# ATM MACHINE

## MURAT CAN ALAN – BORAN KUZUKIRAN
### 201928003 - 202028301

start

more account object? — no → print "User does not have an account." → return -1 → end

yes ↓

account.number==number? — no → (loop back to more account object?)

yes ↓

account.isAdmin==1 — no → print "User is client." → return 0 → end

yes ↓

print "User is admin." → return 1 → end
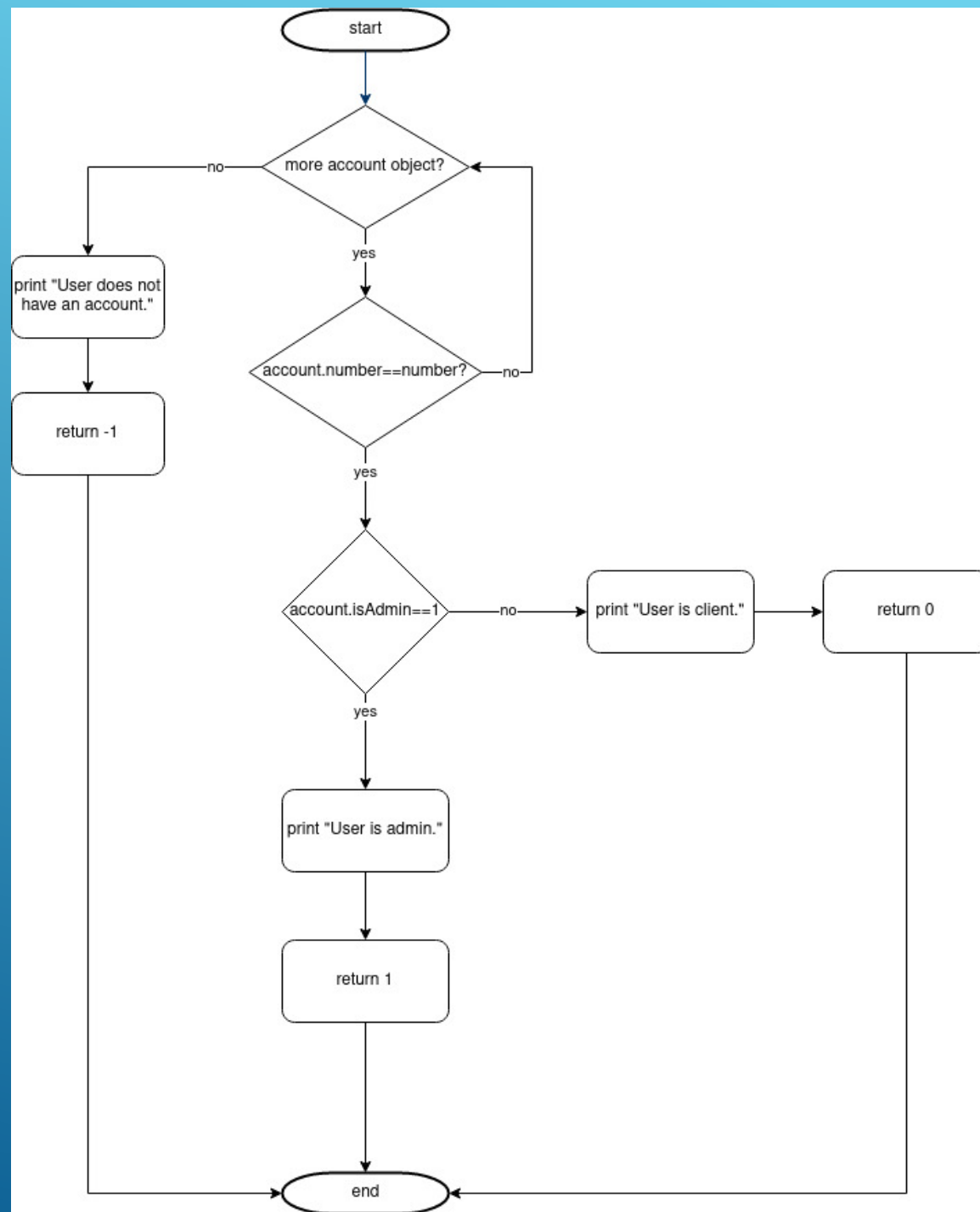
```java
// BankDatabase.java                                    Account.java

25
26          // retrieve Account object containing specified account number
27
28          public int getAccountType(int accountNumber) {
29              for (Account currentAccount : accounts) {
30                  if (currentAccount.getAccountNumber() == accountNumber) {
31                      if (currentAccount.getIsAdmin() == 1) {
32                          System.out.println("User is admin.");
33                          return 1;
34                      }
35                      else {
36                          System.out.println("User is a customer.");
37                          return 0;
38                      }
39                  }
40              }
41              System.out.println("Not a user.");
42              return -1;
43          }
44
```

```java
// BankDatabaseTest.java

190    @Test
191    public void getAccountType_WBTest_1(){
192        BankDatabase testBankDatabase = new BankDatabase();
193        int accountNumber = 98765;
194        assertTrue( condition: 0 == testBankDatabase.getIsAdmin(accountNumber));
195    }
196
197    @Test
198    public void getAccountType_WBTest_2(){
199        BankDatabase testBankDatabase = new BankDatabase();
200        int accountNumber = 99999;
201        assertTrue( condition: 1 == testBankDatabase.getIsAdmin(accountNumber));
202    }
203
204    @Test
205    public void getAccountType_WBTest_3(){
206        BankDatabase testBankDatabase = new BankDatabase();
207        int accountNumber = 13568;
208        assertTrue( condition: -1 == testBankDatabase.getIsAdmin(accountNumber));
209    }
```

```java
1 usage
public int getAccountType_mutant1(int accountNumber) {
    for (Account currentAccount : accounts) {
        if (currentAccount.getAccountNumber() != accountNumber) { /
            if (currentAccount.getIsAdmin() == 1) {
                System.out.println("User is admin.");
                return 1;
            }
            else {
                System.out.println("User is a customer.");
                return 0;
            }
        }
    }
    System.out.println("Not a user.");
    return -1;
}


@Test
public void getAccountType_mutant_1(){
    BankDatabase testBankDatabase = new BankDatabase();
    int accountNumber = 98765;
    assertTrue( condition: 0 == testBankDatabase.getAccountType_mutant1(accountNumber));
}
```

```java
1 usage
public int getAccountType_mutant2(int accountNumber) {
    for (Account currentAccount : accounts) {
        if (currentAccount.getAccountNumber() == accountNumber) {
