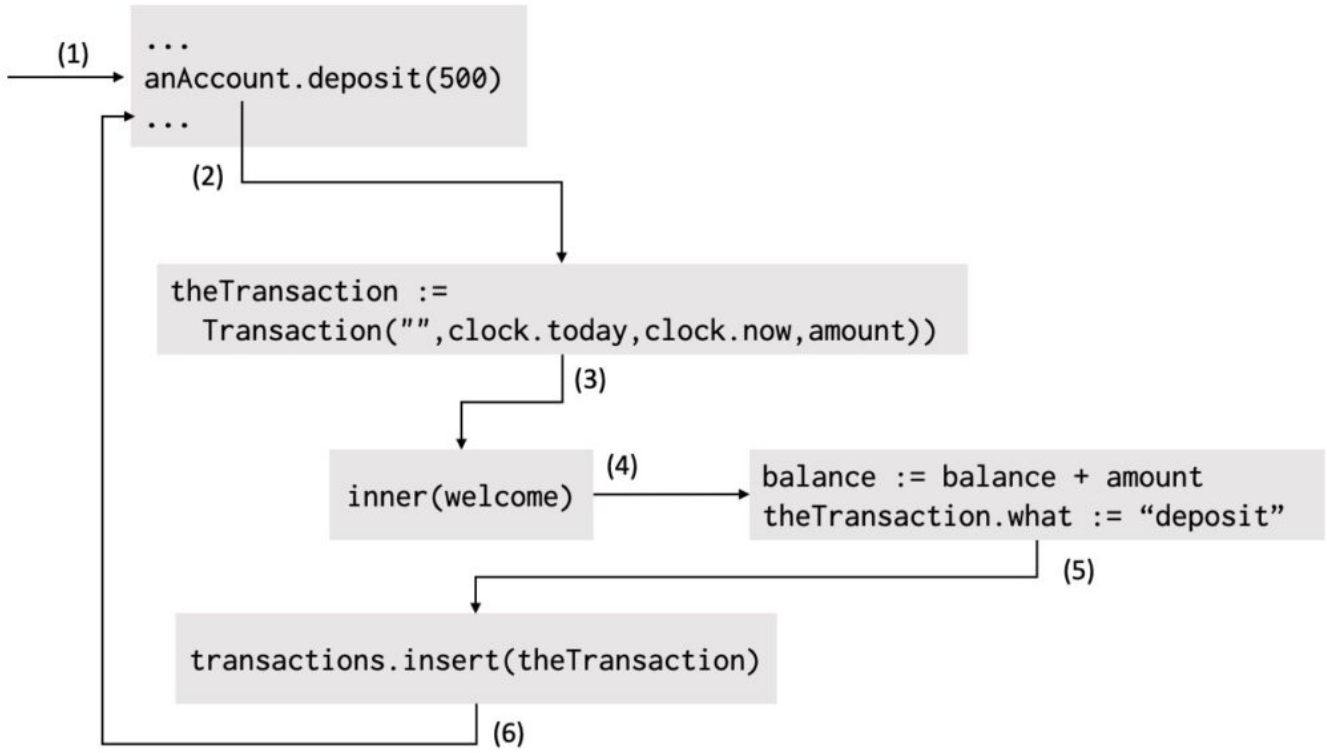
8.2.1 The inner statement

The `inner`-statement is central for defining methods that may be used as supermethods of other methods. We take a closer look at what happens when executing `anAccount.deposit(500)`. This implies the following steps:

- Execution of anAccount.deposit(500) (1).
- Execution starts by the executing the items in `transact`, which implies execution of `theTransaction := Transaction(...)` (2).
- The `inner(transact)` is executed (3), which implies that the items in `deposit` are executed.
- In addition to `balance := balance + amount`, `theTransaction` is marked as a deposit (4).
- After execution of `deposit`, control returns to `transact` (5).
- Then `transactions.insert(theTransaction)` is executed (6).

The figure below illustrates the above steps of execution – the numbers in the figure corresponds to the numbers in the text.



Sequence of actions for the call "anAccount.deposit(500)"

The above example show how invocation of `deposit` takes place – invocation of `withdraw` takes place in a similar way.

Nested inner

If `M` is a method, `inner(M)` may be placed in any nested object-descriptor within `M`.

Suppose we want to inspect all transactions on an `Account` that involves a specific amount. This may e.g. be the case if you want to find out when and if you have made such a transaction.

We extend the example in section where we add a `find` method that goes through the `transactions` of a given `Account`.

```
class Account(owner: ref Customer):
   -"-
   transactions: obj Set(Transaction)
   -"-
   find(amount: var float):
      current: ref Transaction
      transaction.scan
            if current = amount :then
               this(find).current := current
               inner(find)
```

The `find` method has a nested `inner(find)`. It is placed in the singular method descriptor `transaction.scan{...},` which is nested in the method descriptor for `find`.

Note that `find` has a variable `current` that has the same name as the variable `current` in `scan`. We therefore use `this(find).current` to denote the `current` declared in `find`.

We may use `find` in the following way:

```
anAccount: ref Account
...
anAccount.find(450)
   console.print("A " + current.what + " was made on " + current.whichDate)
```

Execution of `inner(find)` implies that statements in submethods of `find` are executed. In this case, the `console.print` statement will thus be executed for each transaction where `amount` is equal to 450 – if any.

Inner in classes

`Inner` may also be used to combine the statements of a superclass and a subclass. We give examples of this in sections .