

КОМБІНОВАНА ТЕСТОВА ПРОГРАМА ЯК ЗАСІБ АВТОМАТИЗОВАНОГО НАВЧАННЯ І КОНТРОЛЮ ЗНАНЬ

У статті розглянуто засоби й методи автоматизованого навчання і контролю знань. Описано створення і використання програми для тестового контролю знань.

Можливості Delphi не обмежуються створенням складних програм. Delphi володіє вбудованими функціями з можливістю створювати свої власні й комбінувати існуючі функції. Найцікавішими функціями є логічні, котрі разом з функціями обробки тексту й арифметичними функціями дозволяють створювати тестові програми з будь-якого предмету. Такі програми можуть створювати не тільки вчителі інформатики, але й учні, що володіють Delphi на початковому рівні. Ці тести можуть бути використані на спільних уроках з вчителями-предметниками.

У якості прикладу наведемо створену комбіновану тестову програму, яка включає запитання з деяких навчальних предметів, з метою продемонструвати принципіальні можливості Delphi і відносну простоту створення таких програм.

Постановка задачі. Створити проект, що складається з кількох форм: головної форми, форми введення даних про особу, що тестується, форми для виведення результатів тестування. Залежно від правильності відповідей на тест, який містить три рівні (у свою чергу в рівні може бути кілька питань), підраховується результат і виставляється загальна оцінка.

Вимоги до підготовки папки з тестами та критерії оцінювання. Насамперед зазначимо, що програма повинна мати свій (активний) каталог, у якому мають міститися: тести, критерії оцінювання, та каталог, у який мають записуватися результати кожного тестування.

Будуватимемо ці каталоги у такій ієрархічній послідовності:

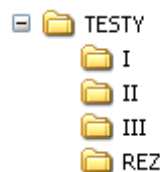


Рис. 1

Назва головного каталога, каталогів з тестами (I, II, III — каталоги з відповідними рівнями) і каталог з результатами повинні співпадати з назвами, зображених на рисунку 1. Також у каталозі «TESTY» повинен міститись текстовий файл з назвою Oc.in.txt, у якому записані такі дані:

1. У першій стрічці файлу — кількість запитань, які повинна згенерувати програма з каталога першого рівня, через пробіл — кількість балів, що присвоюється за правильну відповідь на одне запитання (тестове завдання).

2. У другій стрічці файлу — кількість запитань, які повинна згенерувати програма з каталога другого рівня, через пробіл — кількість балів, що присвоюється за правильну відповідь на одне запитання (тестове завдання).

3. У третій стрічці файлу — кількість запитань, які повинна згенерувати програма з каталога третього рівня, через пробіл — кількість балів, що присвоюється за правильну відповідь на одне запитання (тестове завдання).

Наприклад:

6 2

5 3

1 5

У каталозі рівня повинні міститися такі файли:

1. Файл з назвою A, де $A = [I, II, III]$ — рівень складності завдань (залежно від каталога рівня).

2. У кожному файлі *A* певна кількість файлів (Microsoft Word) (заготовлених учителем і пронумерованих за назвою — 1, 2, 3, 4,...), у кожному з них містяться завдання і чотири варіанти відповіді.

3. У кожному файлі *A* повинен бути текстовий документ, заповнений так:

Завдання №1 {номер завдання}
 4 {кількість варіантів відповідей}
 1 {номер правильного варіанту}
 а) {варіант відповіді}
 б) {варіант відповіді}
 в) {варіант відповіді}
 г) {варіант відповіді}
 Завдання №2
 4
 1
 а)
 б)
 в)
 г)
 і т.д.

Аналогічні файли повинні міститись у всіх каталогах рівня.

План розробки програми

1. Відкрити новий проект.
2. Створити головну форму.
- 2.1. Оформити форму згідно рисунка 2 (див. нижче).
3. Розмістити на формі такі об'єкти:
 - Кнопки BitBtn (4 шт.) і надати їм вигляду згідно рисунка.
 - Об'єкти GruopBox1 та RadioGroup1 і надати їм вигляду згідно рисунка.
 - Об'єкти MainMenu1 та OpenFileDialog1. Перший об'єкт оформити згідно рисунка, а також додати кнопку «Вихід» та запрограмувати її відповідно.
4. Розмістити на формі об'єкт OleContainer1.

