FWD - Bug #3323

Incorrect handling of variables with similar names in some cases

08/12/2017 06:42 AM - Jaroslaw Haziak

Status:	New	**Start date:**	08/12/2017
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD	**version:**	

Description

History

#1 - 08/12/2017 06:45 AM - Jaroslaw Haziak

Example:

```
DEF INPUT PARAM inParam AS CHAR NO-UNDO.
DEF OUTPUT PARAM ouResult AS CHAR NO-UNDO.

DEF VAR i AS INT NO-UNDO.

PROCEDURE test:
END.

DEF VAR i_ AS INT NO-UNDO.

ouResult = inParam + STRING(i) + STRING(i_).
```

This code is translated to non-compilable Java code because of duplication of variable names ('i'):

```
   public void execute(final character _inParam, final character ouResult)
   {
      character inParam = TypeFactory.initInput(_inParam);
      integer i = TypeFactory.integer();
      integer i = TypeFactory.integer();

      externalProcedure(TestProc3.this, new Block((Init) () -> 
      {
         TypeFactory.initOutput(ouResult);
      }, 
      (Body) () -> 
      {
         ouResult.assign(concat(inParam, valueOf(i), valueOf(i)));
      }));
   }

   public void test()
   {
      internalProcedure(new Block());
   }
```

#2 - 08/12/2017 06:49 AM - Jaroslaw Haziak

But when we move declaration of variable 'i_' before the definition of internal procedure 'test' then the translated Java code is OK:

```
   public void execute(final character _inParam, final character ouResult)
   {
      character inParam = TypeFactory.initInput(_inParam);
      integer i = TypeFactory.integer();
      integer i_2 = TypeFactory.integer();

      externalProcedure(TestProc3.this, new Block((Init) () -> 
      {
         TypeFactory.initOutput(ouResult);
      }, 
      (Body) () -> 
      {
         ouResult.assign(concat(inParam, valueOf(i), valueOf(i_2)));
      }));
   }

   public void test()
   {
      internalProcedure(new Block());
   }
```