WIKIPEDIA

JADE (programming language)

JADE is a proprietary object-oriented software development and deployment platform product from the New Zealand-based Jade Software Corporation, first released in 1996.[1][2] It consists of the JADE programming language, Integrated development environment and debugger, integrated application server and object database management system.

The Jade Software Corporation logo

Designed as an end-to-end development environment to allow systems to be coded in one language from the database server down to the clients, it also provides APIs for other languages, including .NET Framework,[3] Java, C/C++ and Web services.

Although a free limited licence is available for development, using the JADE platform requires per-process fees to be paid.

Contents

Language

In syntax, JADE is very similar to Pascal; its syntax is based on the language Modula-2, which was derived from Pascal. While it includes innovations lacking in Pascal or Modula-2, it lacks certain features of other modern object-oriented languages such as C# and Java.[4]

JADE is able to import and run LINC 4GL programs and was initially its core role.

Programming model

Like all of the other popular programming languages used to create database-driven software, JADE is fully object-oriented. JADE was designed to have all the most important features of object-oriented programming, but does not support the overloading of methods or operators, and until Jade 2018 lacked parameterised constructors.[4]

Classes in JADE are kept together in schemas. Schemas serve the same purpose as Java packages or namespaces in .NET, but have a hierarchy, and inherit classes from superschemas. This becomes useful especially when programming using the model–view–controller methodology, as model classes can be put in one schema, then the controller and view classes can be built on top of the model classes in a subschema.

Program structure

JADE programs are developed using a user interface that allows programmers to visually create classes and define their properties and methods. Instead of locating methods in large files, programmers select the method they would like to edit and only the code for that particular method is displayed. Also instead of compiling all the code of a program at once, in JADE, each method is compiled individually as soon as the method is completed, meaning code can be checked immediately.

All the code for a JADE application is stored in its object-oriented database. This allows for multi-user development as the database maintains concurrency control, and with each piece of the code being a separate object in the database, it is often possible to recode a system while it is live and online as long as the parts of the system being changed are not in use.

Features

The main goal of JADE was to create a seamlessly integrated programming language that would allow developers to create one application that would go from end-to-end instead of having to write three separate applications for the database server, application server and presentation client and then write the code for them to communicate with each other.

Object database

The most striking difference between JADE and other object-oriented programming languages is that its object database is a native part of its language. For example, when creating an object in JADE, it can be created as transient or persistent. Creating an object as *transient* is similar to creating objects in other object-oriented programming languages – the object is simply created in memory, and then lost when the program ends. On the other hand, when an object is created as *persistent*, when the program ends, the object will still exist and be there the next time the program starts up. So, when an object is *persistent* JADE automatically works in the background to store and retrieve the object in the database when necessary. Persistent objects can be distributed across multiple co-operating servers, with JADE automatically handling object caching and cache coherency.

There are very few differences between manipulating transient and persistent objects so JADE makes it *appear* to the programmer as if all the objects in the entire database were in local memory. Most of the time, JADE's object-oriented database is used in a multi-user system, and so JADE makes it appear to the programmer as if all the objects in the database were stored in shared memory that all users connected to the system can access, even from different computers.

With all of the program code centralised on the database server as well the data, all client nodes can be programmed as if they were running on the database server.

JADE's database is inherently object-oriented, and ACID-compliant, and has all of the standard features such as atomic transactions, locking, rollback, crash recovery and the ability to keep one or more secondary database servers synchronised with the main database for backup, disaster recovery and performance reasons.

To interoperate with relational databases for reporting, business intelligence and data warehouse purposes JADE since 2010 has provided a "Relational Population Service" that enables automatically replicating objects from its native object-orientated database to one or more relational database. This feature supports Microsoft SQL Server versions 2000, 2005 and 2008.

Three-tier model

Database-driven software, often uses a three-tier methodology with applications being split into three tiers – data storage, processing and presentation. In the data storage and processing tiers, JADE systems are a collection of co-operating servers, called nodes, and multiple nodes may be involved in each tier. Each type of node has a different name and all are capable of manipulating objects and executing business logic. A collection of nodes can be deployed across one or several machines. Database servers handle data storage and can also execute business logic, while application servers handle processing. In a three-tier model, presentation clients provide the user interface. JADE also provides a two-tier client that combines the abilities of an application server and presentation client. Traditionally, these three tiers would be created by combining three programs and having them communicate to form one system. Having the different programs separate like this has many advantages, the main one is that the system becomes scalable, that is, raising the power of the system involves simply adding more nodes.

Designing a system like this gives the programmer a decision to consider every time they want to code in a particular function. They need to decide whether the function would run best on the database server, application server or presentation client before they begin coding as it will be difficult to change that decision once the functionality is coded into one of the tiers.

This is different for JADE applications, as they are coded as one application from end-to-end. When creating a JADE application, the programmer can think as if they were creating a program that will be running on the database server – as if all the data storage, processing and presentation were happening on one computer. When the program runs on three tiers, JADE automatically knows to run all the code by default on the application server, and to send database requests up to the database server and user interface information down to the presentation client. However, it is very easy for the programmer to switch the location at which a particular method is run and move it to a different tier by changing the method signature. Because of this, the decision on where a particular piece of code should run can be made late in the development cycle, and refactoring code to run on different parts of the system ends up being a lot easier because of the way JADE allows end-to-end development of software.

