

современным информационным технологиям возможно только при условии комплексного изучения всех составных частей компьютерных наук.

#### Литература

1. Одинец В.П. Зарисовки по истории компьютерных наук: учебное пособие: в 3 ч. / В.П. Одинец. – Сыктывкар: Коми пединститут, 2011. – Ч. I. – 200 с.
2. Поспелов Д.А. Становление информатики в России / Д.А. Поспелов, Я.И. Фет // Очерки истории информатики в России. – Новосибирск, Научно-издательский центр ОИГМ, 1998. – С. 7-44.
3. Малиновский Б.Н. Очерки по истории компьютерной науки и техники в Украине / Б.Н. Малиновский. – К.: «Феникс», 1998. – 452 с.
4. Брукшир Дж.Г. Введение в компьютерные науки. Общий обзор / Дж.Г. Брукшир. – Computer Science: An Overview. – М.: «Вильямс», 2001. – 688 с.

*Шаров С.В., канд. пед. наук, доцент; Близнюк Я.А., студент  
Мелітопольський державний педагогічний університет імені Богдана Хмельницького,  
м. Мелітополь  
Кафедра інформатики і кібернетики*

## **ВИБІР ІНСТРУМЕНТАЛЬНОГО ЗАСОБУ ДЛЯ РОЗРОБКИ ДОВІДКОВО-ІНФОРМАЦІЙНОЇ СИСТЕМИ**

**Актуальність.** На сьогодні різноманітні інформаційні системи, зокрема довідково-інформаційні системи, є важливою частиною інформаційної інфраструктури суспільства. Зазвичай, вони містять принаймні два функціональні компоненти: власне програмний засіб (або комплекс засобів) для відображення інформації та здійснення обчислень загального призначення та база даних (або сукупність баз даних) для надійного зберігання та оновлення даних, пошуку та обробки інформації. Швидкість обробки інформації та її об'єми, що зберігаються у базі даних, певним чином впливає працездатність довідково-інформаційної системи.

**Мета статті** полягає в аналізі можливостей інструментальних середовищ стосовно доступу до бази даних Access для розробки довідково-інформаційної системи з продажу та ремонту комп'ютерного обладнання.

**Виклад основного матеріалу.** Слід зазначити, що процедурні мови та мови запитів для обробки баз даних (БД) відрізняються семантикою та стратегіями оптимізації. Ці відмінності неформально називаються «втратою відповідності» між імперативними програмами та декларативними запитами; порожніми покажчиками і невизначеними значеннями в аспекті відсутності даних; алгоритмами і структурами даних, з одного боку, і відносинами та індексами з іншого боку; потоками управління та транзакціями [7].

Для забезпечення зв'язку між базами даних та різноманітним програмним забезпеченням призначені так звані драйвери баз даних. І в залежності від того, який передбачається використовувати інструментальний засіб для розробки модулів програмного забезпечення, потрібно обрати відповідний драйвер.

У межах виконання дипломної роботи нами було поставлено за мету розробити довідково-інформаційну систему з продажу та ремонту комп'ютерного обладнання. Вона повинна реалізовувати наступні завдання:

зберігати інформацію про співробітників підприємства з продажу та ремонту комп'ютерів, містити перелік комп'ютерного обладнання, містити інформацію про замовлення на ремонт обладнання, продаж комп'ютерів тощо.

Одним із початкових етапів розробки довідково-інформаційної системи є вибір інструментального засобу, який повинен мати функціональні можливості щодо організації доступу до баз даних. Передбачається, що довідково-інформаційна система з продажу та ремонту комп'ютерного обладнання буде працювати з локальною СУБД Microsoft Access, оскільки вона має потужний графічний інтерфейс. крім того, вона спроможна реалізувати всі передбачені завдання, які висуваються перед довідково-інформаційною системою стосовно зберігання та обробки даних.

На сьогодні існує декілька технологій доступу до баз даних, які використовуються середовищами програмування. Це ODBC (Open Database Connectivity), JDBC (Java Database Connectivity), DAO (Data Access Objects); ADO (ActiveX Data Objects, компактна технологія доступу до даних); OLE DB (база даних OLE, набір інтерфейсів компонентної об'єктної моделі COM) та інші. З урахуванням цих технологій, нами було зроблено аналіз існуючих середовищ програмування, які мають можливості обробляти табличні дані Access та швидко створити якісний програмний продукт. Було проаналізовано наступні інструментальні середовища: Java, C#, Delphi.

