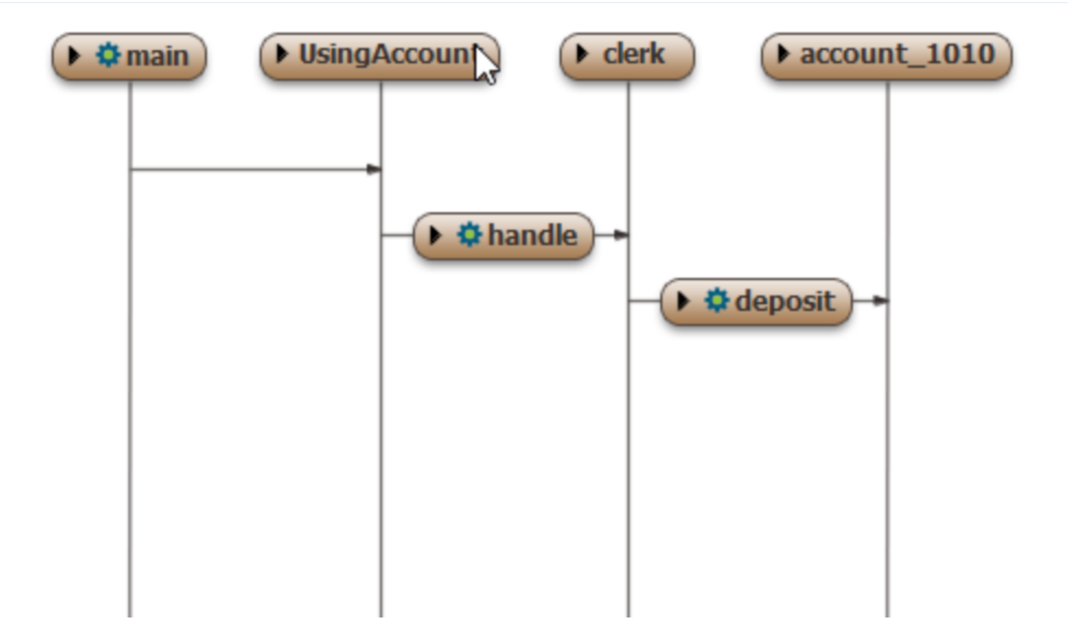
2.8XX Program execution – other stuff

Description

```
usingAccount: obj
   aClerk: obj
      handle:
         newBalance := account_1010.deposit(100)
         -"-
   account_1010: obj
      -"-
      deposit(amount: var float):
-->      balance := balance + amount
      -"-
   aClerk.handle
```

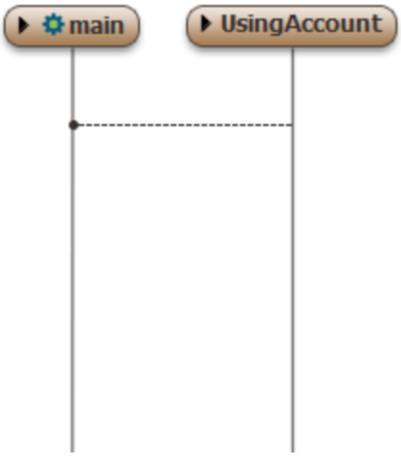


Generation of objects – OLD – SKIP

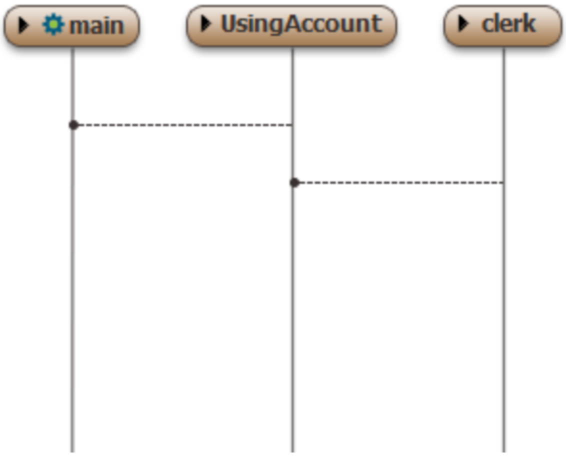
Jeg lavede først denne her udgave men lavede så den ovenfor, som jeg synes er bedre. Så skip dette her afsnit.

In the next snapshots, we show the generation of the objects in the diagram.

