



JDeveloper 3.2

Oracle JDeveloper to graficzne narzędzie do budowania aplikacji w języku Java. Pozwala klientom szybko budować i instalować wielowarstwowe aplikacje bazodanowe. JDeveloper od wersji 2.0 oferuje ścisłą integrację z Oracle8i a także integrację z Internet Application Server. Pozwala ona także na budowę oprogramowania w otoczeniu wielowarstwowym przy pomocy Enterprise JavaBeans (EJB) i 'serwletów' Java. Wysoka produktywność tego narzędzia, wynikająca z zastosowania kreatorów, elementów wrażliwych na zmiany danych czy wbudowanego translatora SQLJ, powoduje, że budowa aplikacji jest szybka i łatwa. Oracle JDeveloper umożliwia tworzenie i uruchamianie aplikacji klient/server i WWW, bez zmiany kodu. JDeveloper tworzy także aplikacje rozproszone. Integracja z Oracle8i i z IAS pozwala składować logikę biznesową aplikacji po stronie servera bazy danych lub servera aplikacji.

Oracle JDeveloper w połączeniu z **Oracle Business Components for Java** stanowi pełnowartościowe środowisko programowania, które zapewnia zintegrowaną obsługę tworzenia kompleksowych aplikacji e-biznesowych do Internetu. Oracle JDeveloper udostępnia programistom zintegrowane środowisko dla efektywnego tworzenia, debugowania i wdrażania pakietów instalacyjnych, ponownego ich wykorzystywania w nowych projektach oraz dostosowywania wielowarstwowych, korzystających z komponentów aplikacji Javy i XML.

Oracle Business Components for Java to zgodne ze standardami środowisko serwerowe do tworzenia skalowalnych, wydajnych aplikacji internetowych. Środowisko to udostępnia usługi wspomagające projektowanie i pracę aplikacji, które niezwykle upraszczają tworzenie, debugowanie, dostosowywanie i ponowne stosowanie oprogramowania warstwy przetwarzania danych. Ten sam komponent aplikacji można wdrożyć jako stronę JavaServer Page (JSP), serwlet Javy, Enterprise JavaBean™ lub komponent CORBA, bez żadnych zmian w kodzie aplikacji. Dzięki tej wyjątkowej elastyczności przedsiębiorstwa mogą zainstalować tę samą aplikację w wielu różnych środowiskach serwerowych lub dokonać jej dystrybucji do różnych klientów, takich jak przeglądarki internetowe, urządzenia kieszonkowe i bezprzewodowe. Dzięki środowisku Oracle Business Components programiści mogą szybciej wyposażać swoje aplikacje w nowe funkcje, rozszerzając i dostosowując ich komponenty przy wykorzystaniu metadanych XML, bez konieczności dostępu do kodu źródłowego komponentów biznesowych.

Podstawowe cechy JDeveloper 3.2 to:

Kompleksowa obsługa języka XML — zapisywanie dokumentów XML i arkuszy stylów XSL, przekształcanie XML w HTML lub WML oraz edytowanie, kompilowanie i debugowanie (zdalne) XML i Javy. Dokumenty XML mogą być tworzone automatycznie przy wykorzystaniu środowiska Oracle Business Components for Java lub bezpośrednio, przy zastosowaniu samej bazy danych. Z główną aplikacją zintegrowano obecnie różne programy narzędziowe Oracle XML, takie jak Oracle XML Parser for Java, XML SQL czy XSQL Servlet.

Debugowanie zdalne — dołączanie i debugowanie dowolnych serwerowych komponentów Javy przy zastosowaniu VM 1.2 z zainstalowanym JPDA. Możliwe jest także zdalne debugowanie serwletów i stron JSP działających na różnych serwerach WWW, w tym Apache. Ponadto obsługiwane jest zdalne debugowanie procedur Javy zapisanych w bazie, obiektów CORBA oraz komponentów EJB działających w bazie danych Oracle8i.

CodeCoach — narzędzie JVM, które analizuje utworzony kod Javy oraz przedstawia propozycje poprawy jego jakości i wydajności.

Rozbudowana przeglądarka baz danych — umożliwia przeglądanie i edycję właściwości użytkowników i opublikowanych obiektów baz danych oraz przeglądanie właściwości różnych typów obiektów baz danych. Przy zastosowaniu czystego JDBC można również przeglądać inne bazy danych, takie jak Microsoft SQL Server i Oracle8i Lite.

