

# ПРИНЦИПИ РЕАЛІЗАЦІЇ ТЕСТОВОГО КОНТРОЛЮ ЗАСВОЄННЯ ЗНАТЬ ЗА ДОПОМОГОЮ КОМП'ЮТЕРНИХ ПРОГРАМ

УДК 681.3.06:51

*Микола Дорошенко, кандидат фізико-математичних наук,  
доцент кафедри інформатики та обчислювальної математики  
Дрогобицького державного педагогічного університету  
імені Івана Франка*

## ПРИНЦИПИ РЕАЛІЗАЦІЇ ТЕСТОВОГО КОНТРОЛЮ ЗАСВОЄННЯ ЗНАТЬ ЗА ДОПОМОГОЮ КОМП'ЮТЕРНИХ ПРОГРАМ

*У статті розглядається структура файлу тесту та принципи реалізації тестового контролю, що дозволяють створювати універсальні тестуючі програми засобами об'єктно-орієнтованого програмування.*

*Ключові слова: тест, файл тесту, програмоване навчання, об'єктно-орієнтоване програмування.*

*Лит. 8.*

**П**остановка проблеми. Контроль за знаннями учнів та студентів набув широкого поширення завдяки впровадженню модульно-рейтингової системи навчання та проведенню незалежного тестування при вступі до вищих навчальних закладів. Доцільно, щоб уся система тестування була комп'ютеризованою, тому що це допоможе прискорити процес тестування та повністю виключити суб'єктивний фактор при його проведенні.

Поступовий перехід від традиційних форм контролю й оцінювання знань до комп'ютерного тестування відповідає тенденціям сучасності та загальній концепції модернізації та комп'ютеризації української системи освіти. Її ефективність багато в чому залежить насамперед від специфіки самої навчальної дисципліни та мети навчання; від якості програмних продуктів і доцільності їх використання для конкретних навчальних цілей; а також від форм представлення навчальної інформації (зокрема, від рівня її візуалізації).

Порівняно з традиційними формами контролю комп'ютерне тестування (КТ) має ряд переваг:

- швидке одержання результатів іспиту і звільнення викладача від трудомісткої роботи з обробки результатів тестування;
- об'єктивність в оцінці;
- конфіденційність при анонімному тестуванні;
- тестування на комп'ютері більш цікаве у порівнянні з традиційними формами опитування, що створює позитивну мотивацію у студентів.

З огляду на це розробка різноманітних програмних засобів для підготовки й організації тестування з використанням комп'ютера є актуальною.

Крім того, на теперішній час є чимала кількість тестуючих комп'ютерних програм з різних навчальних дисциплін. На жаль майже всі з таких програмних продуктів є російськомовними і розроблені для одного предмета. Як наслідок,

виникає ряд проблем з придбанням таких програм, їх експлуатацією, оновленням навчальної інформації. Саме тому постає потреба у розробці універсальної українськомовної програмної оболонки.

**Аналіз останніх досліджень та публікацій.** Комп'ютерній реалізації тестування та створенню українськомовного універсального навчального програмного забезпечення присвячено ряд робіт. У роботах [2, 5] розглянуто принципи створення універсальних тестуючих програм. У роботі [3] для автоматизації тестування використовується засіб XML, який дозволяє створювати ефективний моніторинг викладача, а роботах [4, 8] розглядається методика створення ефективної українськомовної програмної оболонки для вивчення математики.

**Мета статті.** Розробка структури файлу тесту та принципів реалізації тестового контролю знань, які дозволяють створювати універсальні тестуючі програми за допомогою об'єктно-орієнтованих мов програмування.

**Виклад основного матеріалу.** Розглянемо основні принципи реалізації тестового контролю в навчальних комп'ютерних програмах, які створюються за допомогою візуальних об'єктно-орієнтованих мов програмування [7].

На даному етапі найбільшого поширення набули такі варіанти тестових контрольних завдань – процедур [6]:

1. “автоматичний”, коли той, якого навчають, виконує завдання в безпосередньому діалозі з ПЕОМ, результати відразу переносяться в блок обробки;
2. “напівавтоматичний”, коли завдання виконуються письмово, а відповіді зі спеціальних бланків вводяться в ПЕОМ (рішення не перевіряються);
3. “автоматизований”, коли завдання виконуються письмово, рішення перевіряються викладачем, а в ПЕОМ вводяться результати перевірки.

