

УДК 004.4'2

А.Ю. Коцюба, В.В. Матвіюк

Луцький національний технічний університет

### Імітаційне моделювання симуляції розіграшів національної лотереї "Супер Лото"

*Коцюба А.Ю., Матвіюк В.В. Імітаційне моделювання симуляції розіграшів національної лотереї "Супер Лото". Засобами C++ Builder та MS Office розроблено програмний комплекс, за допомогою якого можна будь-яку кількість разів проводити розіграші національної лотереї "Супер Лото". Одержані результати в майбутньому можна досліджувати методами математичної статистики, що дасть можливість відповісти на поставлені в роботі питання.*

*Ключові слова:* імітаційна модель, національна лотерея "Супер Лото", ADO-технологія, OLE-технологія.

*Рис. 5, Лит. 8*

*Коцюба А.Ю., Матвіюк В.В. Имитационное моделирование симуляции розыгрышей национальной лотереи "Супер Лото". Средствами C++ Builder и MS Office разработан программный комплекс, с помощью которого можно любое количество раз проводить розыгрыши национальной лотереи "Супер Лото". Полученные результаты в будущем можно исследовать методами математической статистики, что позволит ответить на поставленные в работе вопросы.*

*Ключевые слова:* имитационная модель, национальная лотерея "Супер Лото", ADO-технология, OLE-технология.

*Рис. 5, Лит. 8*

*Kotsyuba A.Yu, Matviyuk V.V. Imitation modeling of simulation draws National Lottery "Super Lotto". By means of C++ Builder and MS Office developed a software package that allows you to any number of times to conduct the draw of National Lottery "Super Lotto". The results obtained in the future can be studied by methods of mathematical statistics, which give the opportunity to respond to the questions in the question.*

*Keywords:* imitation model, the National Lottery "Super Lotto", ADO-technology, OLE-technology.

*Fig. 5, Ref. 8*

**Постановка наукової проблеми та її значення.** Поняття імітаційне моделювання, імітаційна модель це терміни одного з найбільш потужних методів дослідження реально існуючих об'єктів, систем самої різної природи і складності. Суть методу полягає в побудові так званої імітаційної моделі досліджуваного об'єкту і цілеспрямованого експерименту з такою моделлю для отримання відповідей на ті або інші питання поставлені перед дослідником. В даній роботі побудована і реалізована імітаційна модель симуляції розіграшів національної лотереї "Супер Лото". Завдяки цій моделі можна провести дослідження, які б дали змогу відповісти на наступні питання:

- коли і чи можливо виграти у вищевказаній лотереї;
- скільки ставок мінімально потрібно зробити, щоб хоча б повернути витрачені гроші з мінімальною ймовірністю програшу та мінімальними затратами на ставки;
- чи доцільно проводити системні ігри (мається на увазі, що при однакових затратах для випадків системної та звичайної гри, чи буде ймовірність програшу різнитися у вигідну для системної гри сторону);
- якщо доцільно, то яку кількість кульок найкраще вказувати у системній грі.

Звичайно, всі результати ми отримуємо за допомогою так званого генератора псевдовипадкових чисел [-1]. Але експерименти можна робити будь-яку кількість разів та записувати всі результати. При цьому ніяких грошових затрат не буде. В цьому і полягають недолік та перевага симуляції розіграшів побудованої в даній роботі імітаційної моделі.

Крім вищеописаних недоліків та переваг, побудована модель (програмний комплекс) має ще дві досить суттєві переваги:

1. В національній лотереї "Супер Лото" весь час змінюють кількість кульок (номерів), з якої випадковим чином вибирається 6 кульок (номерів). Для нашого програмного комплексу це не проблема. При початковому запуску exe-файла ця кількість вказується. По замовчуванню вона є такою як в сучасних розіграшах.

2. Під час розіграшу 6 різних номерів можна вказувати за допомогою різних алгоритмів. Побудований в даній роботі алгоритм оптимально моделює роботу лототрона (звісно за умови чесного розіграшу): спочатку з масиву всіх впорядкованих необхідних номерів вибирається випадковий, на наступному кроці цей номер вилучається, одержуємо на один номер коротший масив, з яким аналогічно працюємо далі і т.д. Цей алгоритм продовжують до того часу, доки не буде вибрано 6 різних номерів. Подібний процес відбувається і тоді, коли ми хочемо, щоб комп'ютер сам зробив ставки (  АВТО ) для звичайної або системної гри.

**Аналіз останніх досліджень та публікацій.** У праці [-1] вивчено питання про генератор псевдовипадкових чисел. Щодо ADO-технології та OLE-технології, то ці теорії досить повно роз-

криті у працях відповідно [-2--5] та [-6, -7]. Основні аспекти мови запитів SQL розглянуто у книзі [-8].