Integracja z SQL*Plus — wywoływanie SQL*Plus bezpośrednio z JDevelopera, przez kliknięcie prawym przyciskiem myszki połączenia JDBC lub pliku SQL.

Nowy menedżer Web Object Manager — możliwość zarejestrowania niestandardowych komponentów Web Bean oraz serwletów w menedżerze Web Object Manager w celu ich łatwego dołączania do stron JSP.

Bogatsze możliwości JPublishera — obecnie narzędzie to umożliwia lepszą integrację utworzonych plików i klas, translację funkcji i procedur PL/SQL wysokiego poziomu oraz określenie odwzorowań Javy dla poszczególnych typów danych SQL.

Bogatsze możliwości wykorzystania obiektów sterujących powiązanych z danymi (DAC — Data-Aware Controls) — między innymi możliwość tworzenia aplikacji MDI (interfejs wielodokumentowy) wykorzystujących DAC,

pokazywanie/ukrywanie kolumn w siatce po wybraniu opcji menu wyświetlanego po naciśnięciu prawego przycisku myszki, a także zapisywanie informacji o połączeniu modułu Application Module wraz z wygenerowanymi formularzami. Nowe okno kreatora SessionInfo umożliwia wykorzystanie wszystkich możliwości wdrażania komponentów Business Component.

Bogatsze możliwości pakietu Business Components:

- Aplikacje opracowane z wykorzystaniem Business Component mogą współpracować z Oracle8i Lite.
- Obiekty Entity Objects mogą być oparte na tabelach obiektów Oracle8i, a podstawowe atrybuty — na typach obiektów definiowanych przez użytkowników Oracle8i.
- Generowanie z wyprzedzeniem bibliotek DDL dla tabel i obiektów Oracle8i.
- Możliwe jest dostosowywanie Business Components przez rozbudowę i wymianę komponentów w całych aplikacjach.

Środowisko Oracle Business Components For Java

Specyfikacja Enterprise JavaBeans™ (EJB) opisuje model tworzenia i rozmieszczania rozproszonych komponentów Javy działających po stronie serwera. Opisuje ona standardowy zestaw funkcji API oraz usług świadczonych w czasie wykonywania komponentów przez serwer EJB. Zastosowanie tego modelu umożliwi programiście skoncentrowanie się na logice aplikacji, podczas gdy serwer EJB jest odpowiedzialny za takie usługi, jak obsługa transakcji, bezpieczeństwo i zarządzanie zasobami. Jednak zapewnienie niezawodnej funkcjonalności wymaganej od aplikacji gospodarczych, które mają działać w świecie rzeczywistym, jest nadal zadaniem programisty tworzącego komponenty EJB. Specyfikacja EJB nie określa metod realizacji następujących zadań:

- pisanie kodu aplikacji gospodarczych,
- ponownego zastosowania tego kodu do wielu różnych aplikacji,
- uzyskiwania efektywnego dostępu do danych przy wykorzystaniu interfejsów użytkownika opartych na językach HTML i Java,
- dostosowywania funkcji biznesowych po wdrożeniu aplikacji.

To właśnie tym czynnościom programiści i dostawcy kompleksowych rozwiązań poświęcają najwięcej czasu i wysiłków. Celem pakietu Oracle Business Components for Java jest zaspokojenie potrzeb twórców aplikacji zajmujących się realizacją tych wymagających zadań.

Obsługując takie standardowe architektury tworzenia aplikacji, jak EJB i CORBA, Oracle Business Components for Java znacznie upraszcza tworzenie, generowanie wersji finalnej i dostosowywanie przeznaczonych dla przedsiębiorstw aplikacji napisanych w języku Java. Oracle Business Components to środowisko opracowywania komponentów aplikacji udostępniające zestaw inteligentnych komponentów oprogramowania, które są gotowe do natychmiastowego współdziałania w celu zarządzania wszystkimi typowymi funkcjami niezbędnymi do:

- efektywnego tworzenia i testowania kodu biznesowego w komponentach, które automatycznie integrują się z relacyjnymi bazami danych,
- elastycznego wielokrotnego wykorzystywania kodu biznesowego w różnych prezentacjach danych opartych na języku SQL, dla obsługi różnych zadań realizowanych przez aplikacje,
- wydajnego dostępu do danych i uaktualniania prezentacji danych w klientach, takich jak serwlety, stron JavaServer Pages (JSP) oraz Thin-Java Swing,
- łatwego dostosowywania funkcjonalności aplikacji w odpowiednich warstwach bez konieczności modyfikacji już wygenerowanej i wdrożonej aplikacji.

