.ToString());  }  [TestMethod]  public void OverrideToStringNoNameBankAccount()  {  // #### ARRANGE ####  BankAccount NobodysAccount = new BankAccount(5000);  // #### ACT ####  NobodysAccount.Deposit(100);  NobodysAccount.ChangeLockState();  NobodysAccount.Deposit(42);  // #### ASSERT ####  Assert.AreEqual("Name: , Balance: 5100", NobodysAccount.ToString());  } |