**Мега роботи** полягає в проектуванні і написанні такого програмного комплексу, який би можна було ефективно використовувати для імітаційного моделювання симуляції розіграшів національної лотереї "Супер Лото".

**Методи розробки.** Для підключення БД MS Access до додатку була використана ADO-технологія. Для підключення до додатку табличного процесора MS Excel була використана OLE-технологія.

**Виклад основного матеріалу дослідження.** Опишемо інтерфейс користувача програми, розробленої засобами C++ Builder [-2--7] та мови запитів SQL [-8]. Даний програмний проект складається з трьох форм: стартової, головної та форми-редактора ставок.

Стартова форма (рис 1) призначена лише для встановлення максимальної кількості кульок.

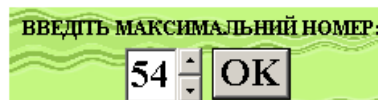


Рис. 1. Стартова форма

В головній формі (рис 2) відображаються всі ставки зроблені гравцями та дані про гравців. В ній проходить процес симуляції розіграшу, виводиться інформація про результат, яку в разі потреби можна занести у звіт (xls-файл). Крім того з цієї форми можна викликати форму-редактор ставок. Передбачено автоматичне сортування наборів номерів: як ставок, так розіграшу; автоматичне виправлення помилок: в разі, якщо користувач неправильно зробив ставки; є можливість перезавантаження, якщо користувач захоче самостійно в базі даних виправити свої помилки.



Рис. 2. Головна форма

У будь-який час користувач може внести зміни або доповнити базу даних, яку опишемо нижче, та перезавантажити головну форму. Але бувають такі випадки, наприклад, системна гра або режим  АВТО, коли користувачу для заповнення бази даних необхідна допомога. Саме для цього і створено форму-редактор ставок (рис 3). Тут спочатку необхідно вказати гравця (причому можна додати нового), а далі вручну або за допомогою режиму  АВТО створити ставку. Для ручного створення також працює автоматичне виправлення помилок.

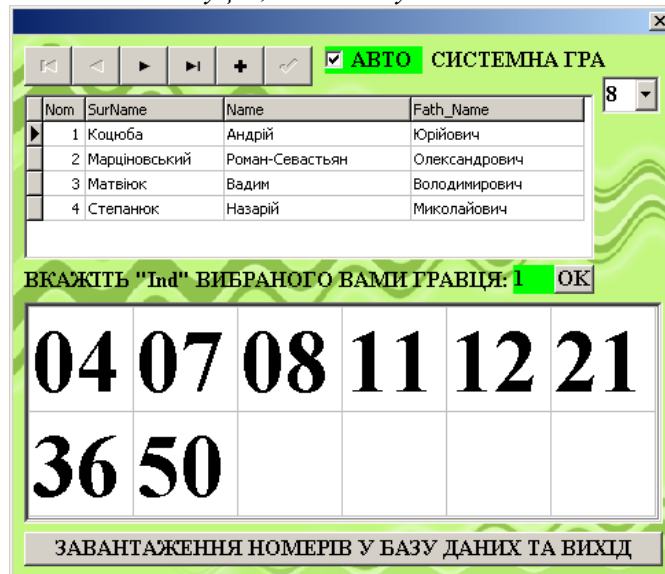


Рис. 3. Форма-редактор ставок

Після натискання на кнопку “Завантаження номерів у базу даних та вихід” буде для першого гравця створено відповідно ще 28 ставок з різних 6 номерів, відсортованих у порядку зростання, закриється ця форма і відкриється перезавантажена головна форма.

Відповідна база даних складається з двох зв’язаних таблиць (рис 4) та запиту, який містить об’єднану з цих таблиць інформацію та відображається у головній формі. Щодо таблиць, то в першу вносяться гравці (вона відображається у формі-редакторі записів), а в другу – ставки і вибираються гравці, які зробили ці ставки.

