



## Розробка класу «Студент» об'єктно орієнтованого простору для підсистеми моніторингу якості знань за кредитно-модульною системою навчання

*Микола Бондаренко,*  
кандидат технічних наук, професор,  
Українська інженерно-педагогічна академія, Харків

*Моніторинг якості знань у повноцінному вигляді міг з'явитися тільки з розвитком та широким впровадженням в управлінську діяльність комп'ютерної техніки та інформаційних технологій*

**М**оніторинг в загальному значенні цього слова означає спостереження і відстеження змін яких-небудь процесів у часі. Динаміка змін якості знань (об'єкт моніторингу) може бути помітною тільки на основі деякої її кількісної характеристики. Для цього розробляються критерії оцінювання якості знань, на базі яких можна її отримати. Після визначення таких критеріїв здійснюється безпосереднє оцінювання якості знань. Це — функція підсистеми контролю якості знань. У даній роботі розглядається розробка нового класу «Студент» для підсистеми моніторингу якості знань засобами системи Delphi. Актуальність роботи полягає в розробці методики нового класу «Студент» засобами системи Delphi. Моніторинг якості знань у рамках одного ВНЗ здійснюється на декількох рівнях — навчальної групи, кафедри, факультету, ВНЗ в цілому. На кожному з них інформація, отримана від попереднього рівня, аналізується та узагальнюється, і, якщо необхідно, приймаються відповідні рішення щодо поліпшення навчального процесу.

Одна з особливостей кредитно-модульної системи навчання, як системи

обліку початкових досягнень студентів, полягає в систематичній оцінці усіх видів їх роботи в навчальному процесі. До таких видів навчальних досягнень відносяться: відвідуваність занять; аудиторний контроль; виконання лабораторних робіт; виконання домашніх завдань.

Другою особливістю кредитно-модульної системи є розбиття усього курсу навчальної дисципліни на окремі модулі. По кожному модулю дається оцінка (бали) навчальних досягнень студента відповідно до їх видів.

Дані моніторингу на рівні навчальної групи доцільно поєднати відносно кожної дисципліни і окремого студента. Така задача може бути зручно вирішена шляхом створення нового класу «Студент». Об'єкт класу «Студент» має містити основні моніторингові дані про студента по кожній дисципліні — прізвище, ім'я та по-батькові студента (П.І.Б.), навчальну групу, назву дисципліни, бали по кожному виду навчальних досягнень і по кожному модулю.

Аналіз задачі показує, що задання таких балів за допомогою діалогових вікон дуже уповільнює введення інформації.

Таблиця 1

	Відвідування занять	Аудиторний контроль	Лаб. роботи	Домашні завдання	Разом по модулю
Модуль 1	5	6	8	6	25
Модуль 2	5	6	8	6	25
Модуль 3	5	6	8	6	25
Модуль 4	5	6	8	6	25
Разом	20	24	32	24	100

Її доцільно готувати засобами текстових редакторів та зберігати в окремому файлі для кожного студента. У самому класі має бути вказано посилання на цей файл.

Таким чином клас «Студент» має мати такі властивості:

- 1) Прізвище, ім'я та по-батькові студента (PIB). Тип даних — string.
- 2) Навчальна група (Grupa). Тип даних — string.
- 3) Назва дисципліни (Dis). Тип даних — string.
- 4) Ім'я або повний шлях до файлу (FileName) — файл, з якого має бути отримана моніторингова графічна інформація. Тип даних — string.

Як параметр окремої дисципліни в класі мають бути визначні обмежені максимальні бали, які може отримати студент по тому чи іншому виду навчальних досягнень (для кожного модуля).

Усі властивості мають повний доступ для запису та читання. Перевірка значень деяких властивостей недоцільна або взагалі неможлива (наприклад, PIB, Dis).

Властивість FileName має бути перевірена на коректність (файл має існувати). Моніторингові дані з цього файлу мають бути перевірені на неперевищення максимального допустимих значень.

Всі дані, які задані властивостями класу, мають бути (необов'язково) відображені на сторінці (формі). При цьому тільки користувач класу (розробник прикладного додатку має визначати, де саме на сторінці і у якому вигляді буде розміщена ця інформація.

У процесі розробки та тестування класу «Студент» для підсистеми моніторингу якості знань в середовищі Delphi було розроблено програмний код. Розробка була здійснена для дисципліни, яка складається з 4 модулів. Максимальні бали для кожного виду навчальних досягнень по кожному модулю наведені в табл. 1.

Для тестування класу було розроблено окремий проект в Delphi. До відповідного проекту в Delphi було додано розроблений модуль (unit) з класом TStud. Вигляд форми для тестування класу дано на рис. 1.