Eliminując znaczną ilość pracy podczas tworzenia i testowania kodu programu, związanej z typowym „łataniem” aplikacji, Oracle Business Components pozwala twórcom aplikacji na poświęcenie całego swego czasu i uwagi tworzeniu rozwiązań gospodarczych. Zalety wykorzystania tego środowiska są natychmiast dostępne i oczywiste: mniejszy koszt tworzenia aplikacji, niższe ryzyko przy podejmowaniu projektów programistycznych i krótszy czas tworzenia wersji komercyjnej aplikacji.

Warto zauważyć, że chociaż to środowisko obsługuje wszystkie typowe przypadki przy użyciu wbudowanych procedur działań, to wykorzystanie tych pożytecznych funkcji w żaden sposób nie ogranicza możliwości samodzielnego ich zdefiniowania. Ponieważ każde automatyczne działania środowiska można łatwo zmienić w komponentach przeznaczonych do określonych zastosowań przy użyciu kilku najważniejszych linii kodu, programista nigdy nie jest ograniczony do realizowania tych funkcji w identyczny sposób we wszystkich przypadkach.

Pakiet Oracle Business Components for Java stanowi efekt wielu lat wspólnej pracy projektowej i programistycznej działów Oracle zajmujących się aplikacjami, narzędziami i technologiami serwerów (*Oracle Applications, Oracle Tools, Oracle Server Technologies*). Jest to realizacja wizji Oracle dotyczącej tworzenia w nowym tysiącleciu skalowalnych, wielowarstwowych aplikacji Javy dla przedsiębiorstw.

Więcej informacji można znaleźć w dokumentacji [Oracle Business Components for Java Whitepaper](#) (aby uzyskać do niego dostęp, użytkownik musi być członkiem bezpłatnej sieci Oracle Technology Network).

Opracowywanie aplikacji XML

Integracja z Oracle XDK

Do pakietu Oracle JDeveloper 3.2 jest dołączany parser Oracle XML Parser for Java, narzędzie XML SQL oraz serwlet XSQL. Możliwe będzie proste dodawanie nowszych wersji narzędzi, które będzie można pobierać z sieci Oracle Technology Network.

JDeveloper 3.2 oferuje szeroką gamę funkcji ułatwiających programowanie:

- podświetlanie elementów składniowych Javy, XML, XSL i XSQL,
- sprawdzanie składni XML w plikach XML, XSL i XSQL,
- uzupełnianie kodu w plikach Javy — funkcja niezwykle użyteczna przy tworzeniu aplikacji z wykorzystaniem parsera XML Parser for Java,
- lokalne i zdalne debugowanie aplikacji Java XML, w tym serwletów i stron JSP,
- lokalne uruchamianie serwletów XML lub stron JSP,
- wygodna przeglądarka JavaDoc oraz przeglądarka klas do współpracy z XDK.

Opracowywanie mobilnych aplikacji przy użyciu pakietów Portal –to-Go i JDeveloper

Współdziałanie Portal-to-Go oraz Oracle JDevelopera zaowocowało powstaniem niezwykle wydajnego środowiska tworzenia mobilnych aplikacji. Pakietu JDeveloper można użyć do tworzenia dokumentów XML z bazy danych lub z Business Components for Java Application. Następnie można wykorzystać pakiet Portal-to-Go do dostarczenia prezentowanych treści do przeglądarek internetowych, cyfrowych asystentów osobistych lub telefonów komórkowych.

Z poziomu pakietu JDeveloper można obecnie testować i debugować całą aplikację. JDeveloper umożliwia też debugowanie zdalne w celu przetestowania już wdrożonych aplikacji mobilnych.

Opracowywanie aplikacji WWW z wykorzystaniem serwletów i stron Java ServerPages

Za pomocą JDevelopera można tworzyć i wdrażać aplikacje Javy działające po stronie serwera, które dostarczają dynamicznie generowane dokumenty HTML do każdego klienta z uruchomioną przeglądarką WWW.

