Visual programming languages (VPL)

Juri Babkin
Main ideas

• Allows user to specify a program in a two(or more)-dimensional way.

• Allows programming with visual expressions, spatial arrangements of text and graphic symbols, used either as elements of syntax or secondary notation.
VPL classification

- Many VPLs are based on the idea of "boxes and arrows", where boxes or other elements are treated as entities, connected by arrows, lines or arcs which represent relations.
  - Icon-based
  - Form-based
  - Diagram languages
VPLs paradigms

- Functional languages
- Imperative languages
- Logic languages
- Object-oriented
- Multi-paradigm
- Concurrent languages
- etc.
Dataflow approach

• Gives immediate access to the program state resulting in online debugging or automatic program generation and documentation.
• Allow automatic parallelization.
Language features

• Abstraction
• Control flow
• Data types and structures
• Documentation
• Event handling
• Exception handling
Common mistake

• Visual Basic, Visual C++ and the entire Microsoft Visual family are not, despite their names, visual programming languages.

• “executable graphics”
Some examples

- **LabVIEW** – VPL for engineers and scientist, it is commonly used for data acquisition, instrument control, and industrial automation on a variety of platforms.

- **Prograph** is a visual, object-oriented, dataflow, multiparadigm programming language that uses iconic symbols to represent actions to be taken on data.

- **Microsoft Visual Programming Language** - dataflow language for robotics programming that is a component of Microsoft Robotics Studio.