

# 2

## An introduction to programming

### Outcomes:

- √ Use objects, properties and methods in the components.
- √ Learn the requirements for graphical interface design.
- √ Identify different types of errors in a program.

### Introduction

When you design a program, two components are included, namely 1) the **analysis** of the problem and 2) the **design** of the graphical interface or 'screen'. The first part has to do with design and later on you will learn about the components.

### 2.1 Overview on object-oriented development

#### Analysis of the problem

It is difficult to determine in an object-oriented approach which 'parts' must be programmed. You must firstly understand the problem and analyse it. There are different 'patterns' that can be used when analysing a problem:



#### Concepts and approach to the problem

In spite of the techniques and models already in existence, it is still difficult to use and apply an object-oriented approach. To help you, it is important to think in terms of concepts. Remember that concepts merely represent ideas and that it is not a fixed 'diagrammatical' language.

#### Representation

- The input and output are indicated with a square.
- The processing (task to be done) are indicated with an oval.
- The arrows indicate the direction of events.

Several modelling languages and diagrams exist as an aid to write object-oriented programs. **Make a choice between the diagrams that are discussed.** These types are: 1) written algorithms, 2) *UML*-diagrams, 3) *CRC*-cards. Other diagrams are: 4) *KADS* (*Knowledge-Acquisition and Design structures*) and 5) *Patterns* (different types of patterns used in analysis). Consult **Appendix A** for more information.