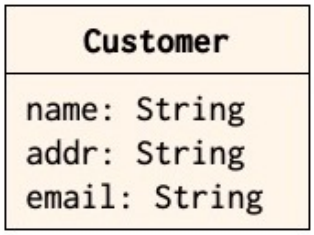
3.3 Introducing a Customer class

Description

Customers of a bank and thereby potential owners of an account will usually have a profile with more than just the name (like “John Smith”). In addition, the same customer may have more than one account in the bank, and there may be other situations where the profile of a customer is needed.

With classes of objects, the way to model this is to introduce the class `Customer` with attributes representing the properties of a customer profile.

```
class Customer(name: var String):
   addr: var String
   email: var String
```



We may declare a `Customer` object as follows:

```
JohnSmithProfile: obj Customer("John Smith")
```

We may subsequently assign values to the data-items `addr` and `email` of `JohnSmith`:

```
JohnSmithProfile.addr := "1011 Eagle Road, Moon Town, Utopia"
JohnSmithProfile.email := "john.smith@utopia.org"
```

We would like to be able to refer to the `JohnSmithProfile`-object from the `Account` object. In order to handle this, we introduce the mechanism of a *variable reference*, which may be declared as follows:

```
aCustomer: ref Customer
```

The variable, `aCustomer` may refer to different `Customer`-objects during program execution. While the data-items we have seen so far (`balance` and `interestRate`) may hold different `float` values representing different values of `balance` and `interestRate`, the data-item `aCustomer` is a reference data-item that may refer to `Customer` objects.

We may e.g. set `aCustomer` to refer to the `JohnSmithProfile`-object using an assignment statement:

```
aCustomer := JohnSmithProfile
```

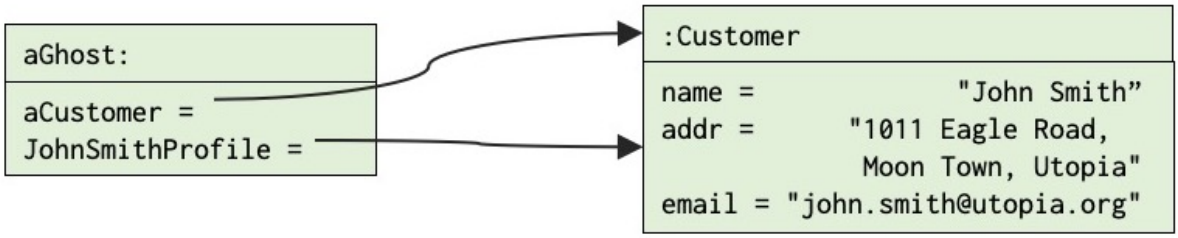
In order to illustrate the above, we place the code fragments in a *ghost* object, `aGhost`:

The term *ghost object* is an object that is a placeholder for items used in an example – they do not make sense by themself.

```
aGhost: obj
   aCustomer: ref Customer
   JohnSmithProfile: obj Customer("John Smith")
   JohnSmithProfile.addr := "1011 Eagle Road, Moon Town, Utopia"
   JohnSmithProfile.email := "john.smith@utopia.org"
```

```
   aCustomer := JohnSmithProfile
-->
```

The snapshot shows the situation after the last assignment as marked by the red arrow (–>):



The class `Account` is correspondingly changed so that the `owner` of an `Account` is a reference to a `Customer` object:

```
class Account(owner: ref Customer):
   balance: var float
   interestRate: var float
   addInterest:
      balance := balance + (balance * interestRate) / 100
   deposit(amount: var float):
      balance := balance + amount
   withdraw(amount: var float) -> newB: var float:
      balance:= balance - amount
      newB := balance
```

We now use a reference variable, `owner`, to represent the owner of class `Account`:

```
owner: ref Customer
```

