Oracle предлагает самый полный в своей отрасли интегрированный набор средств для разработки приложений, баз данных для поддержки самых разных подходов к разработке, технологических платформ и операционных систем.

Кроме того, Oracle предлагает разнообразные бесплатные средства [Oracle JDeveloper 11*g*](http://www.oracle.com/technetwork/developer-tools/jdev/index.html?ssSourceSiteId=ocomru) и Eclipse, способные помочь разработчикам быстрее создавать интернет-приложения и базы данных, а также упростить разработчикам, использующим .NET, развертывание на платформе Windows приложений и веб-сервисов, основанных на технологиях Oracle.

|  |
| --- |
| **Средства разработки Oracle**  Корпорация Oracle предлагает целый ряд средств разработки, которые позволяют создавать надежные, работающие приложения, способные пережит не одну смену компьютерных технологий, аппаратно-программных платформ и групп разработчиков.  В зависимости от масштаба и назначения разработки, инструменты Oracle могут комплектоваться в наборы, наиболее оптимально соответствующие поставленным задачам. Из основных инструментов Oracle необходимо выделить следующие:   1. **Designer/2000** - Средство описания разрабатываемой системы в виде всеохватывающей модели и генерации на ее основе законченных приложений для различных средств разработки. 2. **Developer/2000**- Многоплатформленное и масштабируемое средство визуального создания приложений, легко настраиваемых в зависимости от мощности сервера и клиентских компьютеров и переносимых на работу в среду Internet/Intranet. 3. **Power Objects** - Инструмент для разработки приложений небольшого масштаба, способного работать с разнообразными источниками данных. 4. **Programmer/2000** - Набор предкомпиляторов   для C/C+, заметно упрощающих создание приложений на C/C+ для сервера Oracle7.   1. **Mobile Agents** - Позволяет разрабатывать приложения для мобильных пользователей или для удаленных систем, работающих через плохие, неустойчивые каналы связи.   Необходимо отметить, что в отличие от "универсальных" средств разработок, ориентированных на работу с любыми СУБД (Delphi, Visual Basic, PowerBuilder), "родные" инструменты полностью используют все возможности сервера Oracle7. А именно, поддержка последовательностей и синонимов, работа с механизмом обеспечения секретности на сервере, доступ к хранимым на сервере процедурам и переменным, управление оптимизатором выполнения SQL-команд - использование этих возможностей либо невозможно в универсальных средствах разработок, либо требует большого труда при кодировании.  **Инструменты сквозного проектирования и разработки приложений.**  На этапе предварительного обследования деятельности предприятия (той деятельности, которую необходимо автоматизировать) используется компонента **Designer/2000**- средство построения *диаграмм деловых процессов***BPR** (Business Process Modeler). С его помощь возможно не только построить модель всех процессов, протекающих в ходе повседневной деятельности организации (предприятия), но и произвести ряд анализов, способных выявить узкие места. Даже без последующего создания приложения, такая модель позволяет лучше понять как протекает деятельность организации и найти пути по ее улучшению.  На следующем этапе строится концептуальная модель будущего приложения с помощью ряда стандартных диаграмм - диаграмм Сущность-Связь, Иерархий Функций и Потоков Данных. Эта концептуальная модель подробно детализируется, расписывается ее "проекция" на средства разработки и характеристики будущего приложения. Диаграммы концептуальной модели достаточно просты для понимания, и на этапе моделирования достаточно легко наладить обратную связь с заказчиком, для достижения полного взаимопонимания в постановке задачи.  Процесс создания модели предполагает коллективную работу над проектом. Существуют мощные средства организации такой работы, разделения модели на части для независимой разработки и "сшивания" частей модели в целое.  После подробной детализации модели можно приступать к генерации приложения. Необходимо отметить, что имеется в виду именно генерация законченного приложения, а не некого прототипа, который далее будет развиваться вручную.  Генерация приложения заключается в генерации объектов Базы Данных и генерации для выбранного средства разработки клиентских частей приложения. При генерации объектов Базы Данных создаются не только таблицы и индексы, как в большинстве конкурирующих продуктов, но и другие необходимые объекты (последовательности, синонимы, представления, ограничения целостности и т.д.), а также процедурная логика, которая должна выполняться на сервере. Создаются так же определения ролей и пользователей с предоставлением им необходимых прав по работе с Базой Данных.  Генерация для клиентской части возможно для целого ряда различных средств разработок. Сейчас поддерживаются:   1. **Oracle Developer/2000** 2. **Oracle Power Objects** 3. **Oracle WebServer** 4. **MS Visual Basic 3.0 и 4.0** 5. **Классы C/C+**   Этот список с течением времени постоянно расширяется. Необходимы минимальные изменения в настройке модели приложения, что бы произвести генерацию приложения для другого средства разработки. Таким образом обеспечивается независимость приложения от:   * **Средств разработки.**Смена средства разработки осуществляется простой перегенерацией приложения, а не переписыванием заново, с предварительным изучением нового инструмента. * **Аппаратно-программной платформы.