Serwlety Javy

Pakiet JDeveloper można wykorzystać do tworzenia, testowania i debugowania serwletów Javy. Serwlety to przenośne klasy Javy, które obsługują żądania HTTP i dynamicznie generują HTML. Serwlety można przenosić między platformami i warstwami i skalować w miarę wzrostu potrzeb przedsiębiorstwa. Serwlety działają szybciej niż skrypty CGI i są zdolne do działania w środowisku wielowątkowym: ich zadaniem jest raczej obsługa prezentacji danych i warstwy gospodarczej niż niskopoziomowy kod systemowy. Kreatory JDevelopera pozwalają na tworzenie serwletów HTTP i serwletów baz danych, a zintegrowane środowisko wykonawcze serwletów umożliwia ich testowanie i debugowanie.

JavaServer Pages

Technologia JavaServer Pages (JSP) oddziela interfejs użytkownika od dynamicznego generowania treści, umożliwiając projektantom łatwe wprowadzanie zmian w wyglądzie strony bez zmian w dostarczanych dynamicznie treściach. Technologia JSP umożliwia projektowanie aplikacji przy wykorzystaniu komponentów wielokrotnego użytku, dzięki czemu tworzenie aplikacji WWW jest łatwiejsze i szybsze niż kiedykolwiek.

Pakiet JDeveloper zapewnia wyjątkową obsługę stron JavaServer Pages dzięki kompleksowemu zestawowi narzędzi ułatwiających ich tworzenie, debugowanie i wdrażanie. JDeveloper udostępnia kreatory do tworzenia stron JSP i umożliwia ich dalsze udoskonalanie za pomocą kreatora znaczników Insert Tag Wizard oraz edytora kodu źródłowego. Kreator Insert Tag Wizard umożliwia łatwe dodawanie indywidualnie zdefiniowanych elementów HTML i Javy do tworzonego kodu źródłowego. Ponadto pakiet JDeveloper jest wyposażony w pewną liczbę gotowych komponentów Web Bean oraz kreator do tworzenia komponentów własnych. Elementy Web Bean mogą być wielokrotnie wykorzystywane w stronach JSP i zapewniają możliwość wygodnej organizacji typowych elementów logicznych używanych do generowania HTML.

Korzystając ze środowiska JDeveloper Business Components for Java, można wywołać kreatora Business Component JSP Application Wizard w celu wygenerowania interfejsu użytkownika w języku HTML dla swojej aplikacji gospodarczej.

JDeveloper zawiera mechanizm wykonawczy stron JavaServer Pages. Dzięki temu możliwe jest szybkie uruchamianie stron JSP z poziomu JDevelopera w celu ich podglądania i debugowania przy wykorzystaniu zintegrowanego debugera.

JavaServer Pages to pożyteczne uzupełnienie istniejących metod tworzenia dynamicznej treści stron WWW. JDeveloper udostępnia dodatkowe narzędzia pracujące w trybie graficznym oraz zintegrowane możliwości testowania i debugowania, co sprawia, że jest to doskonałe narzędzie dla twórców stron WWW.

Współpraca programów Javy działających po stronie serwera z Oracle8i

Procedury Javy zapisywane w bazie

Za pomocą pakietu JDeveloper można tworzyć, debugować i wdrażać procedury Javy zapisywane w bazie. Ponieważ procedury takie są uruchamiane w przestrzeni pamięci systemu baz danych, złożone funkcje obsługi baz danych oraz zadania powtarzalne są wykonywane szybciej niż w przypadku kodu wykonywanego przez klienta. Ponieważ język Java ma charakter standardu branżowego i ma wielu zwolenników, wydajne programowanie baz danych w przedsiębiorstwach nigdy nie było łatwiejsze.

JDeveloper może wdrożyć dowolną klasę Javy jako zbiór procedur zapisanych w bazie. Gdy programista wybierze metody, które powinna udostępniać baza danych i dokona odpowiednich czynności w edytorze połączeń, JDeveloper automatycznie utworzy odpowiednie elementy pakietowe dla SQL, wyznaczy zależności, utworzy pakiet JAR i dokona instalacji w Oracle8i.

Ponadto JDeveloper udostępnia zintegrowany interfejs użytkownika do przeglądania bazy danych Oracle w poszukiwaniu obiektów Oracle8 oraz procedur Javy i innych.