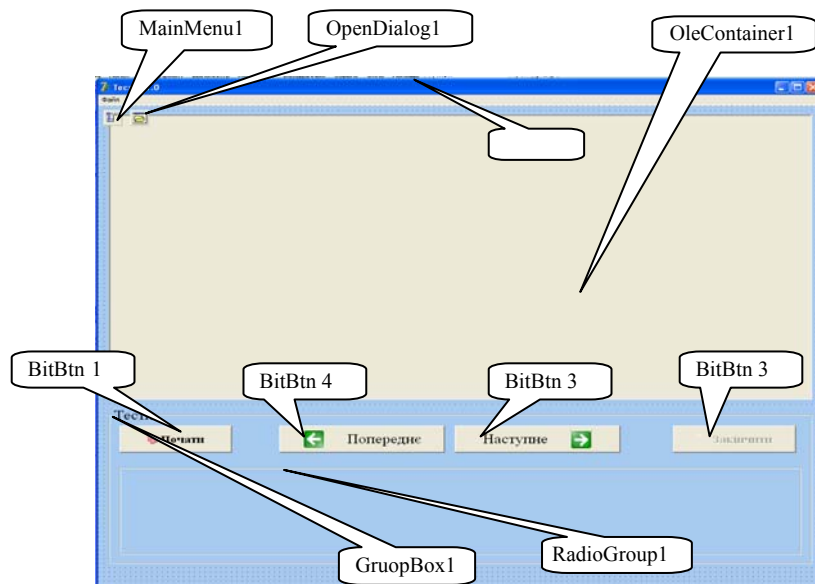


Рис. 2

5. Створити у тому ж проекті ще дві форми Form2 та Form3 і оформити їх згідно рисунків 3 та 4 відповідно.

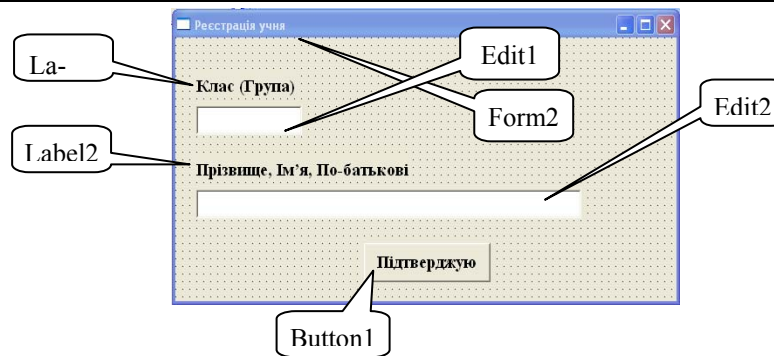


Рис. 3

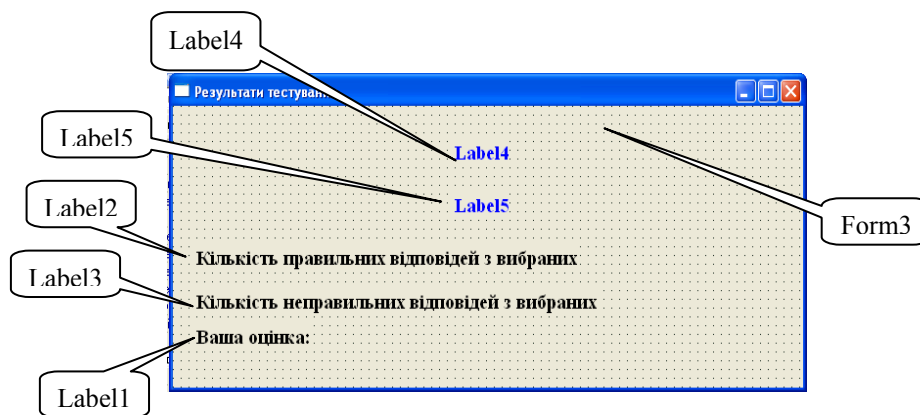


Рис. 4

6. Зберегти всі зроблені зміни та новостворені модулі в одній папці з проектом.

7. Описати модуль Unit1, який несе інформацію про одне тестове запитання.

8.1. Першою процедурою цього модуля записати FormActivate. Вона повинна викликати форму, де той, хто тестується, повинен вказати свої дані, після чого можна приступати до тестування.

8.2. Другою процедурою є перевірка існування всіх потрібних для роботи програми файлів і каталогів. Для цього використати стандартні процедури FileExists та DirectoryExists; Змінна catalog — шлях до каталога з тестами.