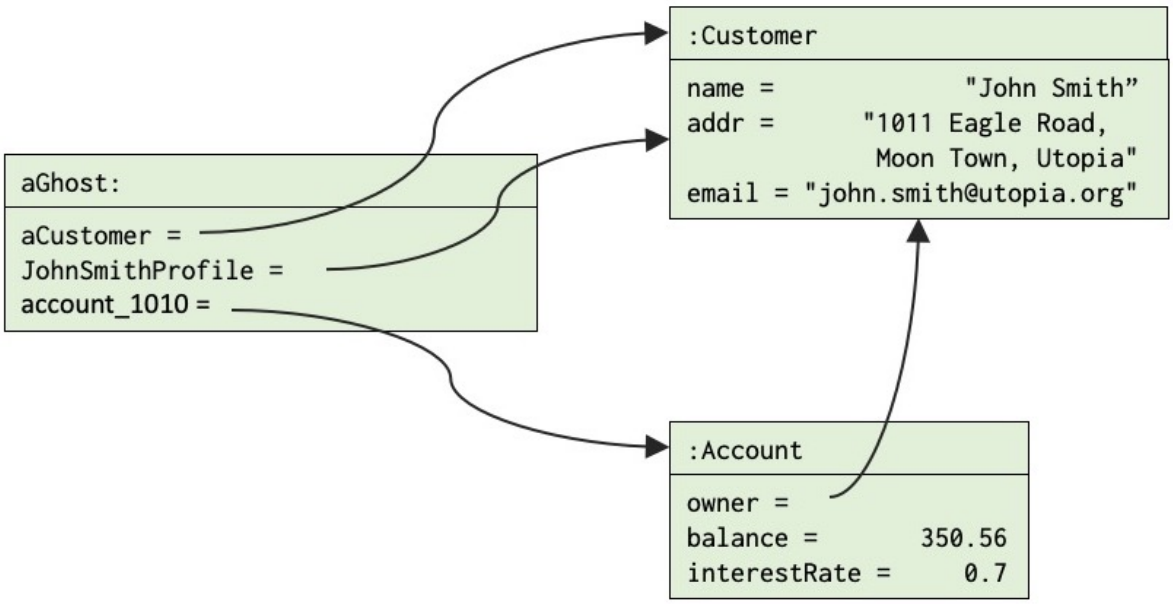
An `Account`-object now has a reference to a `Customer` object representing the owner of the account. We may create an `Account` object with the `Owner` object `JonhSmithsProfile` as the actual parameter as follows:

```
account_1010: obj Account(JohnSmithsProfile)
```

Again, we place the above code in a ghost object to illustrate the situation where `aCustomer` and `account_1010.owner` both refer to the same `Customer` object. The following snapshot shows the situation at the end of execution of `aClerk`:

```
aGhost: obj
   aCustomer: ref Customer
   JohnSmithProfile: obj Customer("John Smith")
   account_1010: obj Account(JohnSmithsProfile)
   JohnSmithProfile.addr := "1011 Eagle Road, Moon Town, Utopia"
   JohnSmithProfile.email := "john.smith@utopia.org"
   aCustomer := JohnSmithProfile
-->
```

As can be seen, both `aCustomer` and the `owner` attribute of the `Account` object refers to the same `Customer` object.

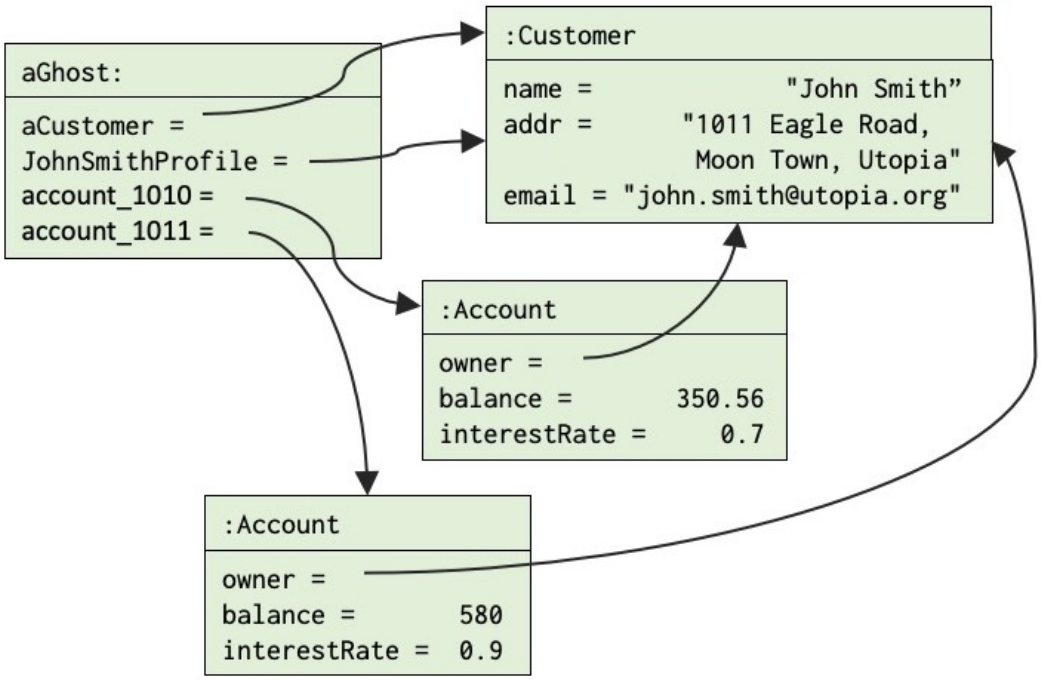


Reference owner from Account object to Customer object

We create yet another `Account` for John Smith using:

```
account_1011: obj Account(JohnSmithsProfile)
```

Assuming we add the above declaration to our ghost object, this situation may be illustrated by the following snapshot:



Two owner references from Account objects to Customer object

In a paper based bank the fact that two or more accounts are owned by the same customer is represented by all these accounts having the same name on the owner field of the account sheet. In a computerised bank system this would also be possible, however, with objects we have seen that there is an alternative where the name (and other types of information) of a customer is represented by attributes of an object and thereby maintained in one place only.

The details of reference assignment are described in section .