*Java* може бути вбудована в СУБД за допомогою різних способів:

1. Застосування повноцінної мови програмування для складання процедур, що зберігаються. У сервер вбудовується віртуальна машина Java і внутрішній інтерфейс JDBC. Драйвер JDBC, який входить до складу дистрибутиву Java SE, є незалежним промисловим стандартом взаємодії Java-додатків з різними СУБД. Він є інтерфейсом, заснованим на специфікаціях SAG CLI (інтерфейс рівня виклику групи доступу SQL).

Загальна схема взаємодії інтерфейсів Java певним чином нагадує схему ODBC з її драйвер-менеджером. JDBC Driver Manager – це основний стовбур JDBC-архітектури. Його первинні функції полягають у з'єднанні Java-програми і відповідного JDBC драйверу. З іншого боку, на відміну від інтерфейсу ODBC, JDBC організований набагато простіше. Головною його частиною є драйвер, що поставляється фірмою JavaSoft для доступу з JDBC до джерел даних. Цей драйвер є самим верхнім в ієрархії класів JDBC і називається DriverManager [1].

2. Використання бібліотеки jackcess. На думку деяких дослідників, API бібліотеки достатньо прозоре. Проте, із зростанням об'єму бази даних швидкість запису даних зменшується. Наведемо декілька програмних рядків для роботи з БД за допомогою цієї бібліотеки [2]:

а) Створення нової бази даних:

```
File file = new File("d:/proba.mdb")
```

```
Database db = Database.create(file);
```

б) Створення нової таблиці: db.createTable("tabl", columns);

в) Створення запису у таблиці:

```
Table newTable = db.getTable("tabl");
```

```
newTable.addRow(new Object[] {"value"});
```

C#. Для з'єднання з базою даних Access в C# слід використовувати клас OleDbConnection з наступними параметрами з'єднання:

provider=Microsoft.Jet.OLEDB.4.0;data source=databaseFile, де databaseFile – абсолютний шлях до файлу бази даних Access. Приклад параметрів з'єднання з файлом бази даних Microsoft Access : string connectionString = "provider=Microsoft.Jet.OLEDB.4.0;" + "data source=C:\data\proba.mdb";

Вищезазначений рядок параметрів з'єднання слід передати конструктору OleDbConnection з System.Data.OleDb. Нижче приведений лістинг програми, яка підключається до бази даних у форматі MS Access і виконує запит на вибірку даних з використанням оператора вибірки Select та Where [6].

```
// створюємо об'єкт OleDbCommand
OleDbCommand myOleDbCommand =
myOleDbConnection.CreateCommand();
// задаємо SQL-запит до бази даних у властивості CommandText об'єкту
OleDbCommand.
myOleDbCommand.CommandText = "SELECT CompanyName,
ContactName, Address "+ "FROM Clients WHERE ContactName = noname";
// відкриваємо з'єднання з БД за допомогою методу Open()
myOleDbConnection.Open();
```

Інший, більш вдалий, варіант доступу до БД полягає у використанні ADO.NET – набору класів, які використовуються з C# і .NET Framework для доступу до даних у таблично-орієнтованому форматі. З його допомогою можна підключитися до реляційних баз даних, таких як Microsoft SQL і сервер Microsoft Access та ін., і навіть до нереляційних джерелами даних. Технологія ADO.NET інтегрована в .NET Framework і спроектована для використання з будь-якою мовою .NET, особливо з C#. Набір класів ADO.NET містить простір імен System.Data і його вкладені простори імен, такі як System.Data.SqlClient та System.Data.Linq, та деякі специфічні, пов'язані з даними, класи з простору імен System.Xml. Слід зазначити, що ADO.NET виконує ті ж завдання, що і технологія ADO, надаючи легкі у використанні класи для доступу до даних, розширені для середовища програмування .NET. Але класи, властивості та методи ADO.NET суттєво відрізняються від їх аналогів в ADO [3, с. 891].