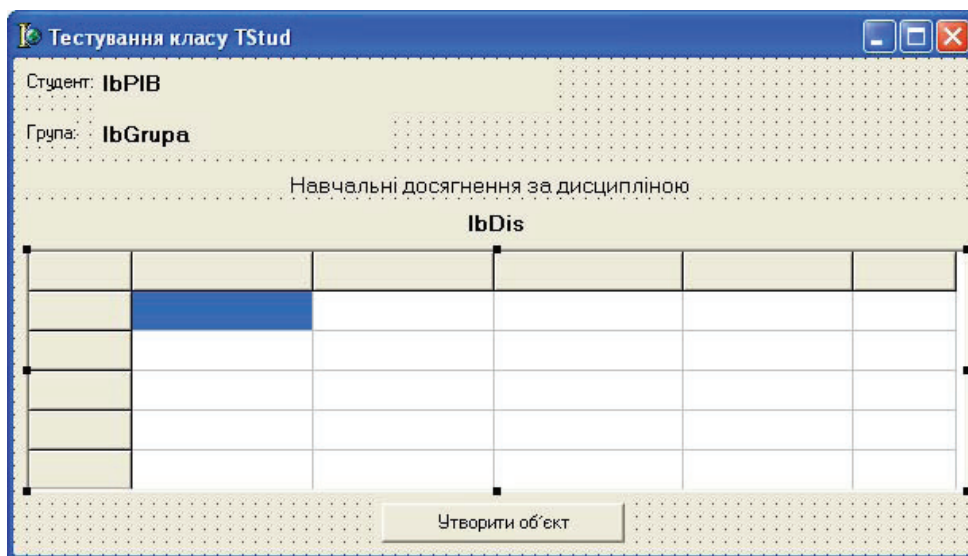


Рис. 1. Вигляд форми для тестування класу

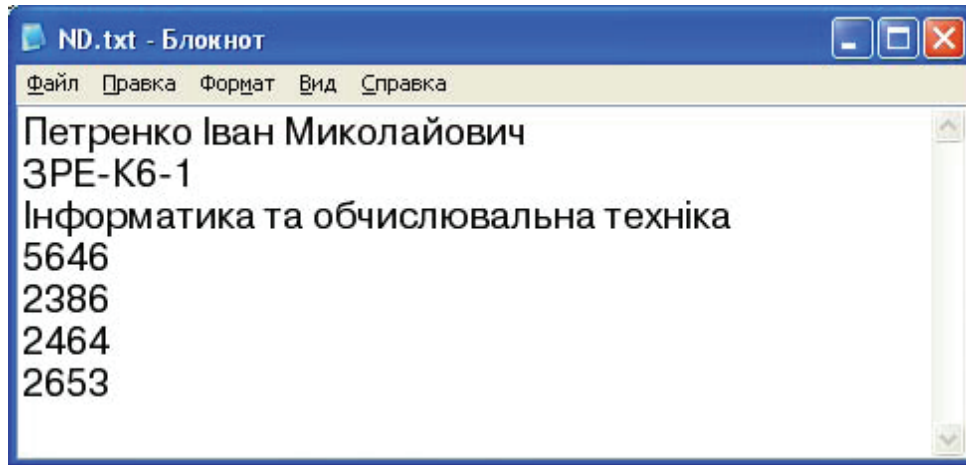


Рис. 2. Текстовий файл для тестування класу

Призначення компонентів:

lbPIB — мітка для відображення П. І. Б. студента;

lbGrupa — мітка для відображення назви навчальної групи;

lbDis — мітка для відображення назви дисципліни.

Командна кнопка «Утворити об'єкт» — для перевірки утворення об'єкта класу TStud, задання його властивостей та перевірки методів.

Під компонентом lbDis розташовано рядкову таблицю (клас TStringGrid), в якій буде відображатися інформація про навчальні досягнення студента по дисципліні.

Для зручності тестування класу всі дані для визначення об'єкта класу «Сту-

дент» було підготовлено у окремому текстовому файлі. Структура файлу така:

1-й рядок — П. І. Б. студента;

2-й рядок — назва навчальної групи;

3-й рядок — назва дисципліни;

4-й рядок — бали по видах навчальних досягнень по модулю 1;

5-й рядок — бали по видах навчальних досягнень по модулю 2;

6-й рядок — бали по видах навчальних досягнень по модулю 3;

7-й рядок — бали по видах навчальних досягнень по модулю 4.

Приклад такого файлу наведено на рис. 2.

Процедура працює таким чином. На початку утворюється новий об'єкт Stud за допомогою оператора Stud:=TStud.