**Даже если средство разработки не переносимо на выбранную платформу, то можно произвести генерацию приложения для того средства разработки, которое существует на избранной платформе, с полным сохранением функциональности приложения. Переход на другую платформу не вызывает необходимости полностью переписывать приложение. * **Разработчиков.**Так как разработка ведется с помощью четко формализируемых методов, то дальнейшее развитие приложения может вести любой знакомый с методологией разработчик.   Важной особенностью является возможность восстановить модель приложения по уже существующему приложению, созданного без применения **Designer/2000,** с помощью процедуры*реинжиниринга.*Так же и вручную сделанные изменения в сгенерированной системе можно отразить в модели приложения и при последующей генерации они не будут потеряны.  Таким образом, с помощью**Designer/2000**осуществляется инженерный подход к созданию промышленных приложений, позволяющий гарантированно создавать работающие, надежные приложения с требуемой функциональностью, независимые от средств разработки и аппаратно-программной платформы.  **Обеспечение независимости от смены платформ и технологий.**  Другой иллюстрацией подхода корпорации Oracle к обеспечению создания независимых от платформы приложений является **Developer/2000**. Это инструментарий визуального проектирования клиентских приложений для технологии Клиент-сервер. **Developer/2000**полностью переносим на все используемые ныне платформы, начиная от символьных терминалов и кончая графическими средами вроде Windows или Motif. Если при разработке не использовались платформенно зависимые особенности (например компоненты OCX/ActivX для Windows), то перенос приложения на другую платформу не требует никакого кодирования! Рост популярности приложений для Internet/Intranet вызвал необходимость для разработчиков изучения как новых технологий, так и новых сред разработки для Web. Перенос приложения в среду Internet/Intranet означал практически полностью его переписывание на новом средстве разработки. Поэтому многие фирмы-производители инструментальных средств поддержали технологию Netscape Plug-In, которая позволяла определенным способом распространять и вызывать приложения через Web, не сильно их переделывая. Но на самом деле, это только временное решение, т.к. для выполнения приложения необходимо держать на клиентском компьютере полностью Run-Time среду, а само приложение целиком закачивается с Web-сервера.  Применение **Developer/2000**позволяет перенести прикладную систему в среду Internet/Intranet более элегантным способом. Существует возможность разместить Run-Time среду **Developer/2000** на Web-сервере, а откомпилированное приложение передается ей безо всякой модификации. Специальный*кэтридж*Web Developer'а формирует на лету Java Applets, которые передаются на клиентский компьютер в любую программу просмотра Web. Пользователь видит перед собой тот же пользовательский интерфейс, как если бы приложение выполнялось на его компьютере, а работать может даже на DOS-компьютере с 640 КБ памяти!  Таким образом, наряду с мощными возможностями по созданию полнофункциональных клиентских приложений с богатым пользовательским интерфейсом, **Developer/2000**позволяет легко переносить созданные системы на любые существующие ныне платформы и использовать самые передовые технологии не переписывая ни строчки кода!  **Поддержка разработчиков.**  Серьезные промышленные системы создаются не на один год, требуют постоянного развития для поддержания должной функциональности. Что бы разработчики могли полноценно справляться с поставленными задачами, корпорация Oracle осуществляет ряд мер по поддержке и собственно разработчиков и ведущихся ими проектов. В их числе:   * Соответствующая международным стандартам техническая поддержка продуктов Oracle, функционирующая на территории СНГ. * Возможность ознакомления с пробными (trial) версиями продуктов Oracle, прежде чем принять решения об их закупке и затем возможность предварительного ознакомления с новыми версиями продуктов. * Два периодических русскоязычных журнала ("Мир Oracle" и "Oracle Magazine - Русское издание"), посвященных продуктам Oracle, содержащих большое число статей для разработчиков. * Действующие группы пользователей Oracle, периодические семинары. * Большой выбор учебных курсов по продуктам Oracle, как и в представительстве корпорации Oracle в СНГ, так и у его партнеров.   Все это позволяет разработчикам иметь доступ к самым передовым технологиям Oracle, максимально эффективно использовать продукты корпорации.  **Заключение.**  Средства разработки Oracle позволяют реализовать подход к разработке сложных приложений, направленный не только на получение быстрого первоначального результата, но и на обеспечение долгого жизненного цикла созданного приложения, простоты его развития и миграции на новые платформы. Использование инструментов Oracle не только позволяет полностью воспользоваться всей мощью сервера Oracle7, но и обеспечить:   * Создание многократно используемой модели прикладной системы. * Свободу в генерации приложений для различных средств разработок на различных аппаратно-программных платформах. * Независимость развития созданного приложения от существующих и будущих платформ и технологий. * Использовать самые передовые технологии. |

