Exercise 5: table class with custom variable types

ex05/ex05_Table_PK.txt in the metacode project map

execute the code generation after each step and watch the result in the destination project

We take the same file as in exercise 4, and rename ex04 to ex05.

We also create a table class where we use our primary key classes.

In both classes we now define custom variable types

- java.lang.String STRING
- java.sql.Date DATE
- java.sql.Time TIME

and we introduce a few metacode constraints

- iffieldtype
- java.lang.String
- java.sql.Date
- java.sql.Time
- other

primary key file

code changes in ex05_Table_PK.txt (copy from ex04_Table_PK.txt

```
ex05:Table:PK
```

variable declaration

iffieldtype selects the code block that is equal to the fieldtype. If not, other code block is used

```
:repeatpkfields:
:notfk:
:iffieldtype:
:java.lang.String:
    STRING :column:
:java.lang.String:
:java.sql.Date:
    DATE :column:
:java.sql.Date:
:java.sql.Time:
    TIME :column:
:java.sql.Time:
:other:
    :columntype: :column:
:other:
:iffieldtype:
:notfk:
:repeatpkfields:
```

equally, the constructor must also change

```
    FUNCTION ex05:Table:PK(:repeatpkfields::iffieldtype::java.lang.
String:STRING:java.lang.String::java.sql.Date:DATE:java.sql.Date::java.
sql.Time:TIME:java.sql.Time::other::columntype::other::iffieldtype: :
column::,::repeatpkfields:)
```

and the getter and setter functions replace `:columntype:` with

```
:iffieldtype::java.lang.String:STRING:java.lang.String::java.sql.Date:
DATE:java.sql.Date::java.sql.Time:TIME:java.sql.Time::other::
columntype::other::iffieldtype:
```

The new primary key file

```
CLASS ex05:Table:PK

:repeatforeignkeys:
:inpk:
    ex05:Pktable:PK :uniquename:PK
:inpk:
:repeatforeignkeys:
:repeatpkfields:
:notfk:
:iffieldtype:
:java.lang.String:
    STRING :column:
:java.lang.String:
:java.sql.Date:
    DATE :column:
:java.sql.Date:
:java.sql.Time:
    TIME :column:
:java.sql.Time:
:other:
    :columntype: :column:
:other:
:iffieldtype:
:notfk:
:repeatpkfields:

    FUNCTION ex05:Table:PK(:repeatpkfields::iffieldtype::java.lang.
String:STRING:java.lang.String::java.sql.Date:DATE:java.sql.Date::java.
sql.Time:TIME:java.sql.Time::other::columntype::other::iffieldtype: :
column::,::repeatpkfields:)
    {
:repeatforeignkeys:
```

```
:inpk:
        SELF.:uniquename:PK = new ex04:Pktable:PK(:
repeatforeignkeyfields::foreigncolumn::,::repeatforeignkeyfields:)
:inpk:
:repeatforeignkeys:
:repeatpkfields:
:notfk:
        SELF.:column: = :column:
:notfk:
:repeatpkfields:
    }

:repeatforeignkeys:
:inpk:
    FUNCTION :Pktable:PK get:Uniquename:PK()
    {
        RETURN SELF.:uniquename:PK
    }

    FUNCTION set:Uniquename:PK(ex05:Pktable:PK :pktable:PK)
    {
        SELF.:uniquename:PK = :pktable:PK
    }

:inpk:
:repeatforeignkeys:
:repeatpkfields:
:infk:
    FUNCTION :iffieldtype::java.lang.String:STRING:java.lang.String::
java.sql.Date:DATE:java.sql.Date::java.sql.Time:TIME:java.sql.Time::
other::columntype::other::iffieldtype: get:Foreigncolumn:()
    {
        RETURN SELF.:uniquename:PK.get:Primarycolumn:()
    }

    FUNCTION set:Foreigncolumn:(:iffieldtype::java.lang.String:STRING:
java.lang.String::java.sql.Date:DATE:java.sql.Date::java.sql.Time:TIME:
java.sql.Time::other::columntype::other::iffieldtype: :foreigncolumn:)
    {
        SELF.:uniquename:PK.set:Primarycolumn:(:foreigncolumn:)
    }

:infk:
:notfk:
    FUNCTION :iffieldtype::java.lang.String:STRING:java.lang.String::
java.sql.Date:DATE:java.sql.Date::java.sql.Time:TIME:java.sql.Time::
other::columntype::other::iffieldtype:  get:Column:()
    {
        RETURN SELF.:column:
    }
```

```
    FUNCTION set:Column:(:iffieldtype::java.lang.String:STRING:java.
lang.String::java.sql.Date:DATE:java.sql.Date::java.sql.Time:TIME:java.
sql.Time::other::columntype::other::iffieldtype: :column:)
    {
        SELF.:column: = :column:
    }

:notfk:
:repeatpkfields:
```

Table class

equally we build the table class ex05_Table_.txt

with the same structure and tags we already know from ex05_Table_PK.txt

This class has a variable for the primary key and each foreign key that is not part of the primary key,

and all remaining fields.

```
CLASS ex05:Table:

    ex05:Table:PK :table:PK
:repeatforeignkeys:
:notpk:
    ex05:Pktable:PK :uniquename:PK
:notpk:
:repeatforeignkeys:
:repeatfields:
:iffieldtype:
:java.lang.String:
    STRING :column:
:java.lang.String:
:java.sql.Date:
    DATE :column:
:java.sql.Date:
:java.sql.Time:
    TIME :column:
:java.sql.Time:
:other:
    :columntype: :column:
:other:
:iffieldtype:
:repeatfields:

    FUNCTION ex05:Table:(:repeatpkfields::iffieldtype::java.lang.String:
STRING:java.lang.String::java.sql.Date:DATE:java.sql.Date::java.sql.
Time:TIME:java.sql.Time::other::columntype::other::iffieldtype: :
column::,::repeatpkfields:)
```

```
    {
        SELF.:table:PK = new ex05:Table:PK(:repeatpkfields::
columnjavaname::,::repeatpkfields:)
    }

    FUNCTION ex05:Table:(:Table:PK :table:PK)
    {
        SELF.:table:PK = :table:PK
    }

                FUNCTION setPrimarykey(ex05:Table:PK :table:PK)
                {
                                SELF.:table:PK = :table:PK
                }

                FUNCTION ex05:Table:PK getPrimarykey()
                {
                                RETURN SELF.:table:PK
                }

:repeatfields:
    FUNCTION :iffieldtype::java.lang.String:STRING:java.lang.String::
java.sql.Date:DATE:java.sql.Date::java.sql.Time:TIME:java.sql.Time::
other::columntype::other::iffieldtype: get:Column:()
    {
        RETURN SELF.:column:
    }

    FUNCTION set:Column:(:iffieldtype::java.lang.String:STRING:java.
lang.String::java.sql.Date:DATE:java.sql.Date::java.sql.Time:TIME:java.
sql.Time::other::columntype::other::iffieldtype: :column:)
    {
        SELF.:column: = :column:
    }

:repeatfields:
:repeatforeignkeys:
:notpk:
    FUNCTION :Pktable:PK get:Uniquename:PK()
    {
        RETURN this.:uniquename:PK
    }

    FUNCTION set:Uniquename:PK(ex05:Pktable:PK :uniquename:PK) {
        SELF.:uniquename:PK = :uniquename:PK;
    }

:notpk:
:repeatforeignkeys:
```