*Delphi.* Для доступу до бази даних Access найбільш оптимальним є використання технологія ADO. Ця технологія багато в чому схожа на технологію BDE. Кожна з цих технологій підтримує навігацію по наборам даних, операцію з ними, обробку транзакцій тощо. Проте ADO більш нова технологія, яка відповідає сучасним вимогам і не несе з собою безліч застарілих обмежень. Крім того, використовувати її значно легше. Технологія ADO входить до більш загального поняття – компоненти доступу до даних Microsoft, Microsoft Data Access Components (MDAC), яка об'єднує декілька технологій Microsoft для доступу до баз даних (OLE DB, ADO, ODBC та RDS (Remote Data Services) [5, с. 17].

На комп'ютері, який виступає у якості серверу даних (це може бути або сервер за технологією клієнт/сервер або моделлю файл/сервер) встановлюється програмна надбудова над технологією OLE DB, яка може “розуміти” запити об'єктів технології ADO та “вміє” переводити ці запити до бази даних у відповідні дії з даними. Ця надбудова отримала назву провайдер даних. Взаємодія провайдера та компонентів ADO здійснюється на основі технології ActiveX, яка є універсальною для ОС Windows. Причому ADO-компоненти

реалізується як клієнти COM, а провайдер – як COM-сервер. У ролі провайдера використовується Microsoft Jet 4.0 OLE DB Provider [4, с. 539]. Нижче наведено програмний код для реалізації доступу до БД:

```
ADOConnection1.ConnectionString := 'Data Source=' +
OpenDialog1.FileName; ADOConnection1.Open;
```

Для роботи з базою даних з проекту Delphi використовується ADOQuery (властивість SQL). Нижче приведений приклад створення запиту засобами середовища Delphi:

```
// підготовка компоненту для роботи
ADOQuery2.Close;
ADOQuery2.SQL.Clear;
// створення SQL-запиту
StrQuery := 'insert into Temperatura (p1) values (' + IntToStr(до[4]) + ')';
ADOQuery2.SQL.Add (StrQuery);
ADOQuery2.ExecSQL;
ADOQuery1.Open;
```

Отже, програмна організація доступу до баз даних не викликає жодних проблем. Існує значна кількість середовищ програмування, які мають можливості для роботи із базами даних (Delphi, Microsoft Visual Studio, C++ Builder та ін.). Кожний з таких продуктів містить значний набір методів, процедур і компонентів для обробки складних структурованих даних.

Для доступу до баз даних, зокрема СУБД Access, використовується або інтерфейс ODBC, який вважається застарілим та важким, або технологію ADO. Одним із найбільш легких у використанні цієї технології при доступі до баз даних є візуальне середовище розробки додатків Delphi, яке містить цілу низку компонентів для роботи з базами даних та дозволяє швидко створити програмний продукт.

#### Література

1. Дунаев С.Б. Доступ к базам данных из Java-программ и проблемы русификации [Електронний ресурс] / С.Б. Дунаев. – Режим доступу: <http://www.javaportal.ru/java/articles/jdbc.html>.
2. Работа с MS Access из Java [Електронний ресурс]. – Режим доступу: <http://www.javenue.info/post/71>.
3. Уотсон К. Visual C# 2008: базовый курс / Уотсон К., Нейгел К., Педерсен Я., Рид Д., Скиннер М. – М.: ООО «И.Д. Вильямс», 2009. – 1216 с.
4. Фаронов В.В. Delphi 5. Руководство разработчика баз данных / Фаронов В.В., Шумаков П.В. – М.: «Нолидж», 2000. – 640 с.
5. Елманова Н. Delphi и технология COM. Мастер-класс / Елманова Н., Трепалин С., Тенцер А. – СПб : Издательский дом «Питер», 2002. – 640 с.
6. C# и БД MS Access [Електронний ресурс]. – Режим доступу: <http://2lx.ru/2009/01/csharp-i-msaccess>.
7. Cook William R. Integrating Programming Languages & Databases: What's the Problem? (переклад Сергія Кузнєцова) [Електронний ресурс] / William R. Cook, Ali H. Ibrahim. – Режим доступу: [http://citforum.ck.ua/database/articles/impedance\\_mismatch](http://citforum.ck.ua/database/articles/impedance_mismatch).