Komponenty Enterprise JavaBean

Za pomocą JDevelopera programista może opracować standardowe komponenty Enterprise JavaBean (EJB) i wdrożyć je w bazie Oracle8i. Kreator EJV, w który wyposażony jest JDeveloper, automatycznie tworzy te nowe komponenty lub zamienia w nie klasy Javy. Kreator tworzy również wymagany do wdrożenia EJB plik deskryptorów, plik interfejsu lokalnego oraz plik interfejsu zdalnego.

Podczas wdrażania JDeveloper współpracuje z Oracle8i w celu przyspieszenia i uproszczenia wdrażania komponentów EJB. JDeveloper może również wspomóc tworzenie klas klienta Javy, które będą wywoływały wdrożony już komponent EJB. W przypadku wdrożenia komponentów EJB na serwerze Oracle8i serwer generuje klasę zdalną, która będzie pełniła w kodzie klienta rolę serwera proxy dla komponentu EJB.

Obiekty CORBA

JDeveloper ułatwia tworzenie i wdrażania dowolnej klasy Java jako obiektu CORBA działającego po stronie serwera, bez znajomości technik niskopoziomowego programowania zgodnego ze standardem CORBA.

Zintegrowane środowisko tworzenia aplikacji JDeveloper zawiera funkcje wdrażania, które dla każdej klasy Java mogą automatycznie utworzyć IDL i określić zależności niezbędne dla utworzenia pakietu. Pliki te są następnie łączone w plik JAR i wdrażane w bazie Oracle8i.

Opracowywanie interfejsu Javy za pomocą kreatorów i obiektów sterujących powiązanych z danymi

JDeveloper ułatwia tworzenie klientów Javy dzięki zaawansowanym kreatorom kodu oraz elastycznym, zgodnym ze standardami branżowymi obiektom sterującym powiązanim z danymi.

Obiekty sterujące powiązane z danymi — InfoBus

JDeveloper zawiera zestaw obiektów sterujących powiązanych z danymi, które są zgodne ze standardem branżowym InfoBus i umożliwiając oparty na formularzach dostęp do bazy danych Oracle bez potrzeby ręcznego tworzenia kodu JDBC (Java-Database Connectivity). Powiązane z danymi obiekty sterujące JFC można łatwo dodać do formularza — wystarczy przeciągnąć je myszą z palety narzędziowej. Sposób obsługi transakcji oraz informacje o połączeniach można łatwo ustawić w edytorze właściwości pracującym w trybie graficznym. Pomimo prostoty obiektów sterujących, ich dopracowana architektura obejmuje wiele zaawansowanych funkcji, takich jak zapobiegawcze blokowanie na poziomie wierszy. Ponieważ obiekty sterujące są wdrażane jako komponenty JavaBean i stanowią uzupełnienie JFC, programiści mogą łatwo zwiększyć zakres dostarczonych elementów.

Kreator formularzy danych

JDeveloper ułatwia również szybkie tworzenie w pełni funkcjonalnych formularzy danych przy użyciu kreatora formularzy danych (Data Form Wizard). Kreator ten umożliwia elastyczne tworzenie formularzy jednotabelowych lub odzwierciedlających złożone struktury bazy danych. Po zakończeniu pracy w edytorze połączeń programiści mogą dokonać wyboru konkretnych tabel bazy danych lub komponentów gospodarczych, jak również określić warunki tworzenia połączeń, jeśli jest to potrzebne. Wygenerowane formularze wykorzystujące powiązane z danymi obiekty sterujące InfoBus nie wymagają dalszego tworzenia kodu, ale mogą być z łatwością rozszerzane o nowe funkcje w celu ich dostosowania do potrzeb użytkowników.

