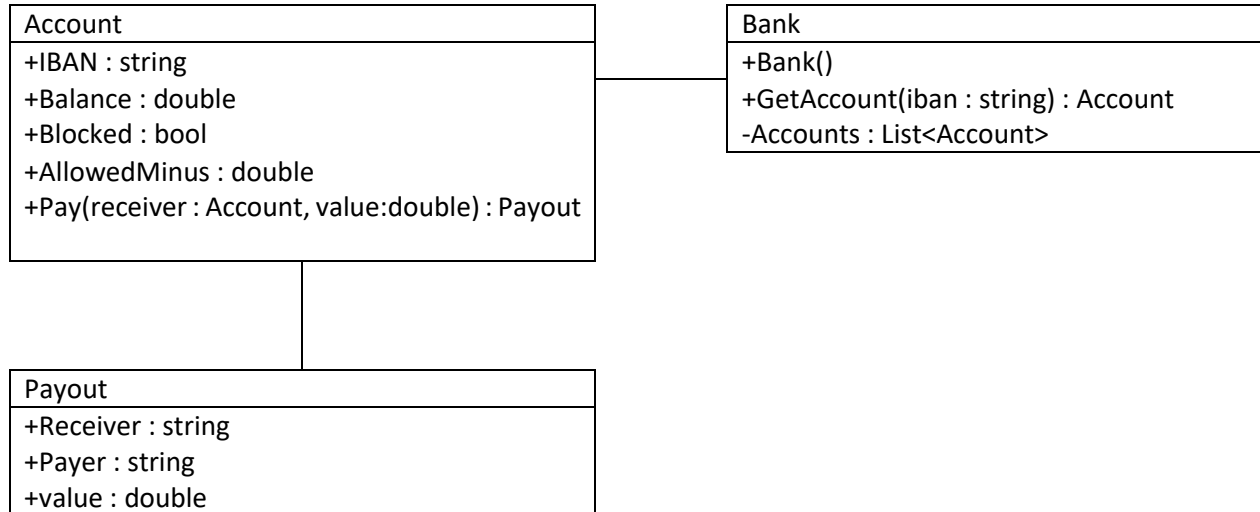


Please test the existing solution Finances with new Unit Test project and unit tests for classes Invoice and Banks. Each of the bellow stated scenarios should be covered with separate test method. Tests with red color will initially fail which means that some corrections in the code should be made.



```

Accounts.Add(new Account { IBAN = "HR11", Balance = 100000, Blocked = false });
Accounts.Add(new Account { IBAN = "HR22", Balance = 50000, Blocked = false });
Accounts.Add(new Account { IBAN = "HR33", Balance = 12000, Blocked = false });
Accounts.Add(new Account { IBAN = "HR44", Balance = 36000, Blocked = true });
Accounts.Add(new Account { IBAN = "HR55", Balance = 8000, Blocked = false });
Accounts.Add(new Account { IBAN = "HR66", Balance = 2000, Blocked = false, AllowedMinus = 3000 });
  
```

Scenarios to test

1. If method **GetAccount** in class **Bank** is called with IBAN **HR12** (unknown account), method should return **null**.
2. If method **GetAccount** in class **Bank** is called with IBAN **HR11** (known account), method should return object with IBAN property equals **HR11**.
3. If method **GetAccount** in class **Bank** is called with IBAN **HR44** (blocked account), method should throw exception **BankAccountBlockedException**.
4. If method **Pay** in class **Account** is called trying to transfer **30 000 kn** from account **HR11** to account **HR21** (unknown account), method should throw exception **TransactionFailedException**.
5. If method **Pay** in class **Account** is called trying to transfer **30 000 kn** from account **HR11** to account **HR22**, method should be able to complete the transaction, and the account **HR11** should have the balance of **70 000 kn**, and account **HR22** should have the balance of **80 000 kn**.