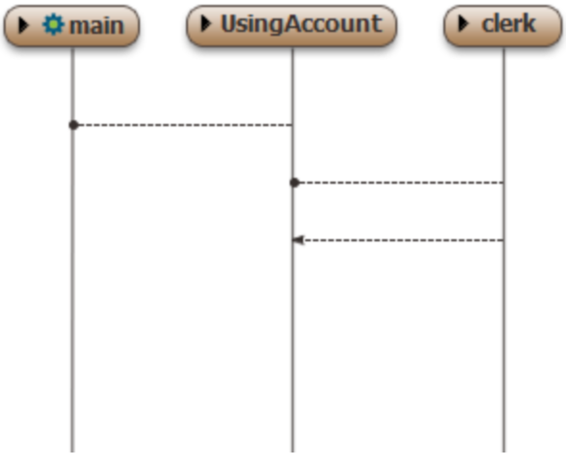
- First `usingAccount` is generated from `main`.
- Then clerk is generated from `usingAccount`; clerk is is declared using **`obj`**, which means that its is generated as part of the generation of `usingAccount`.
- Then generation of clerk returns to `usingAccount` – the dotted arrow represents a return to the caller.
- The account_1010 is generated from `usingAccount` followed by a return to `usingAccount`
- Finally the generation of `usingAccount` returns to `main` and generation of `usingAccount` with local objects `clerk` and `account_1010` is completed.



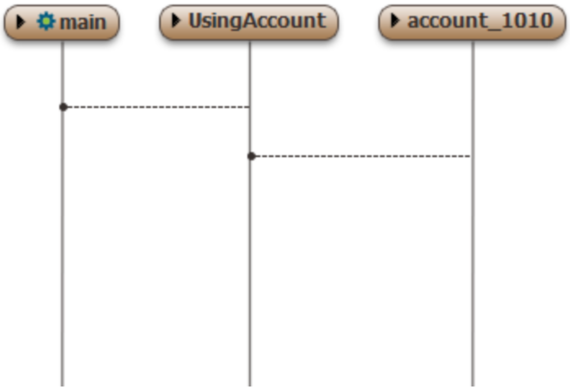
Snapshot after `main` has generated `usingAccount`.

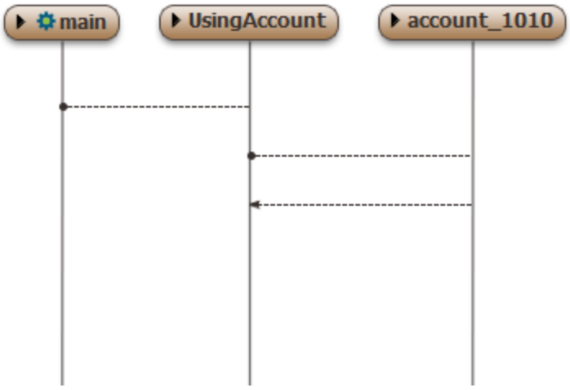


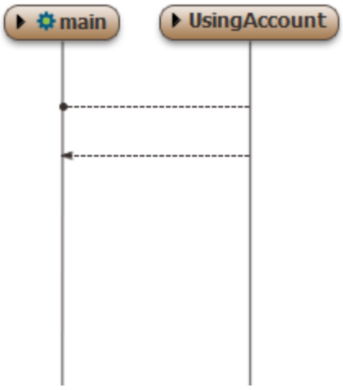
Snapshot after `usingAccount` has generated `clerk`.



Snapshot after return of generation of `clerk`.







Snapshot after `usingAccount` has generated `account_1010`.

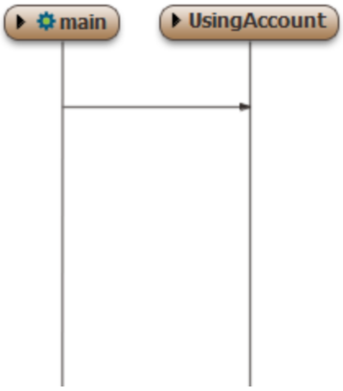
Snapshot after return of generation of `account_1010`.

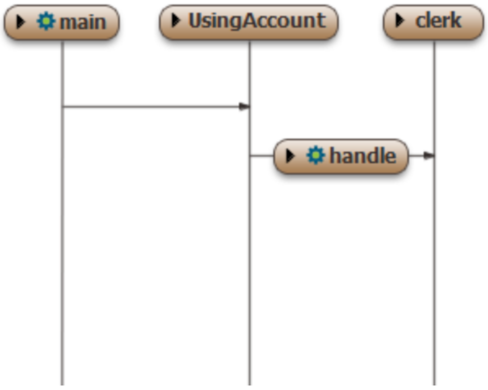
Snapshot after return of generation of `usingAccount`.