Programowanie z wykorzystaniem komponentów JavaBean

W zakresie narzędzi do graficznego tworzenia komponentów JavaBean pakiet JDeveloper 3.2 oferuje rozwiązania bardziej zaawansowane niż jakiegokolwiek inne zintegrowane środowisko programowania w Javie. JDeveloper 3.2 udostępnia kompleksowy zestaw narzędzi z graficznym interfejsem użytkownika, który ułatwia zaawansowane modyfikacje elementów kodu nie tylko w przypadku JavaBean, ale również każdej innej klasy — wystarczy wypełnić odpowiednie pola. Nowa, graficzna metoda „edytowania” kodu jest podobna do narzędzi „dwukierunkowych”, ale służy do zastosowań ogólnych (nie tylko graficzny interfejs użytkownika, nie tylko komponenty JavaBean). Modyfikowaną reprezentację kodu łatwo zrozumieć, a wygenerowany kod, chociaż skomplikowany, jest zawsze wolny od błędów. Dzięki narzędziom graficznego interfejsu użytkownika niezwykle proste staje się dodawanie, edytowanie i usuwanie wszystkich ważnych elementów dowolnej klasy, włączając pełną obsługę Fields (pól), Methods (metod), Properties (właściwości), BeanInfos (informacji o komponentach typu bean), Event Sets (zestawów zdarzeń) i innych.

JDeveloper udostępnia kreatory umożliwiające rozpoczęcie pracy nad JavaBean, BeanInfo, Property Editors, Customizers i Custom Events. Następnie można użyć projektanta klas do wykonania bardziej zaawansowanych prac. Dzięki kreatorowi JDeveloper Deployment tworzenie wersji użytkowej aplikacji i jej wdrożenie jest proste. Co więcej, paleta komponentów JDeveloper umożliwia dodanie wszystkich ulubionych komponentów JavaBean do dowolnej strony na Palecie. W rzeczywistości, dzięki funkcji Insert JavaBean (Wstaw komponent JavaBean do Palety komponentów) z menu Package (Pakiet), możliwa jest jednoczesna praca nad tworzeniem komponentu JavaBean i jego wykorzystywanie, co znacznie przyspiesza jego powstanie. Warto zauważyć, że podczas tworzenia JavaBean w JDeveloperze pakiet ten generuje tylko to, co jest potrzebne — nie powstają żadne ukryte interfejsy ani klasy podstawowe — jedynie wymagana klasa. Ponieważ JDeveloper traktuje JavaBean jako kod rodzimy i wykorzystuje zaawansowaną architekturę analizy kodu, może zaoferować wyjątkowe narzędzia graficznego interfejsu użytkownika do tworzenia komponentów JavaBean.

Uproszczony dostęp do baz danych z poziomu programów Javy

JDeveloper oferuje niezrównane narzędzia i technologie obsługi dostępu do baz danych przy użyciu języka Java.

Zintegrowany JDBC

W celu umożliwienia współpracy z bazami danych dostarczono kilka programów sterujących (*driverów*) JDBC dla uzyskania najbardziej efektywnego dostępu z dowolnej implementacji Javy: „Oracle thin JDBC” jest najlepszym rozwiązaniem dla apletów, a program sterujący „Oracle-OCI” zapewnia najlepszy dostęp aplikacjom Java uruchamianym w warstwie pośredniej. Dla implementacji używających bazy danych innej niż Oracle JDeveloper udostępnia program pośredniczący firmy Sun — „ODBC-JDBC Bridge”, który pozwala na uzyskanie dostępu w standardzie JDBC do dowolnej bazy danych zgodnej z ODBC. JDeveloper umożliwia również stosowanie dowolnych programów sterujących JDBC innych firm.

Zintegrowany translator i debugger SQLJ

Gdy wymagane jest tworzenie kodu obsługi bazy danych, JDeveloper zapewnia najlepsze środowisko tworzenia kodu Java-baza danych. Oprócz zwykłego JDBC, możliwości kompilacyjne JDevelopera obejmują zintegrowany translator i debugger SQLJ, które umożliwiają osadzanie instrukcji SQL bezpośrednio w kodzie Javy oraz ich debugowanie jako kodu wplecionego w kod źródłowy. Dla doświadczonych projektantów baz danych SQL jest znanym językiem, który opisuje interakcje z bazami danych w sposób bardziej zwięzły niż czysty JDBC.

Instrukcje SQLJ są przekładane na standardowe instrukcje JDBC przed zwykłą kompilacją Javy, a zatem nie wprowadza się własnych rozszerzeń języka. Translator może opcjonalnie sprawdzić składnię kodu SQL oraz zweryfikować kod w zestawieniu z rzeczywistym schematem bazy danych przed dołączeniem go do wywołań JDBC; komunikaty o błędach często kierują później programistę do zakwestionowanych instrukcji. Kod SQL jest debugowany w trybie bezpośrednim, tak samo jak standardowy kod Javy.