## СРЕДСТВА РАЗРАБОТКИ ORACLE

**JAVA и SOA**

* [Oracle JDeveloper](http://www.oracle.com/us/products/tools/019657.htm?ssSourceSiteId=ocomru)
* [Инструментарий разработчика Oracle для Spring](http://www.oracle.com/technetwork/topics/sdk-index-098464.html?ssSourceSiteId=ocomru)
* [Oracle Enterprise Pack for Eclipse](http://www.oracle.com/us/products/tools/enterprise-eclipse-pack-161756.html?ssSourceSiteId=ocomru)
* [Oracle TopLink](http://www.oracle.com/us/products/tools/toplink-adf-161769.html?ssSourceSiteId=ocomru)
* [NetBeans IDE](http://www.oracle.com/us/products/tools/050845.html?ssSourceSiteId=ocomru)
* [JavaFX](http://www.oracle.com/us/technologies/java/050854.html?ssSourceSiteId=ocomru)
* [Hudson](http://hudson-ci.org/)

[**БИЗНЕС-АНАЛИТИКА**](http://www.oracle.com/ru/solutions/business-intelligence/index.html)

* [BI Publisher](http://www.oracle.com/us/solutions/ent-performance-bi/bi-publisher-066551.html?ssSourceSiteId=ocomru)
* [Oracle BI Standard Edition One](http://www.oracle.com/us/solutions/ent-performance-bi/standard-edition-one-066552.html?ssSourceSiteId=ocomru)
* [Oracle Data Integrator](http://www.oracle.com/ru/products/middleware/data-integration/index.html)
* [Oracle Reports](http://www.oracle.com/technetwork/middleware/reports/index.html?ssSourceSiteId=ocomru)

**SOLARIS И LINUX**

* [Oracle Solaris Studio](http://www.oracle.com/us/products/tools/050872.html?ssSourceSiteId=ocomru)
* [CoolTools](http://cooltools.sunsource.net/index.html) for Solaris Studio
* [Набор инструментов Oracle Message Passing Toolkit](http://www.sun.com/software/products/clustertools/index.xml)

**NET**

* [Oracle Developer Tools for Visual Studio](http://www.oracle.com/technetwork/developer-tools/visual-studio/overview/index-097110.html?ssSourceSiteId=ocomru)
* [Oracle Data Provider for .NET](http://www.oracle.com/technetwork/topics/dotnet/index-085163.html?ssSourceSiteId=ocomru)
* [Oracle Database Extensions for .NET](http://www.oracle.com/technetwork/topics/dotnet/index-085095.html?ssSourceSiteId=ocomru)

**РОДСТВЕННЫЕ РЕШЕНИЯ**

* [Oracle GlassFish Server](http://www.oracle.com/us/products/middleware/application-server/oracle-glassfish-server/index.html?ssSourceSiteId=ocomru)
* [Oracle WebLogic](http://www.oracle.com/ru/products/middleware/appserver/index.html)
* [Oracle Grid Computing](http://www.oracle.com/us/technologies/grid/index.html?ssSourceSiteId=ocomru)
* [Oracle Grid Engine](http://www.oracle.com/us/products/tools/oracle-grid-engine-075549.html?ssSourceSiteId=ocomru)
* [Oracle и открытый исходный код](http://www.oracle.com/us/technologies/open-source/index.html?ssSourceSiteId=ocomru)
* [Сервис-ориентированная архитектура Oracle](http://www.oracle.com/ru/technologies/soa/index.html)
* [Oracle TopLink и ADF](http://www.oracle.com/us/products/tools/toplink-adf-161769.html?ssSourceSiteId=ocomru)
* [Виртуализация Oracle](http://www.oracle.com/us/technologies/virtualization/index.html?ssSourceSiteId=ocomru)

[**БАЗА ДАННЫХ И PL/SQL**](http://www.oracle.com/ru/products/database/index.html)

* [Application Express](http://www.oracle.com/technetwork/developer-tools/apex/index.html?ssSourceSiteId=ocomru)
* [Forms](http://www.oracle.com/us/products/tools/oracle-forms-161771.html?ssSourceSiteId=ocomru)
* [SQL Developer](http://www.oracle.com/technetwork/developer-tools/sql-developer/index.html?ssSourceSiteId=ocomru)

|  |
| --- |
| **JAVA API**  [Java TV API](http://java.sun.com/products/javatv/)  [Java Card API](http://java.sun.com/javacard/) |
|  |