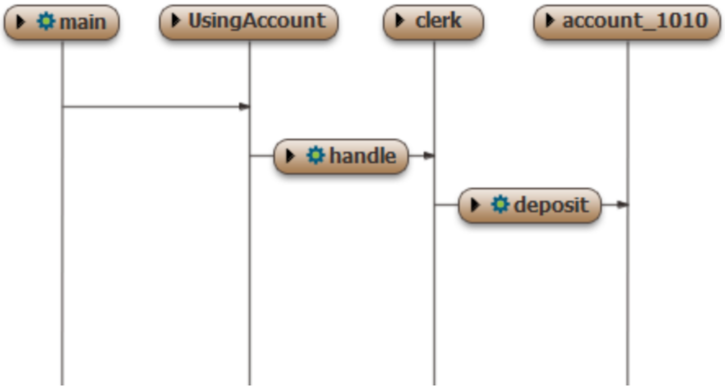
Execution of statements -OLD

The next snapshots shows how the execution of statements in `usingAccount` takes place. As mentioned before, the linewith arrow with no method from `main` to `usingAccount` illustrates execution of the statements in `usingAccount`. Thesnapshot is before invocation of `clerk.handle`.

The next snapshot show how execution of handle takes place. First `account.deposit` is invoked followed by yet another `deposit`; invocation of `addInterest`; invocation of `withdraw` and finally invocation of `print`.



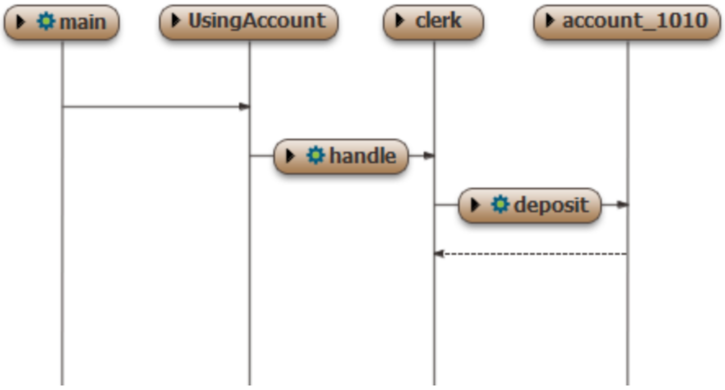




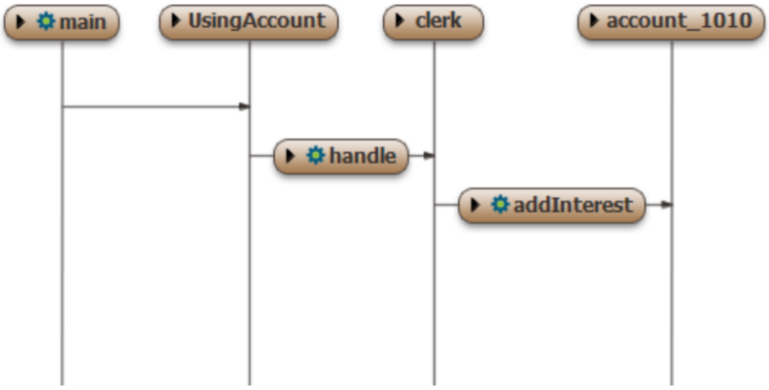
Start of execution of statements in `usingAccount`; before `aClerk.handle`.

Snapshot when `clerk.handle` has been invoiced; before first statement in `handle`.

Snapshot when `account_1010.deposit` has been invoked.

