

## Chapter 1

# Prograph: The Language

- t Overview
- t Classes
- t Methods
- t Operations
- t Multiplex Operations
- t Controls
- t Persistents

t

## Overview

<sup>33</sup>This chapter describes the Prograph language. Since Prograph is inherently visual and iconic, Prograph graphics are used throughout this description. Editor actions are described where appropriate; for a complete treatment refer to chapter 2, “The Editor Environment.” For a description of Prograph syntax and semantics in formal terms, refer to appendix IV.

Prograph is fully pictorial and dataflow in style, and it supports the single-inheritance model of object-oriented programming (OOP). All elements of Prograph take the form of icons in windows on a computer screen. Text is used only for naming Prograph elements and for comments that can be attached to those elements.

---

<sup>34</sup>NOTE: Prograph is case sensitive. The name myMethod is different from MyMethod.

---

<sup>34</sup>A Prograph program consists of the following elements

- o classes (with their associated attributes and methods)
- o universal methods
- o operations (user-defined and system-supplied, with associated controls)
- o data objects (instances of classes and of primitive data types)
- o persistents

The Prograph editor provides windows for working with classes, universal methods, persistents, and data objects. In addition, each class has a pair of windows for working with its attributes and methods. Each method, whether universal or class-based, has at least one case window in which a dataflow diagram of operations appears. Data objects can be created and modified in Value windows.

Appendix I describes the Prograph data types.