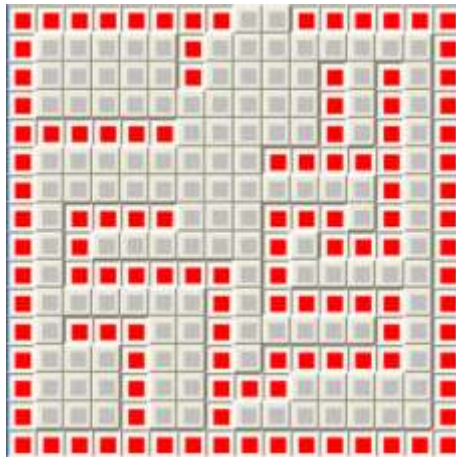# PROGRAMMING A RECORD AND A ARRAY

Objective
To be able
- o   to distinguish a record and an array in a problem
- o   to be able to program a record and a array in a Delphi program

Record

Definition

A record helps to collect together a set of different data related to a bigger one.

For exemple :
- o   **Date** is defined by a day, a month a year
- o   **Time** is defined by the Hours, the minutes, the seconds....
- o   **Student** is defined by his Name, his first Name, his birth day, his gender, his class

These complex data is called a **Structure**.

in a program, a record is a useful feature to simplify data description.

How to use a record in a program

Overview
A record is a **type** designed by a **name** and different **fields**. Each field has a name and a type.

For exemple :
```
Type
    TimeType = Record
        hour:integer;
```

```
      mn:integer;
      second:integer;
  end;
```

Type is the keyword which defined the statement.
TimeType is the name of the record type
Record is the keyword which defined the.... record. The different fields are given after this keyword .
end defined the end of the description

Other example :

```
Type
  Tstudent = Record
    name:string[25];
    firstname:string[25];
    birthday:string[10];
    gender:char;
    classname:string[10];
  end;
```

Variable record in a program

When the structure is declared, it's possible to create a variable with this type.

Example (Tstudent record) :

```
    mystudent : Tstudent;
```

In the program, this variable could be defined with the fields :
```
    mystudent.name:='Taylor';
    mystudent.firstname:='Sophie';
    mystudent.birthday:='24/04/1998';
    mystudent.gender:='F';
    mystudent.classname:='2G';
```
It's also possible to use de WITH statement.

Example (Tstudent record):

```
  WITH mystudent DO
    BEGIN
      name:='Taylor';
      firstname:='Sophie';
      birthday:='24/04/1998';
      gender:='F';
      classname:='2G';
    END;
```

Constant record in a program
It's also possible to declare a constant with a record.

Example (Tstudent record) :

```
CONST
    constantStudent : Tstudent =
        (
        name:'Taylor';
        firstname:'Sophie';
        birthday:'24/04/1998';
        gender:'F';
        classname:'2G'
        );
```
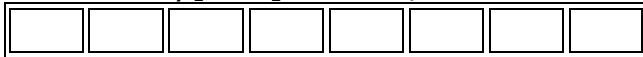
Array

Definition
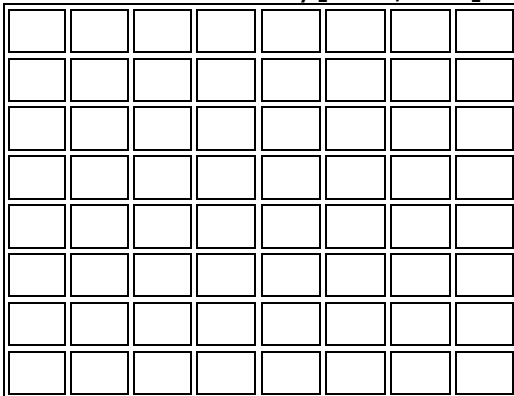
An array is a single or multidimensionnal table of data.
Each data is called an element and is accessed by its position. The position is called *INDEX*.

Examples :

A string is a one dimension array :

chaine : array[1..8] of char;

A chessboard could be an 2D array

chessboard : array[1..8,1..8] of byte;

How to declare an Array

As seen previously, an array is defined with the keyword *ARRAY*.
An array could be a variable, a constant or a type. The type of the data must be declared

Array as a variable :

Example :

```
VAR
    myTab : array[1..10] of byte;//10 bytes array
```

Array as a constant :
Example :

```
    CONST
        Days          :          array[1..7]          of          string
=('Mon','Tue','Wed','Thu','Fri','Sat','Sun');
```

Array as a type :
Example :

```
    TYPE
        TDays : array[1..7] of string;
    CONST
        Days : TDays = ('Mon','Tue','Wed','Thu','Fri','Sat','Sun');
```

How to use an Array

Access to the cells
The index helps to access to the different cells.
Example (with the array Days) :

```
For i:=1 to 7 DO writeln(Days[i]);
```

Change a data value
Example (with the array myTab) :

```
myTab[2]:=10;
```

Array of a record
It's also possible to declare an array of a record :
Example :

```
TYPE TVerb = RECORD
            infinitive:string[25];
            past:string[25];
            participle:string[25];
        end;




VAR
    irregular : array[1..100] of TVerb;
```

Vocabulary

array	tableau
cell	cellule
collect	recueillir
data	donnée(s)
grid	grille
feature	fonctionnalité
record	enregistrement
table	table, tableau
watch	montre