**JAVA SDK**

* [Java Platform, Standard Edition (JDK)](http://java.sun.com/j2se/)
* [Java Platform, Micro Edition (SDK)](http://java.sun.com/javame/sdk/)
* [Java Platform, Enterprise Edition (SDK)](http://java.sun.com/javaee/sdk/)

**CAF Фреймворк.**

CAF фреймворк - это фреймворк для создания приложений системы управления базами данных (СУБД) Oracle. Термин фреймворк означает каркас, основа. То есть, начинать создавать информационную систему можно не с нуля, а с предоставляемой основы, при этом нет необходимости тратить время и усилия на функциональность, присущую, в той или иной мере большинству корпоративных информационных систем. Таким образом, появляется возможность сразу сосредоточится на бизнес логике. CAF фреймворк содержит серверную и клиентскую части.

Работа с информационной системой обычно начинается с регистрации пользователя. Что при этом происходит? Пользователь, в ответ на [приглашение системы](http://www.caf.org.ru/oraconnect.html), вводит учетное имя и пароль, которые передаются в приложение. Дальше возможны варианты. Главный вопрос - какими средствами пользователь получает разрешение на те или иные действия с информацией, то есть - насколько надежно информация защищается. Это может быть как внутренняя (реализованная производителем СУБД) система разграничения доступа, так и специально разработанная внешняя (реализованная разработчиком приложения) система. Использование внешней системы разграничения доступа, может быть оправдано применением специального сервера приложения, когда между СУБД и пользователем существует дополнительный промежуточный слой. Дополнительный промежуточный слой, в свою очередь, экономически оправдан, если число активных одновременных подключений к базе данных начинает превышать несколько тысяч, и, при этом, требования по времени ожидания на подключения к СУБД является критичными (например, не допускается ожидание подключения более 10 секунд для любого пользователя). В корпоративных информационных системах почти всегда можно разделить пользователей на группы, в зависимости от важности, с точки зрения защиты информации. В этом случае пользователи, выполняющие операции с критически важными данными, должны иметь индивидуальные учетные записи на уровне СУБД, а разграничение прав доступа выполняться средствами СУБД. Иначе уровень безопасности, предоставляемый Oracle, будет скомпрометирован. Определенная сложность встроенной системы безопасности современных СУБД, и Oracle в том числе, требует дополнительных затрат на разработку сопряжения интерфейса управления пользователями из корпоративного приложения с интерфейсом, предоставляемым СУБД. Этот факт, в условиях, ограниченного времени разработки, может служить мотивом для замены встроенных механизмов СУБД собственными, с вытекающим риском создания уязвимости в системе безопасности. Если же разработке такого интерфейса не уделяется внимания вообще, то потребуется дополнительные усилия со стороны администраторов баз данных, которые будут вынуждены управлять пользователями вручную.

CAF имеет административный интерфейс управления пользователями. Управлять учетными записями может назначенный пользователь приложения, при этом разграничение доступа выполняется средствами СУБД Oracle.

Для разработки клиентской части фреймворка CAF были выбраны Sybase PowerBuilder и Microsoft Visual C++. PowerBuilder использует "родные" высокопроизводительные интерфейсы для подключения к Oracle и патентованную технологию работы с данными - DataWindow. Это позволяет наиболее полно использовать мощь Oracle и быстро разрабатывать удобные красивые формы доступа к данным и отчеты. PowerBuilder - является объектно-ориентированным средством разработки, использующим наследование, полиморфизм, инкапсуляцию и делегирование. Для PowerBuilder написана библиотека классов PFC, построенная в соответствии с концепцией сервис-ориентированной архитектуры. PowerBuilder выполняет высокоэффективную компиляцию проекта, полностью анализируя всю иерархию классов. Следствием достоинств получения высокоэффективного исполняемого кода, является значительное время компиляции, при котором анализируется весь проект. Более того, ели используется обычная архитектура приложения PowerBuilder, нельзя использовать ранее скомпилированные модули совместно с перекомпилированными новыми. Нельзя, также, разделять (делать общими) библиотеки PowerBuilder между разными приложениями. Иными словами, если корпоративное приложение достаточно большое, то при внесении изменений всего лишь в один объект, с последующей перекомпиляцией проекта, необходимо целиком обновлять приложение у пользователей, независимо от того, что объект находился в отдельном модуле. Появление технологии PBNI (PowerBuilder Native Interface) позволило создать архитектуру, устраняющую этот недостаток. Для этого с помощью Microsoft Visual C++ создаются объекты PowerBuilder. Такая технология имеет еще и дополнительное преимущество - она изолирует системный уровень приложения от прикладного, что позволяет разработчикам сконцентрироваться на бизнес-задачах.