Jako standard branżowy, SQLJ gwarantuje wyższą jakość kodu dzięki weryfikowaniu wywołań SQL w czasie kompilacji i zwiększa produktywność programistów dzięki znacznemu ograniczeniu wielkości tworzonego kodu odpowiedzialnego za współpracę z bazą danych.

Edytor połączeń i przeglądarka baz danych

JDeveloper udostępnia graficzny edytor połączeń, stanowiący standardowy element wszystkich kreatorów i narzędzi do wdrażania. Edytor ułatwia wybór programu sterującego JDBC, wprowadzanie nazwy użytkownika i hasła oraz przeglądanie rzeczywistego schematu bazy danych. Edytor jest bardzo łatwy w użyciu, a mimo to na każdym kroku dostępna jest dodatkowa pomoc. Wygenerowana konfiguracja jest następnie zamieniana na kod JDBC, a odpowiednie powiązane biblioteki JDBC są automatycznie importowane do projektu. JDeveloper zapamiętuje połączenia z bazami danych, umożliwiając ich wygodne wykorzystanie w późniejszym czasie.

JDeveloper zapewnia również zintegrowany interfejs użytkownika służący do przeglądania schematów Oracle8i, co ułatwia pracę przy każdym programowaniu bazy danych. JDeveloper spełnia wszystkie wymagania związane z tworzeniem aplikacji typu Java-bazy danych.

Kreator JPublisher

JPublisher ułatwia stosowanie obiektów Oracle8 w programach Javy przez automatyczne generowanie odpowiednich definicji klas Javy. Klasy utworzone przez JPublisher'a zawierają metody służące do konwersji danych z SQL na Javę i z Javy na SQL, jak również metody pozwalające na uzyskiwanie informacji o atrybutach obiektów i ustawianie tych atrybutów. JPublisher zwiększa produktywność programistów przez generowanie klas Javy przyporządkowanych obiektom Oracle8, przy jednoczesnym zapewnieniu elastyczności umożliwiającej rozszerzenie wygenerowanych klas dla zaspokojenia specyficznych potrzeb użytkowników.

Przełęczarka baz danych zawarta w pakiecie umożliwia programistom nawigację po zawartości schematów baz danych, uzyskiwanie podglądu procedur zapisanych w bazie i obiektów Oracle8. Z poziomu tego interfejsu możliwe jest wywołanie kreatora JPublisher Wizard w celu wygenerowania klas pakietowych Javy dla obiektów Oracle8.

Efektywne środowisko programowania wizualnego (IDE)

Technologia dwukierunkowa

Środowisko IDE JDeveloper wykorzystuje technologię dwukierunkową, która zapewnia synchronizację kodu i projektu widocznego w formie graficznej podczas programowania. Kompilacja nie jest konieczna: zmiana dokonana w narzędziu wizualnym zostaje natychmiast odzwierciedlona w kodzie. Podobnie zmiany w kodzie są natychmiast odzwierciedlane w narzędziu do projektowania wizualnego. Co więcej, modyfikacje właściwości komponentów JavaBean są odzwierciedlane zarówno na poziomie kodu źródłowego, jak i w wizualnym edytorze właściwości. Uporządkowany widok wszystkich projektów obejmuje każdą klasę wraz z jej metodami, właściwościami i zasadami dziedziczenia, a wszystkie te informacje są zsynchronizowane w czasie pracy programisty z tworzonym kodem i graficzną reprezentacją aplikacji.

Kompilator przystosowany do kodu Unicode, ze zintegrowanym sprawdzaniem zależności i możliwością „zaciemniania” generowanego kodu

Zaawansowany kompilator JDeveloper, przystosowany do kodu Unicode, automatycznie weryfikuje zależności podczas kompilacji i przyporządkowuje komunikaty o błędach do odpowiednich fragmentów kodu (zarówno w zakresie standardowego kodu Javy, jak i SQLJ). Kompilator umożliwia również „zaciemnianie” kodu.

Wizualny debugger

Zintegrowany wizualny debugger obsługuje punkty przerwania, wyliczenia wartości, obserwatory oraz szeroki zakres poleceń sterujących wykonywaniem programu. Wizualne debugery mogą współpracować zarówno z JDK 1.1, jak i Javą 2.

