

```
Debug Server
Debugger Console →
Eric Andy has 2 bank accounts.
Eric Andy has:
1) ACME Credit Union : checking -- 112233445
2) ACME Credit Union : savings -- 554433221
This contact (Eric Andy) does not have an account at National Bank
William Andy has 2 bank accounts.
William Andy has:
1) ACME Credit Union : checking -- 345678900
2) National Bank : checking -- 232323566
This contact (William Andy) has an account at National Bank
```

4: Run 5: Debug 6: TODO 7: Properties 8: Terminal 9: Messages 10: Codegen



## 8.4 Lab Solution: Coding with Gosu



### 8.4.1 Write it down

1. Question: Navigate to the Summary page of the following company: Burlingame Saab. On the Summary page, what is the Public ID for Burlingame Saab?

ab:78

2. Question: Navigate to the Summary page of the following doctor: Rebecca Stevens. On the Summary page, what is the Public ID for Rebecca Stevens?

ab:70

## 8.4.2 Configuration



**Note:** there are multiple correct solutions for this exercise.

### 1. Open Gosu Scratchpad in Studio

- Verify that the *Run in Debug Process* icon is available in Gosu Scratchpad.

### 2. Write Gosu code

```
uses trainingapp.base.QueryUtil

//**** START - my first Gosu program ****

//before running the code, replace this with the contact's PublicID
var publicID = "ReplaceThisWithThePublicIDOfTheContact"

var contact = QueryUtil.findContact(publicID)

if(contact != null) { //Requirement 1

    //Requirement 1/A
    print(contact.DisplayName + " was created " + contact.CreateTime + ".") 

    //Requirement 1/B
    if(contact.PrimaryAddress != null) {
        print("Primary address state is " + contact.PrimaryAddress.State + ".")
    }

    //Requirement 1/C
    var isStrategicPartner = contact.IsStrategicPartner_Ext ? " and is a strategic partner" : " and is NOT a strategic partner."
    print("The contact is of the subtype " + contact.Subtype.DisplayName + isStrategicPartner)

    //Requirement 1/D
    if(contact typeis ABDoc) {
        var doctorCategory = contact.DoctorCategory != null ? contact.DoctorCategory.DisplayName : "NOT set"
        var doctorSpecialty = contact.DoctorSpecialty != null ? contact.DoctorSpecialty.DisplayName : "NOT set"

        print("Doctor category is " + doctorCategory + " and doctor specialty is " + doctorSpecialty + ".")
    } else { //Requirement 1/E
        print("Contact is NOT of the type " + ABDoc.Type.DisplayName + ".")
    }

} else { //Requirement 2/A
    print("No contact found for PublicID: " + publicID + ".")
}

//**** END - my first Gosu program ****
```

## 8.5 Lab Solution: Working with arrays



### Solution

#### 8.5.1 Write it down

1. Question: Navigate to the Summary page of the following contact: Eric Andy. On the Summary page, what is the Public ID for Eric Andy? Go to the Bank accounts card on the Details page and verify that the contact has 2 bank accounts: ACME Credit Union – checking and savings accounts.

ab:98

2. Question: Navigate to the Summary page of the following contact: William Andy. On the Summary page, what is the Public ID for William Andy? Go to the Bank accounts card on the Details page and verify that the contact has 2 bank accounts: SUCCEED Credit Union – checking and National Bank - savings accounts.

ab:5

#### 8.5.2 Configuration



**Note:** there are multiple correct solutions for this exercise.

##### 1. Open Gosu Scratchpad in Studio

- a) Verify that the *Run in Debug Process* icon is available in Gosu Scratchpad.

##### 2. Write Gosu code

```
uses trainingapp.base.QueryUtil

var personIDs = new String[] {"98", "5"}

for (personID in personIDs) {
    var person = QueryUtil.findPerson(personID)

    if (person != null) {
        print(person.DisplayName + " has " + person.BankAccounts.length + " bank accounts.")

        if (person.BankAccounts.length > 0) {
            print(person.DisplayName + " has: ")
        }
    }
}
```

```
for (bankAccount in person.BankAccounts index i) {
    print(i + 1 + " " + bankAccount.BankName + " : " + bankAccount.AccountType + " -- " +
bankAccount.AccountNumber)
}
}

var bankName = "National Bank"

if (person.BankAccounts.hasMatch(\account -> account.BankName == bankName)) {
    print("This contact (" +person.DisplayName + ") has an account at " + bankName)
} else {
    print("This contact (" +person.DisplayName + ") does not have an account at " + bankName)
}

} else {
    print("No person found for PublicID: " + personID)
}
}
```