8.3. Запрограмувати процедуру **FormCreate** для форми. Процедура повинна заповнити значення змінної catalog, якщо його не було збережено раніше у файлі www.txt, що має міститися в активному каталозі програми. Іншими словами, якщо програма була завантажена вперше, то користувачеві прийдеється вказати місцезнаходження каталога із запитаннями, після чого програма створить файл www.txt, у якому буде записано значення шляху, де міститься каталог з тестами. Якщо ж програма запускається не вперше, то значення змінної catalog повинно братися з уже існуючого файлу www.txt.

8.4. Створити та запрограмувати процедуру, яка б дозволяла переміщуватися по вже згенерованих запитаннях і залежно від аргументу (змінної *an*) виводила б у форму варіанти відповідей на запитання за номером *an*. Слід зазначити, що у цій процедурі змінна *an* не та ж, що глобальна змінна AN, а також заповнювати (завантажувати) у OleContainer1 відповідний файл з відповідного каталога рівня та виводити його у поле OleContainer1.

8.5. Запрограмувати кнопку «Почати», яка повинна виконати такі операції:

- перевірити існування всіх каталогів, потрібних програмі;

- заповнити масив усіх запитань (fl), перевірити правильність заповнення файлів A given, де A = [I, II, III] (залежно від каталога рівня);
- заповнити значення змінних: KIR, KIIR, KIPIR, k1, k2, k3, o1, o2, o3, масив zaput, MAN, AN, ZAVD_NOM;
- згенерувати відповідно до файла Ocin.txt відповідну кількість запитань;
- активувати OleContainer1;
- запустити на виконання процедуру ZM_AN з початковим аргументом AN = 1.

8.6. Повернутися до кнопок «Попереднє» та «Наступнє» (BitBtn4, BitBtn3). Маючи процедуру ZM_AN, нам неважко буде запрограмувати події, що повинні виконуватися при натисканні на ці кнопки: TForm1.BitBtn3Click — завантажується наступне запитання, TForm1.BitBtn4Click — попереднє запитання.

Нам доведеться щоразу перевіряти, чи не завершився процес тестування (вибрано всі тести), тому потрібно реалізувати процедуру (all), яка б перевіряла, чи в усіх згенерованих програмою тестах користувач вибрав хоча б один варіант відповіді. У такому випадку кнопка «Завершити» залишалася б неактивною.

8.7. Потрібно також реалізувати процедуру, яка б реагувала на дії користувача під час вибору варіанта відповіді, тобто заносила б значення вибраного варіанта у змінну fl[..].nvv;

8.8. Потрібно реалізувати процедуру, яка б змогла провести підрахунок балів і вивести результати на екран, а також записати їх в окремому файлі у каталозі TESTY\REZ\

9. Розглянути модуль Unit2, що стосується Form2 — форми, яка використовується для введення даних про користувача, що тестується.

При натисканні кнопки «Підтверджую» (Button1) повинна виконатись перевірка на цілісність даних, що були введені користувачем.

Якщо користувач не має бажання вводити свої дані ((Edit1.Text="" and Edit2.Text="")), то не допустити його до тестування (Form1.Close).

10. Розглянути модуль Unit3, що стосується Form3 — форми, яка використовується для виведення даних про користувача, та результати тестування.

Реалізувати процедуру, що повинна виконатися при закритті форми.

Висновок. Програмний продукт пройшов апробацію і може бути успішно використаний для контролю знань учнів у загальноосвітній школі.

ЛІТЕРАТУРА

1. Моисеев В. Б., Усманов В. В., Таранцева К. Р., Пятирублевый Л. Г. Статистический подход к принятию решений по результатам тестирования для тестов открытой формы // Дистанционное образование. — 2001. — №1.
2. Кузовлева К. Т. Конструирование педагогических тестов на основе современных математических моделей //www.informika.ru.
3. Гусева А. И., Шпурик А. П. Дистанционная система тренинга и адаптивного тестирования ВИОТ-2 //Труди X міжнародного науково-технічного семінара «Современные технологии в задачах управления, автоматизации и обработки информации». — Алушта, 2001.
4. Бобровский С. И. Delphi 7: Учебный курс. — СПб.: Питер, 2004. — 736 с.
5. Давыдов Е. В. Создание базовых элементов интерфейса Windows — программы в среде Delphi // Информатика и образование. — 2002. — №4. — С. 21–30.
6. Еремин Е. А. Почему системам Delphi интересна для образование. // Информатика и образование. — 1997. — №1. — 569 с.
7. Федоров А. Г. Создание Windows-приложений в среде Delphi. — М.; ТОО Компьютер-Пресс, 1995.