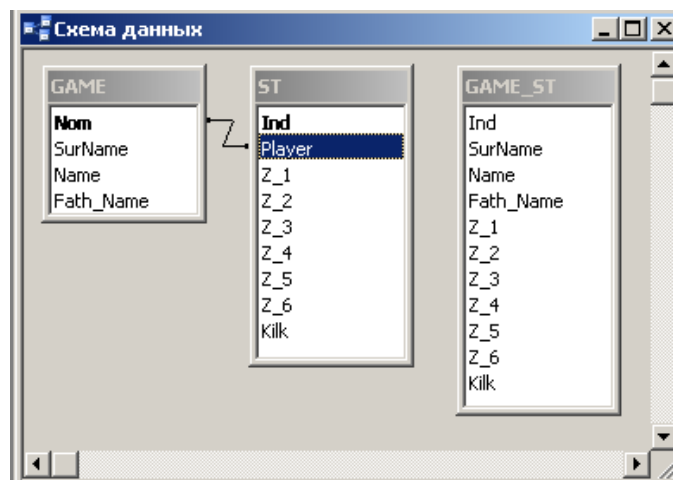


Рис. 5. Фізична модель бази даних

Звернемо увагу на ще один досить цікавий факт ADO-технології. Як правило за допомогою цієї технології загрузка бази даних є стаціонарною, тобто потрібно наперед вказати адресу mdb-файла і якщо на іншому комп’ютері за такою адресою нема бази даних, то exe-файл так як потрібно працювати не буде. Нами ж в цій роботі запропоновано дещо інший підхід:

- по-перше, для ефективної роботи даної технології використані наступні компоненти: TADODConnection, TDataSource, TADODDataSet, TADOQuery, TADODCommand, TDBNavigator, TDBGrid;

- по-друге, перед початковим завантаженням або перезавантаженням форми, на якій буде відображено інформація з бази даних (залежно від SQL-запиту), необхідно встановити у властивості Connected компоненти TADODConnection значення false (це лишнім не буде);

- по-третє, представимо фрагмент коду програми, у якому відловлюється поточна адреса exe-файла і приймається, що вона є адресою mdb-файла:

```
ADODConn->Connected = false;
```

```
String filename = "BAZZA.mdb";
if (!FileExists(ExtractFilePath(Application->ExeName)+filename)) {
    AnsiString ds="Файла бази даних не виявлено \n\n\t" +filename;
    ShowMessage(ds); }
else {
    String WayToBase=ExtractFilePath(Application->ExeName)+filename;
    ADOConn->ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;
User ID=Admin; Data Source="+WayToBase+"; Mode=Share Deny None; Extended
Properties="""; Jet OLEDB:System database="""; Jet OLEDB:Registry
Path="""; Jet OLEDB:Database Password=""";Jet OLEDB:Engine Type=5;
Jet OLEDB:Database Locking Mode=1; Jet OLEDB:Global Partial Bulk Ops=2;
Jet OLEDB:Global Bulk Transactions=1; Jet OLEDB:New Database Password
="""; Jet OLEDB:Create System Database=False; Jet OLEDB:Encrypt Data-
base=False; Jet OLEDB:Don't Copy Locale on Compact=False; Jet OLEDB:
Compact Without Replica Repair=False;Jet OLEDB:SFP=False;";
    ADOConn->Connected = true;
    ADODataSet->Active = true;
    ADODataSet->Open(); }
```

**Висновки.** Як правило імітаційне моделювання є найбільш ефективним, коли аналітичне дослідження ускладнене або неможливе. За допомогою системи диференціальних рівнянь для ймовірностей станів або рівнянь алгебри для ймовірності станів відповісти на вище поставленні питання досить важко, а деколи і неможливо. Тому доцільно використовувати ідею методу Монте-Карло, яка є надзвичайно простою і полягає у тому, щоб випадковий процес моделювати за допомогою так званих “розіграшів”, тобто за допомогою деякої процедури, яка дає випадковий результат. Особливо ця методика оправдовує себе, коли можна зробити таке програмне за безпечення, яке дасть змогу проробляти велику кількість “розіграшів” і аналізувати їх засобами математичної статистики.

1. Герасимчук О. І. Генератори псевдовипадкових чисел, їх застосування, класифікація, основні методи побудови і оцінка якості / О. І. Герасимчук, В. М. Максимович // Науково-технічний журнал “Захист інформації”, 2003, №3, С. 29-36.

2. Пахомов Б. Самоучитель C/C++ и C++ Builder 2007 / Б. Пахомов. – Санкт-Петербург: “БХВ Петербург”, 2008. – 672 с.

3. Послед Б. С. C++ Builder 6. Разработка приложений баз данных / Б. С. Послед. – Санкт-Петербург: “ДиаСофт”, 2003. – 310 с.

4. Хоменко А. Работа с базами данных в C++ Builder / А. Хоменко, С. Ададунов. – Санкт-Петербург: БХВ-Петербург, 2006. – 475 с.

5. Матеріали сайту [www.h-l-l.ru/publ/28-1-0-185](http://www.h-l-l.ru/publ/28-1-0-185), 11.11.2011 р.

6. Бруно Бабэ. Просто и ясно о Borland C++ / Бабэ Бруно. – М., 1996. – 400 с.

7. Матеріали сайту [www.realcoding.net](http://www.realcoding.net), 11.05.2007 р.

8. Кириллов В. В. Структуризованный язык запросов (SQL) / В. В. Кириллов. – Санкт-Петербург: ИТМО, 2005. – 80 с.