**Язык программирования VIP**

Язык программирования VIP предназначен для разработки бизнес-приложений. Он охватывает все аспекты разработки приложений и обеспечивает:

* использование технологии объектно-ориентированного программирования при разработке приложений;
* работу с базой данных с использованием встроенных конструкций языка SQL;
* проектирование интерактивных многооконных пользовательских интерфейсов;
* обработку потока событий в интерактивных формах;
* проектирование выпадающих меню, диалоговых окон и строк статуса;
* проектирование выходных отчетов.

**Средство разработки Атлантис**

Средство разработки Атлантис предназначено для разработки и функциональной поддержки прикладного программного обеспечения. Все программные изделия Галактики разработаны на базе средства Атлантис. С его помощью квалифицированные пользователи могут расширять возможности системы Галактика в выбранном направлении.

Средство разработки Атлантис функционирует как в двухуровневой архитектуре (клиент, включающий в себя сервер приложений, и сервер базы данных), так и в трехуровневой архитектуре (клиент, сервер приложений, где находится вся прикладная логика, и сервер базы данных).

Средство разработки Атлантис может взаимодействовать с СУБД нескольких наименований. Актуальная информация о наименованиях и версиях поддерживаемых СУБД приведена в инструкциях по установке Галактики ERP.

Взаимодействие осуществляется через специально разработанные драйверы БД. Драйверы транслируют операторы диалекта языка SQL, встроенного в средство разработки Атлантис, в запросы к конкретным базам данных.

Компоненты средства разработки Атлантис:

* Компилятор VIP. Инструментальное средство, предназначенное для разработки приложений и отчетов реляционных баз данных.
* Библиотеки пользовательского интерфейса. Библиотеки предлагают набор объектов (табличные и анкетные формы, окна визуализации, экранные кнопки и т.п.) для создания экранных форм (окон) системы Галактика. Кроме того, библиотеки обеспечивают связь окон с операционной системой (Windows) и связь окон с действиями пользователя.
* Интерпретатор ресурсов. Большая часть прикладного кода, созданного на базе средства разработки Атлантис, хранится в ресурсных файлах в откомпилированном виде. Для взаимодействия прикладных программ с ресурсами в рамках средства Атлантис разработана специальная среда выполнения прикладных проектов — встроенный интерпретатор ресурсов.
* Оболочка для работы с серверами БД транслирует утверждения встроенного в Атлантис диалекта языка SQL в запросы к конкретным базам данных.
* Сервисные функции: встроенный текстовый редактор, архивация базы данных, система навигации по меню, система тестирования AQA, система лицензирования и т.д.

**Среда разработки Viper**

Среда разработки состоит из редактора текстов программ, средств управления проектом, интерактивного отладчика и других функциональных возможностей.

Viper предназначен для создания программного кода на языке VIP и компиляции ресурсных файлов.

Viper состоит из следующих функциональных модулей:

* редактор исходных текстов;
* компилятор;
* отладчик;
* средства внешнего запуска и выполнения.

Главное окно приложения имеет вид, типичный для приложений с графическим интерфейсом.

**Объектная компонентная модель**

Атлантис позволяет разрабатывать многоуровневые, распределенные (по площадкам) приложения, поддерживающие открытые стандарты взаимодействия приложений. Эта возможность базируется на объектной компонентной модели, поддерживаемой средством разработки Атлантис.

Логической единицей программного кода является объект. Объекты объединяются в компоненты. Компонент является структурной единицей программного кода. Каждый компонент инкапсулирует в себе свой кусочек БД, и не может напрямую обращаться к данным другого компонента. Для взаимодействия объектов, принадлежащих разным компонентам, предназначены интерфейсы межкомпонентного взаимодействия.

Структурной единицей приложения является модуль. Модуль может состоять как из одного компонента, так и из нескольких.

Интерфейсы межкомпонентного взаимодействия вынесены в отдельную сущность системы. Это позволяет менять реализацию программного кода и структуру БД, сохраняя при этом сам интерфейс взаимодействия. Таким образом разработчики получают возможность развивать компоненты независимо. В результате можно перейти от выпуска версий приложения целиком, к выпуску версий отдельных модулей приложения.

Декомпозиция приложения идет по двум направлениям – структурному и объектному. Соответственно, образуются структурные (вертикальные) компоненты и сквозные (горизонтальные) компоненты. Структурная декомпозиция позволяет разбить приложение на отдельные модули (например, "Управление снабжением", "Управление персоналом").

Объектная декомпозиция позволяет привнести в систему сущности, функциональность которых будет востребована в разных модулях (например, "Почтовый адрес", "Каталог подразделений"). Данные сущности образуют ядро приложения.

Для взаимодействия с другими приложениями используются стандарты COM и ActiveX. Возможные варианты:

* использование стандартных и собственных COM-объектов;
* встраивание стандартных и собственных ActiveX-компонентов (Microsoft Word, Excel, ...);
* использование функциональности объектов приложения через COM-интерфейс;
* встраивание окон приложения как ActiveX-компонентов.

Объектная модель, которую поддерживает Атлантис, содержит стандартный набор объектно-ориентированных возможностей и, сверх того, много дополнительных возможностей. К стандартному набору относятся наследование, инкапсуляция и полиморфизм.

Как объекты, так и объектные интерфейсы поддерживают наследование. Наличие двух параллельных иерархий наследования предоставляет богатейшие возможности для разработки. Можно использовать обычную иерархию объектов для наследования реализации. В то же время можно использовать иерархию объектных интерфейсов, чтобы придать общие свойства различным сущностям.

Атлантис поддерживает одиночное наследование. Однако для реализации множественного наследования есть такая возможность, как динамическое приведение типа. По сути, это наследование через включение (в себя ссылки на как бы "родителя"), которое снаружи реализации объекта выглядит прозрачно.

Дополнительные объектно-ориентированные возможности: реляционные свойства объектов.

Очень существенной возможностью является поддержка хранения и считывания объектов из БД. Наличие реляционных свойств позволило реализовать в Атлантисе объектное расширение языка SQL, в котором объекты могут участвовать в запросе наравне с таблицами. При этом уникальной особенностью является то, что оптимизатор запросов Атлантиса понимает нюансы реализации реляционных свойств конкретных объектов и оптимизирует их таким образом, что конечная производительность доступа к БД получается не хуже, чем при обращению напрямую к данным. Особенно это заметно при использовании SQL платформ (Oracle, MS SQL Server).