Types of clients

Programmers have the facility to allow three different kinds of clients to connect to a JADE system. These three types of clients are named:

- JADE Forms
- HTML Documents
- Web Services

In the same schema, a JADE developer can create many completely separate applications which may provide different interfaces to access the same database.

JADE Forms

JADE Forms applications are made up of forms, as the name suggests. Clients need to connect through the JADE Smart thin client or Standard Client to be able to run applications that use JADE Forms.

The Smart Thin Client works by connecting to an Application Server which generally does all the processing on behalf of the Smart Thin Client, meaning the thin client only needs to be responsible for displaying forms and taking input. This means the computer running the thin client does not have to be a very powerful computer, and it does not require a fast network connection as it is not loading data from the database – JADE thin clients are often run over a dial-up connection. This is the reason they are called thin clients, as there is not a heavy requirement on computational power to run these clients.

The Standard Client is just the Smart Thin Client combined with the Application Server on one machine. In this case, the node running the client does all of the processing as well as the presentation. Standard clients have greater demands on computational power than thin clients, as they must load data from the database as well as do their own processing.

The advantages of using JADE Forms include:

- Out of the three kinds of clients, JADE Forms provide the shortest development time of JADE applications.
- Allows developers to use the same technology end-to-end.
- Smart thin clients can be packaged up so they can be installed and run on client computers in several clicks.

The disadvantages are:

- Cannot reach a worldwide audience as is possible on the World Wide Web.

JADE Forms have an interesting twist to them though. It is actually possible to run a JADE Forms application through a web browser by changing its mode to *web-enabled*. When this happens, JADE automatically generates HTML code to make pages that resemble the forms and controls, without any modifications to the code. This is a very quick way for programmers that are not competent with HTML and other web technologies to deliver a program through the web.

HTML documents

JADE supports deployment of applications to the web through its HTML documents feature. These work very similarly to ASP.NET, where developers create templates of HTML pages and leave parts in the template for the program to fill in.

The advantages of using HTML documents are:

- Allows the application to reach a worldwide audience.

The disadvantages are:

- When JADE applications use HTML documents, they are no longer using the same technology from end to end. Checking at the front end of the system may be done through JavaScript for example.
- Offloading some of the processing to front-end clients is no longer as easy or secure.

Web services

Web services are used to allow different programs to communicate with each other from remote locations in an object-oriented form. Web services cannot be accessed directly by human users. One of the uses of Web services with JADE is to allow other technologies such as .NET or Java to use JADE as the backend object-oriented database. Web services also allow JADE systems to interoperate with services provided by other non-JADE systems.

Interoperability

In addition to Web services, JADE is also capable of interfacing with other programs through language APIs (including .NET, Java, C/C++), DLL calls, ActiveX/COM objects and .NET assemblies. This allows other programs to access objects and execute methods, and can be used to provide a different interface to a JADE application. JADE 6.2 provided a Java API, .NET Assembly integration and the ability to run Smart Thin Clients on Windows Mobile devices. JADE 6.3 provides an API for .NET languages.

Multilingual abilities

JADE natively supports multilingual programs. It does this in several ways:

- Strings can be marked as *translatable*, which means they will be change depending on the current language.
- Many versions of the same form can be created to suit each language. This means interfaces can look entirely different from one language to the next.
- The developer has methods available to access the current locale of the system and so they can implement their own language-dependent features.

JADE will automatically switch to the language it detects on the system if the language is provided by the developer.

Portability

Currently JADE applications can only be run on Windows, Linux support has been dropped. JADE 6.2 allows Smart Thin Clients and a specialised Standard Client to run on Windows Mobile devices.

Code examples

In this section are some short examples of JADE code.

Hello World!

This is the "Hello World!" code in JADE:

```
helloWorld();

begin
   app.msgBox("Hello, World!", "Hello, World!", MsgBox_OK_Only + MsgBox_Information_Icon);
end;
```

or

```
helloWorld();

begin
   write "Hello, World!";
end;
```

History

JADE was originally conceived by Sir Gilbert Simpson and is currently developed by the Jade Software Corporation.[5]

The first version of JADE was JADE 3, released September 1996.

The current version is JADE 2020, replacing JADE 2018[6]

See also

- Database management system
- Object-oriented programming
- Object database

References

1. "Jade – Who We Are" (https://www.jadeworld.com/who-we-are/). Jade Software Corporation.
2. "Jade Software delivers Jade 7.1" (https://www.zdnet.com/article/jade-delivers-version-7-1/). ZDnet. Retrieved 21 December 2019.
3. "Scoop Independent News – JADE 6.3" (http://www.scoop.co.nz/stories/BU0905/S00125.htm). Retrieved 21 December 2019.
4. Toady, Tim (3 June 2009). "JADE: Obscure Programming Language of the Month" (http://www.devtopics.com/jade-obscure-programming-language-of-the-month/). *DevTopics*. Retrieved 12 March 2021.
5. Jade Software Corporation (https://www.jadeworld.com/)
6. "Latest version" (https://www.jadeworld.com/jade-platform/latest-platform).

External links

- Official website (http://www.jadeworld.com/)
- Examples and Tutorials for JADE Programmers (https://www.jadeworld.com/developer-center)

- JADE 6.1 delivers data replication to Microsoft SQL Server (https://web.archive.org/web/20060319105508/http://www.enterprisenetworksandservers.com/monthly/art.php?1899)

Retrieved from "https://en.wikipedia.org/w/index.php?title=JADE_(programming_language)&oldid=1028664018"

This page was last edited on 15 June 2021, at 09:08 (UTC).