## ПРИНЦИПИ РЕАЛІЗАЦІЇ ТЕСТОВОГО КОНТРОЛЮ ЗАСВОЄННЯ ЗНАТЬ ЗА ДОПОМОГОЮ КОМП'ЮТЕРНИХ ПРОГРАМ

Контролюючі тестові завдання також виконують коректуючу, навчальну та діагностичну функції. Коректуюча функція полягає в тому, що, фіксуючи і локалізуючи помилки, контролююча програма допомагає учневі ліквідувати їх. Навчальна роль тестових завдань полягає в тому, що на основі виявлених нею помилок і утруднень можна визначити кількість і характер завдань та вправ, необхідних для ліквідації прогалин у знаннях студента з даної теми. На основі даних тестових завдань про легкість, швидкість, повноту і міцність засвоєння знань можна зробити мотивований висновок про індивідуальні можливості студента. В цьому полягає діагностична функція тестових завдань.

Для того, щоб реалізувати комп'ютерне тестування виконати такі етапи роботи:

1. Вибір типу тестових завдань.
2. Формування змісту тестових завдань та розподіл їх за типами та рівнями складності.
3. Написання сценарію тестування та його формалізація.
4. Розробка структури програмного файлу тесту та реалізація його на ПЕОМ.
5. Вибір мови програмування та програмна реалізація тестового контролю.

Для реалізації тестового контролю можна вибирати такі типи тестів:

1. Тип А (закритий однозначний) – це тип тестових завдань з вибором єдиної правильної відповіді з декількох запропонованих варіантів.
2. Тип В (відкритий однозначний) – це тип тестових завдань із введення єдиної правильної відповіді.
3. Тип С (закритий багатозначний) – тип тестових завдань із множинним вибором відповідей.
4. Тип D (питання на відповідність) – тип тестових питань з підбором пар відповідностей, зіставленням або протиставлення елементів двох представлених множин.

Зміст і постановка тестових питань повинні забезпечувати валідність і надійність тестових завдань і всього тесту в цілому.

Сценарій тестування повинен містити всі повідомлення комп'ютера та можливі відповіді студента. Формалізація сценарію тестування відбувається таким чином.

Множину тестових питань позначимо  $M$ , а множину відповідей студента позначимо  $N$ .

Тоді тестові завдання можна представити як два типи відповідностей множин  $M$  і  $N$ , а саме:

- 1) проста відповідність  $(1; n)$ , де  $n$  – кількість елементів множини  $N$ . Тобто, одному елементу множини  $M$  відповідає  $n$  елементів множини  $N$ .

За допомогою простої відповідності формалізуються сценарії тестування, створені за допомогою тестів типів А, В та С відповідно;

- 2) взаємно-однозначна відповідність  $(n; n)$ , де  $n$  – кількості елементів множин  $M$  та  $N$ . Відповідні один одному елементи множин  $M$  та  $N$  ідентифікуються одним і тим порядковим номером або протиставленням елементів. За допомогою взаємно-однозначної відповідності формалізуються сценарії тестування, які створені за допомогою тесту типу D.

Перший тип відповідності реалізується на ПЕОМ так: на екран дисплея видаються питання та варіанти відповідей.

Прикладом другого типу відповідності може бути таке:

- 1) встановити відповідність між математичними формулами та їх назвами, які в довільному порядку виводяться на екран дисплея;

- 2) є деякий набір речень на рідній мові, які представляють собою логічно зв'язаний текст і відповідна йому множини неупорядкованих речень на іноземній мові; потрібно розмістити речення на іноземній мові в правильному порядку.

Слід зазначити, що всю інформацію, яка забезпечує контроль (питання, варіанти відповідей та правильні відповіді), доцільно зберігати в одному файлі, а саме: програмну частину тесту. Тому при реалізації тестового контролю на ПЕОМ важливим є розробка структури файлу тесту.

Файл програмного тесту для тестів типу А представляє собою послідовність питань, на які потрібно відповісти шляхом вибору правильної відповіді з декількох варіантів.

Файл програмного тесту пропонується реалізувати таким чином.

Файл тесту складається з трьох розділів:

- Розділ заголовку.
- Розділ оцінок.
- Розділ питань.

Починається файл тесту, як правило заголовком, який дає загальну інформацію про тест, наприклад, про його призначення. Ознакою кінця заголовку може бути крапка, яка стоїть на початку наступного рядка.

За заголовком тесту слідує розділ оцінок. В цьому розділі повинні бути перелічені оціночні рівні та кількість балів, які може отримати студент.

За розділом оцінок слідує розділ контролюючих завдань тесту. Кожне тестове завдання починається текстом питання, за яким з окремого рядка може слідувати ім'я файлу ілюстрації, яке починається з символу '\'. Ім'я файлу є ознакою кінця питання. Якщо до тестового завдання немає ілюстрації, то замість імені файлу ставиться крапка.

