

8.1 Subclasses

Description

Let us start by looking at a possible description of two different kinds of accounts, savings accounts and credit accounts. We first show a class representing a savings account. Some of the attributes are the same as in class `Account` from the previous chapters. In addition, new attributes and statements are added – in the example, these are highlighted:

```
class SavingsAccount(owner: ref Customer):
  balance: var float
  interest: var float
  releaseDate: var Date
  addInterest:
    balance := balance + (balance * interestRate) / 100
  deposit(amount: var float):
    balance := balance + amount
  withdraw(amount: var float) -> newB: var float:
    if (today > releaseDate) :then
      balance := balance - amount
    :else
      console.print("It is not possible to withdraw")
      newB := balance
  newReleaseDate(newDate: var Date, newInterest: var float):
    releaseDate := newDate
    interest := newInterest
```

Compared to class `Account` from the previous chapters, we have added a data-item `releaseDate` of type `Date`. It represents the date that the customer have agreed to bind his/her money to in order to get a higher interest than on just an account.

The method `withdraw` differs from the one in class `Account`, since it is not possible to withdraw money before the release date.

We have also added a method `newReleaseDate` with parameters `newDate` and `newInterest`. This method may set a new binding period and a new interest rate for a given a given savings account.

Next let us look at similar description of a class representing a credit account.

```
class CreditAccount(owner: ref Customer):
  balance: var float
  interest: var float
  maxCredit: var float
  addInterest:
    balance := balance + (balance * interestRate) / 100
  deposit(amount: var float):
    balance := balance + amount
  withdraw(amount: var
float):
    if (-balance < maxCredit) :then
      balance := balance - amount
    :else
      console.print("Not possible to withdraw beyond max credit")
  changeCredit(newMax: var float, newInterest: var float):
    maxCredit := newCredit
    interest := newInterest
```

For a credit account it is possible to overdraw money up to a certain limit defined as the maximum limit. We have added a data-item `maxCredit`, that represents the maximum credit limit of the account.

We also have a new method `changeCredit`, which can set a new value for the maximum credit and a new interest rate.

The `withdraw` method is special in the sense that there is a test for not withdrawing beyond the `maxLimit`. We assume that `maxLimit` is a positive float number. If `balance` is positive the customer has deposited money on the account. If `balance` is negative, the customer has loaned money from the bank.

As can be seen, the two types of accounts have some attributes in common like `owner`, `balance`, `interest`, `addInterest`, and `deposit`.

In addition, each of them have attributes that are specific for the kind of account they represent. For class `SavingsAccount` these are `releaseDate` and `newBindingPeriod`. For class `CurrentAccount` these are: `maxCredit` and `changeCredit`.

In the next section, we show how to define the common attributes of the two classes in a class that may be used to define these classes.