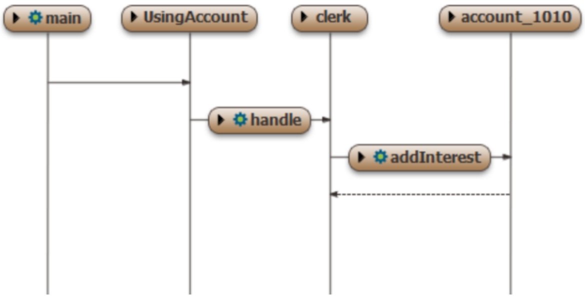


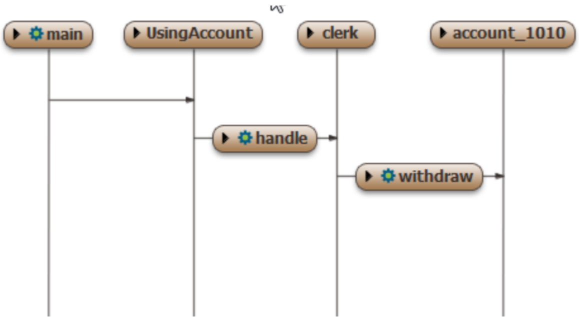
One more deposit

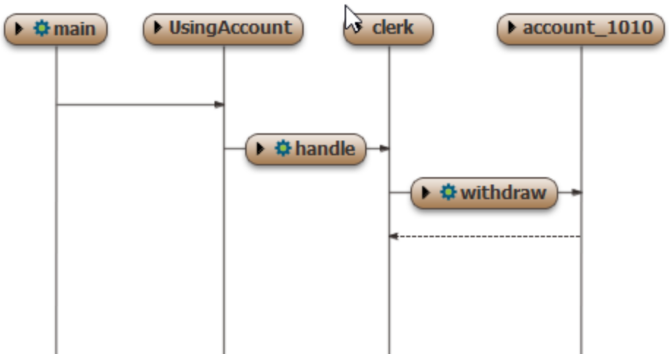


Return from `deposit`.

Invocation of `addInterest`.



