Oberon on the Web

comments on the draft version of the manuscript.

Further, Juice avoids many of the Java security issues, because strong type checking makes it virtually impossible to write an applet that violates security

content across the World Wide Web. It is thus similar to Java from Sun Microsystems. However, Juice outperforms Java in many "downloadable Applets" applications,

procedural programming style and the new object-oriented techniques using the same programming language and environment. Novel object-oriented GUIs such as

with Field-Programmable Gate Arrays (FPGAs). For this purpose a multiple-view editor is being developed, which presents a design textually and graphically. With the

In the past, the programming language Pascal suffered from the "mostly teaching language" opinion. The same should not happen to Oberon, which should be

With so many different Oberon implementations available, one does not fit all. Which one is the best for the reader, depends on the application. If one wants to link

Which implementation is good for me?

source: Amiga, Macintosh (both 68k and PowerPC) and Windows versions of the Oberon System V4. (The source of other implementations is available on request.)

and commercial. For example, a free o2c Oberon-to-C translator can be integrated in the emacs environment. o2c has been ported to a great variety of platforms. The

the number of modules nor their names need to be known a priori when invoking a program.

graphics are portable across all platforms, regardless of byte ordering (little or big endian). Most Oberon-2 programs, written with this environment in mind, can be

application under the control of the host operating system. The Oberon System has exactly the same look-and-feel, regardless of the platform it runs on. Texts and

Additionally, optional graphical user interface kits are available under both V3 and V4 environments, featuring sophisticated pop-up menus and a wide range of

Among many integrated Oberon environments available, there is an outstanding one called the Oberon System [2, 4] It originated as a complete graphical operating

What kind of programming environments are available for Oberon?

What kind of documentation is available?

Oberon is termed a "hybrid language". It supports OOP via extensible types, single inheritance, polymorphism, value and reference based objects, two categories of

visible in the interface. Although symbolic post-mortem and run-time debuggers are available, in practice there is very little need for debugging Oberon programs, and

that if a module interface has changed, all its clients have to be recompiled. Very few security gaps still remaining are well documented and can be easily avoided.

RETURN (me.x + me.y + me.t + me.u + me.v)

In this example, procedure Add is a virtual method "bound to a type T3" through the "receiver" parameter me. (For clarity, the receiver must be explicitely specified. In

another, slightly different notation motivated by user convenience:

result := object.Add (object.x, object.y); (* method call *)

We now turn our attention to the object-oriented side of the Oberon language. A simple illustration is provided below. Assuming a given structured record type

As a legitimate heir in the Pascal family, designed by the same person who also designed Pascal and Modula-2, Oberon is both an old and new programming

as classes, inheritance, methods and messages. At the same time, "Oberon" is also the name of an extensible operating system [2] written in Oberon programming

within this article. They can be obtained by sending a request to the authors.

Bibliography