## ПРИНЦИПИ РЕАЛІЗАЦІЇ ТЕСТОВОГО КОНТРОЛЮ ЗАСВОЄННЯ ЗНАТЬ ЗА ДОПОМОГОЮ КОМП'ЮТЕРНИХ ПРОГРАМ

Далі слідують альтернативні відповіді. За текстом кожної альтернативної відповіді на наступному рядку розміщується кількість балів. Якщо альтернативна відповідь не є останньою для біжучого питання, то перед кількістю балів ставиться кома, якщо останньою, то – крапка.

У файлі тесту наведеної структури контролююче тестове завдання представляється так:

*Текст тестового питання*

*Неправильна відповідь*

*,0*

*Неправильна відповідь*

*,0*

*Правильна відповідь*

*,1*

*Неправильна відповідь*

*.0*

Такий файл тесту створюється в текстовому редакторі Блокнот і дозволяє проектувати універсальні програми, які не залежать від кількості тестових завдань та їх змісту.

Важливою проблемою при реалізації тестового контролю є захист файлу тесту від несанкціонованого доступу. Файл тесту повинен бути доступний тільки адміністратору (викладачу). Тільки адміністратор може змінювати або редагувати тестові завдання. Файл тесту, який створений описаним вище методом, є не захищеним від несанкціонованого доступу. Одним із шляхів вирішення даної проблеми при реалізації тестового контролю мовою програмування Object Pascal є представлення файлу тесту як типізованого файлу, в якому тестове завдання представляється такою змінною типу Record:

*test=record*

*question: array[1..4] of string; { текст контролюючого питання }*

*variant: array[1..5] of string; {- варіанти відповідей }*

*n\_virto: integer; { номер правильної відповіді }*

*end.*

Такі тестові контролюючі завдання створюються програмно адміністратором, використовуючи потрібні стандартні компоненти системи візуального програмування Delphi та зберігаються на магнітному диску у вигляді типізованого файлу, записи якого є величинами типу test. Змінити або редагувати тестові завдання за допомогою текстових редакторів неможливо, бо такі дії можуть привести тільки до пошкодження даного файлу тесту.

Для реалізації тестового контролю на ПЕОМ може бути вибрана система візуального програмування Delphi, в якій код програми

створюється на об'єктно-орієнтованій мові програмування Object Pascal, яка має істотні переваги при розробці програмних систем, а саме: можливість використання візуального методу розробки програм, а також багаторазово використовувати програмний код.

В результаті аналізу різних тестів, на основі принципів програмованого навчання [1] та об'єктно-орієнтованого підходу до створення комп'ютерних програм можна сформулювати такі вимоги до реалізації комп'ютерного тестування:

1. Програма забезпечує роботу з тестом довільної довжини, тобто, без обмеження на кількість завдань.

2. Створює тести різного типу і змісту та використовує можливості представлення символічної та графічної інформації.

3. Проводить тестування на сучасному рівні, з дотриманням конфіденційності та у привабливій формі.

4. Тестові завдання знаходяться в одному програмному файлі тесту.

5. Програма є інваріантна до різних тестів, тобто, зміни в тестах не несуть за собою суттєві зміни в програмі.

6. Виконання кожного тесту починається з "заставки", на якій представлена інформація про тему тесту та конкретні питання навчальної програми, які підлягають перевірці.

7. Не можна здійснити повернення до попереднього тестового завдання.

8. На екрані дисплея з'являється тільки одне тестове завдання. Інше завдання з'являється тоді, коли дана відповідь на попереднє завдання або вичерпався час відведений на виконання завдання.

9. В залежності від порядкового номера студента, кожному студенту програма формує свій набір тестових завдань. В системі візуального програмування Delphi алгоритм формування різного набору тестових завдань реалізується з використанням стандартних функцій роботи з псевдовипадковими числами.

10. Адміністратор, тобто викладач, може модифікувати тестові завдання в процесі виконання програми, а також змінювати час відведений на виконання завдання.

11. Після завершення контролю виставляється оцінка (згідно параметрів контролю), визначається коефіцієнт успішності студента, видається діаграма співвідношень між правильними та неправильними відповідями.

12. При виконанні програми в режимі "Навчання" студенту надається можливість встановлювати параметри контролю, а саме: час, що відведений для виконання тесту; показ правильної відповіді.

## КОНЦЕПТУАЛЬНІ ЗАСАДИ ІНШОМОВНОЇ ПОЛІТИКИ ЄС ТА ЇХ ПРАКТИЧНЕ ВТІЛЕННЯ У ОСВІТНІЙ СФЕРІ

Для експлуатації EXE-файлу програми потрібно, щоб на комп'ютері було встановлено мінімальне програмне забезпечення, саме: тільки операційна система WINDOWS.