            if (currentAccount.getIsAdmin() != 1) { //*
                System.out.println("User is admin.");
                return 1;
            }
            else {
                System.out.println("User is a customer.");
                return 0;
            }
        }
    }
    System.out.println("Not a user.");
    return -1;
}

@Test
public void getAccountType_mutant_2(){
    BankDatabase testBankDatabase = new BankDatabase();
    int accountNumber = 99999;
    assertTrue( condition: 1 == testBankDatabase.getAccountType_mutant2(accountNumber));
}
```

```java
public int getAccountType_mutant3(int accountNumber) {
    for (Account currentAccount : accounts) {
        if (currentAccount.getAccountNumber() == accountNumber) {
            if (currentAccount.getIsAdmin() == 1) {
                System.out.println("User is admin.");
                return 3; //*
            }
            else {
                System.out.println("User is a customer.");
                return 0;
            }
        }
    }
    System.out.println("Not a user.");
    return -1;
}


@Test
public void getAccountType_mutant_3(){
    BankDatabase testBankDatabase = new BankDatabase();
    int accountNumber = 13568;
    assertTrue( condition: -1 == testBankDatabase.getAccountType_mutant3(accountNumber));
}
```

```java
1 usage
public int getAccountType_mutant4(int accountNumber) {
    for (Account currentAccount : accounts) {
        if (currentAccount.getAccountNumber() == accountNumber) {
            if (currentAccount.getIsAdmin() == 1) {
                System.out.println("User is admin.");
                return 3;
            }
            else {
                System.out.println("User is a customer.");
                return 0;
            }
        }
    }
    System.out.println("Not a user.");
    return 0;
}

@Test
public void getAccountType_mutant_4(){
    BankDatabase testBankDatabase = new BankDatabase();
    int accountNumber = 13568;
    assertTrue( condition: -1 == testBankDatabase.getAccountType_mutant4(accountNumber));
}
```

```java
    @Test
    public void test_DT_1(){
        BankDatabase testBankDatabase = new BankDatabase();
        int accountNumber = 12345;
        assertTrue( condition: 0 == testBankDatabase.getAccountType(accountNumber));
    }


    @Test
    public void test_DT_2(){
        BankDatabase testBankDatabase = new BankDatabase();
        int accountNumber = 99999;
        assertTrue( condition: 1 == testBankDatabase.getAccountType(accountNumber));
    }


    @Test
    public void test_DT_3(){
        BankDatabase testBankDatabase = new BankDatabase();
        int accountNumber = 76268;
        assertTrue( condition: -1 == testBankDatabase.getAccountType(accountNumber));
    }
```

# DECISION TABLE

| Conditions | Case 1 | Case 2 | Case 3 |
|---|---|---|---|
| Account number matched. | N | Y | Y |
| Account owner is an admin. | - | N | Y |
| Account owner is a client. | - | Y | N |
| Actions | | | |
| a1: Return -1 | X | | |
| a2: Return 0 | | X | |
| a3: Return 1 | | | X |