Zarządzanie wieloma projektami i biblioteki

Nawigator projektów JDeveloper pozwala programistom na zarządzanie w ramach jednej sesji wieloma projektami, z których każdy ma określone, własne właściwości kompilacji, uruchamiania i debugowania. Obecnie programiści mogą pracować z projektami tworzonymi dla różnych warstw i różnych docelowych wdrożeń w ramach tej samej sesji. Nawigator projektów umożliwia dostęp do zawartości projektów, plików i klas, jak również do całej hierarchii ścieżek do klas Javy z poziomu jednego interfejsu.

JDeveloper ułatwia zarządzanie zależnościami ścieżek do klas dzięki zastosowaniu bibliotek. Biblioteki udostępniają interfejs, który umożliwia szybkie przełączanie się między grupami zapisów ścieżek, ułatwiając lokalizację klas, plików z kodem źródłowym i dokumentacji każdego pakietu.

CodeInsight

Podczas pisania kodu w oknie edytora zintegrowane środowisko programowania JDeveloper może wyświetlać podpowiedzi, takie jak listy parametrów i informacje o elementach danej klasy. Eksplorator pakietów wyświetla wszystkie pakiety w bieżącej ścieżce do klas.

Wyczerpująca dokumentacja online

Dokumentacja online zawiera opis pakietu, jak również kompletny opis języka. Poza tym udostępnione są skomentowane przykłady zastosowania wszystkich technologii w kompletnej, działającej aplikacji wielowarstwowej, ze szczegółowymi instrukcjami opisującymi tworzenie najważniejszych elementów.

Interfejs MDI umożliwiający dokowanie

Narzędzie programowania JDeveloper udostępnia interfejs wielodokumentowy (MDI), co umożliwia szybkie przełączanie się między edytorami i narzędziami do projektowania wizualnego. Okna Nawigatora projektów (Project Navigator), Inspektora właściwości (Property Inspektor) i okna wyświetlające komunikaty podczas debugowania mają charakter okien z paletami narzędzi i mogą być dokowane. Dzięki tej elastyczności programiści mogą uzyskać maksymalną wydajność, dostosowując swoje środowisko do etapu realizacji projektu lub roli pełnionej przez daną osobę.

JDeveloper Addin API

JDeveloper Addin API to kolekcja interfejsów programowania (API) Javy, stanowiąca rozszerzenie środowiska programowania Oracle JDeveloper. Dzięki Addin API programiści zyskują dostęp do wewnętrznych funkcji JDevelopera, co umożliwia im rozbudowanie środowiska IDE o dodatkowe narzędzia opracowane przez użytkowników lub uzyskanie integracji z narzędziami innych firm (np. służącymi do kontroli wersji).

JDeveloper szeroko wykorzystuje Addin API do tworzenia wielu rodzajów rozszerzeń środowiska IDE. Przykładem może być kreator Enterprise JavaBean, narzędzia do projektowania dwukierunkowego, narzędzie do projektowania menu, cała zawartość menu kreatora, Galeria Plik/Nowy (File/New Gallery) i inne elementy. JDeveloper zawiera kompleksowy zestaw przykładowych komponentów Addin z sieci [Oracle Technology Network](#), który za pomocą przykładowego kodu i dokumentacji demonstruje sposób, w jaki można tworzyć różne rodzaje rozszerzeń Addin. Zestaw Addin API został pobrany przez wielu niezależnych producentów oprogramowania, partnerów i klientów i umożliwia dołączanie do JDevelopera dowolnych funkcji. Jest on prosty w użyciu i korzysta wyłącznie z języka Java.

Obsługa języka Java

JDeveloper jest w 100% zgodny z językiem Java — nie tworzy żadnego kodu wymagającego rozszerzenia standardowej wersji tego języka, co zapewnia całkowitą przenośność i możliwość współpracy z różnymi platformami, dzięki zgodności ze wszystkimi specyfikacjami języka Java: JDBC, SQLJ, specyfikacją serwletów, JSP, EJB, JTS, RMI, JAR, serializacją itd.

Java 2 i przełączanie JDK

JDeveloper jest oferowany z zestawem Java 2 SDK w wersji 1.2.2 i w pełni obsługuje standard JFC/Swing. Jednak ze względu na występowanie obecnie wielu różnych zestawów JDK, pakiet JDeveloper umożliwia również przełączanie między dostępnymi zestawami, co pozwala na tworzenie rozwiązań do dowolnego środowiska.