**Висновок.** Запропоновані структура файлу тесту та принципи реалізації тестового контролю знань дозволяють створювати універсальні програми тестування засобами об'єктно-орієнтованих мов програмування, які не залежать від кількості тестових завдань, їх змісту та захищені від несанкціонованого доступу. Таким чином, за допомогою комп'ютера викладач може досить швидко побудувати систему тестових завдань. При цьому значно полегшується процес обробки результатів і виставлення оцінок.

1. Беспалов В.П. Программированное обучение. Дидактические основы. – М.: Высшая школа, 1970. – 300 с.

2. Дубровін О., Дубровін М. Розробка структури

універсальної тестуючої системи // Молодь і ринок. – 2008. – № 1. – С. 45 – 49.

3. Дубровін О.Ф. Розробка комп'ютерної системи тестування на базі XML // Актуальні проблеми фізики, математики та інформатики. – 2009. – № 1. – С. 74 – 77.

4. Кардаш А.І., Лисишин Ю.Я. Навчальна програма на базі шкільного курсу тригонометрії // Матеріали Всеукраїнської наукової конференції “Сучасні проблеми прикладної математики та інформатики”. – Львів, 2003. – С. 66 – 68.

5. Любченко К.М. Інструментально-контролююча програма Master of Logig. // Комп'ютер в школі та сім'ї. – 2000. – № 2. – С. 34 – 36.

6. Олійник М.М. Тест як інструмент кількісної діагностики рівня знань в сучасних технологіях навчання. – Донецьк: ДонНУ, 2001. – 84 с.

7. Синтес А. Объектно-ориентированное программирование. – М.: Издательский дом “Вильямс”, 2002. – 672 с.

8. Триус Ю.В. Комп'ютерно-орієнтовані системи навчання математики: монографія / Юрій Васильович Триус. – Черкаси: Брама-Україна, 2005. – 400 с.

Стаття надійшла до редакції 15.08.2011

УДК 378

**Віта Гаманюк**, кандидат педагогічних наук,

доцент кафедри німецької мови і літератури з методикою викладання  
Криворізького державного педагогічного університету

## КОНЦЕПТУАЛЬНІ ЗАСАДИ ІНШОМОВНОЇ ПОЛІТИКИ ЄС ТА ЇХ ПРАКТИЧНЕ ВТІЛЕННЯ У ОСВІТНІЙ СФЕРІ

У статті розглядається діяльність структур ЄС на шляху до реалізації концептуальних засад політики багатомовності у сфері освіти та виявляються напрямки та інструменти її практичного втілення.

**Ключові слова:** іношомовна політика, багатомовність, білінгвальне навчання, адаптивна мова, навчання іноземних мов.

**Літ. 4.**

**Постановка проблеми.** Розвиток суспільства, потужні інтеграційні процеси, міграція населення та глобалізація усіх сфер життя спонукають до зміни пріоритетів у політиці, зокрема у іношомовній політиці країн ЄС, а це в свою чергу впливає на систему іношомовної підготовки фахівців. Останнім часом з'явилося чимало нових підходів до вирішення мовного питання у нових суспільних умовах та концепцій у сфері іношомовної освіти як у межах окремих країн, так і в межах євросони в цілому.

**Мета статті** – вивчити діяльність структур ЄС щодо практичного втілення концептуальних засад іношомовної політики та виявити основні напрямки та інструменти їх реалізації у сфері іношомовної освіти.

**Виклад основного матеріалу.** Підтримка політики багатомовності у ЄС розпочалася запровадженням у 1990 р. програми LINGUA, яка

мала на меті збільшення кількості та підвищення якості занять з іноземних мов. Вперше були сформульовані конкретні цілі, визначено низку заходів та засобів, почала проводитися цілеспрямована політика щодо іноземних мов. На європейському та національному рівні з'явилися структури, котрі систематично займалися питаннями мови та заходами по сприянню іношомовній підготовці. LINGUA охоплювала п'ять напрямків, а саме: співробітництво між закладами освіти та освітніми установами, які готують вчителів мов, індивідуальні дотації вчителям мов; сприяння підвищенню рівня знань іноземної мови у професійній сфері; обмін молодіжними групами з метою отримання іношомовної освіти; розробка навчальних посібників для тих мов, що найменше представлені у освітніх закладах як іноземні.

Згодом LINGUA була включена до програм SOKRATES та LEONARDO DA VINCI. Програма LEONARDO DA VINCI сприяє