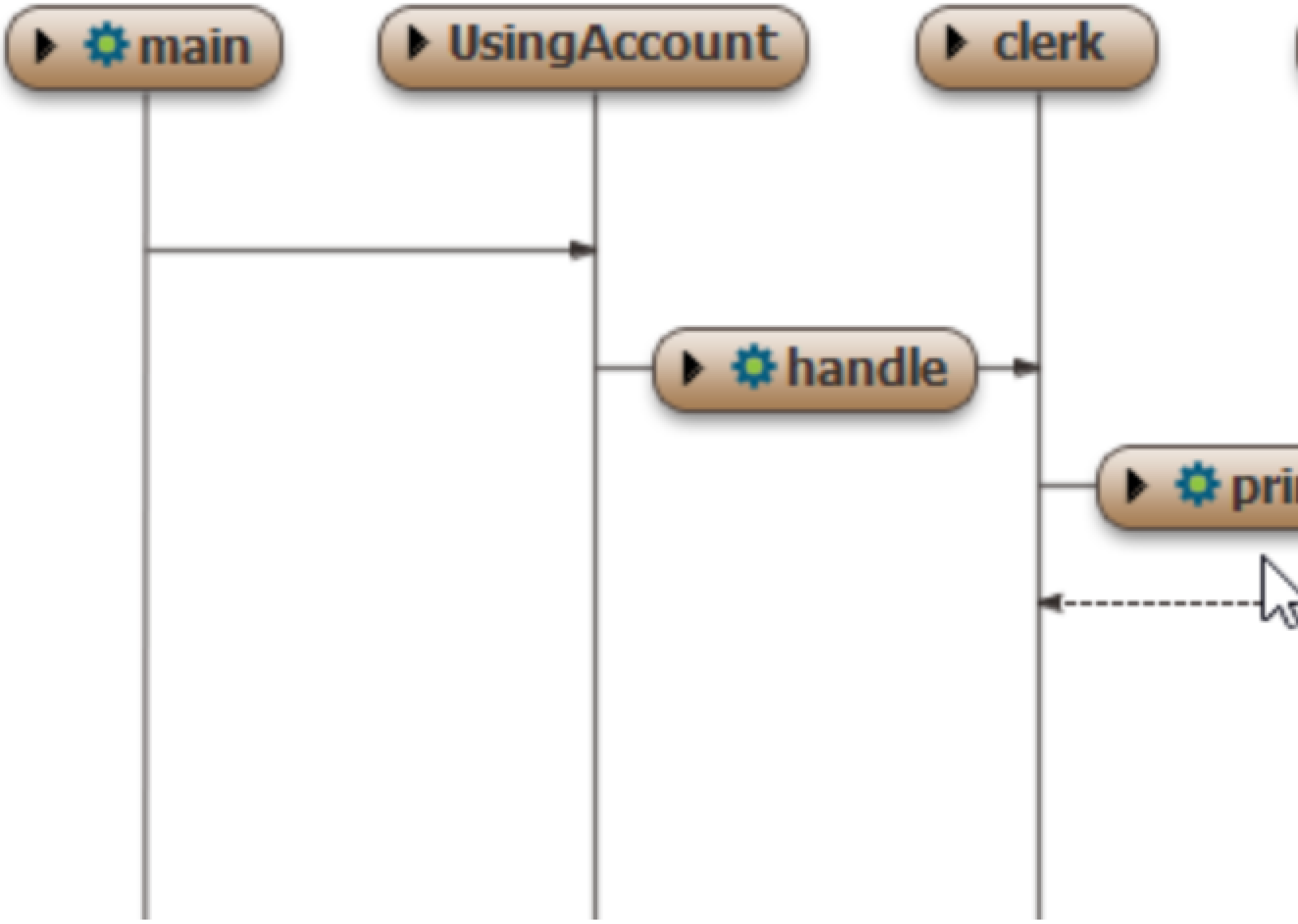


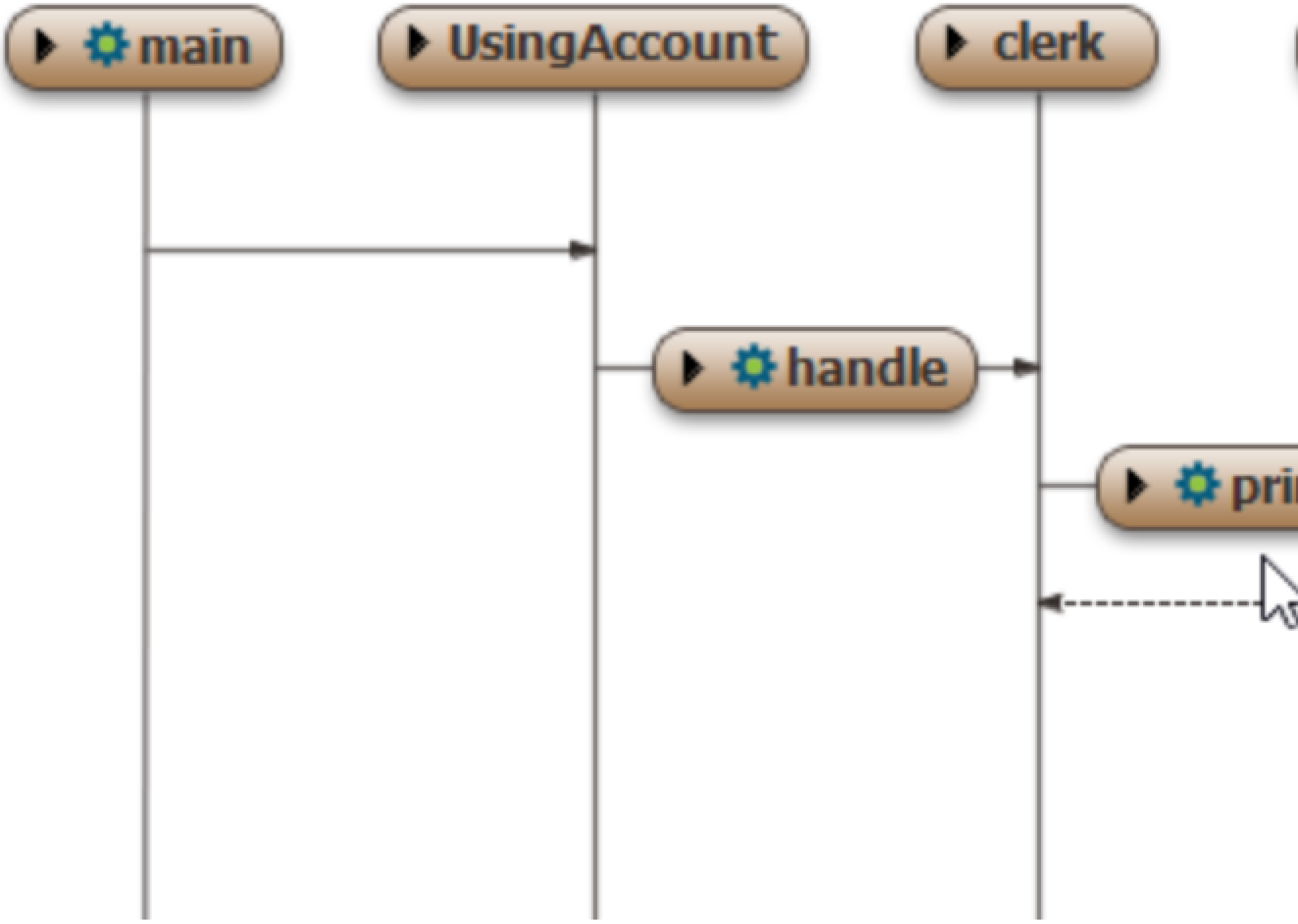


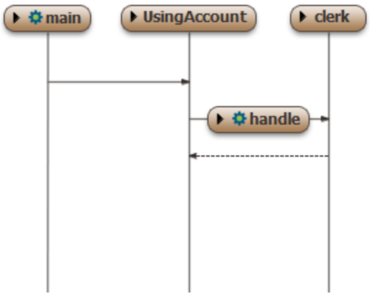
Return from `addInterest`.

Invocation of `withdraw`.

Return from `withdraw`.


main
UsingAccount
clerk
handle
pri





Invocation of print

Return from print

Return from handle.