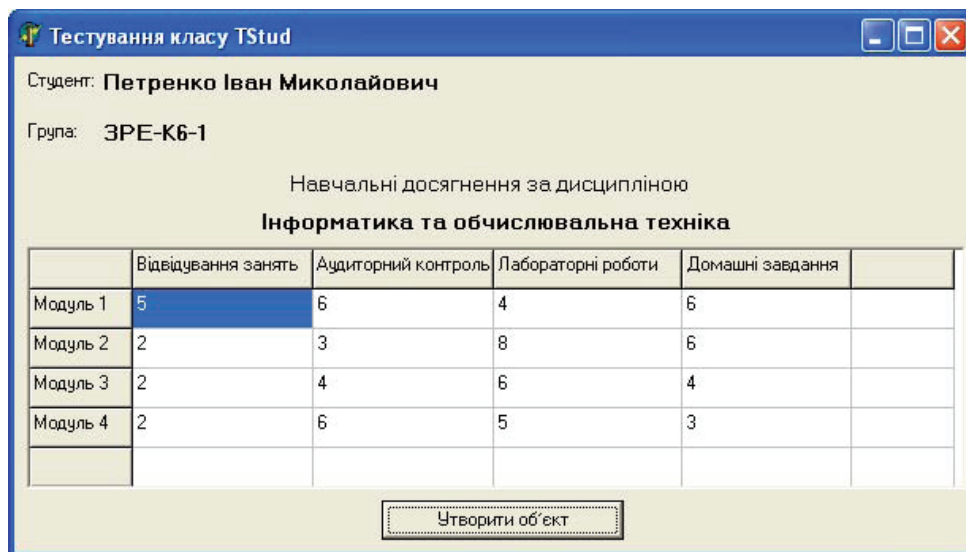


Рис. 3. Результат тестування (всі дані без помилок)

Create;. Далі задається ім'я файлу з даними (Stud.FileName:='ND.txt');

Потім до створеного об'єкта послідовно застосовуються всі методи класу.

Після запуску цієї програми та натискання на кнопку «Утворити об'єкт» було отримано такий результат (рис. 3).

Цей результат повністю відповідає очікуванням — саме такі значення були в файлі і саме на цих місцях форми вони мають бути розташовані. Тобто усі властивості та методи відпрацювали відповідно до вимог.

Після цього в програмному коді оператор Stud.FileName:='ND.txt'; було замінено на оператор Stud.FileName:='ABC.txt';, тобто зроблена спроба призначити властивості FileName ім'я файлу, який не існує.

Після запуску програми та натискання на кнопку «Утворити об'єкт» було отримано таке повідомлення (рис. 4).

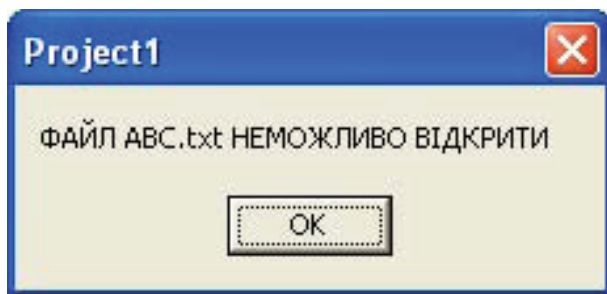


Рис. 4. Реакція на спробу призначити неіснуючий файл

Таким чином, процедура призначення властивості FileName правильно відреагувала на помилкове значення.

Далі було зроблено перевірку на спробу введення помилкового значення бала навчального досягнення, яке перевищує максимально допустиме. Для цього в файлі ND.txt (файл з даними) у 5-му рядку значення «2386» було замінено на значення «2389». Тобто виконання домашніх завдань по 2-му модулю оцінено в 9 балів, у той час як максимальна оцінка по цьому параметру складає 6 балів.

Після запуску програми та натискання на кнопку «Утворити об'єкт» було отримано таке повідомлення (рис. 5).

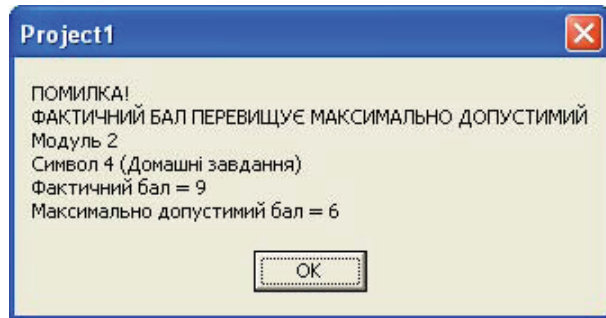


Рис. 5. Реакція на спробу ввести помилкове значення бала

Таким чином, процедура вводу даних про навчальні досягнення правильно відреагувала на помилкове значення.

Аналогічне повідомлення було отримано при помилковому значенні для іншого модуля ті іншого виду навчальних досягнень (рис. 6).

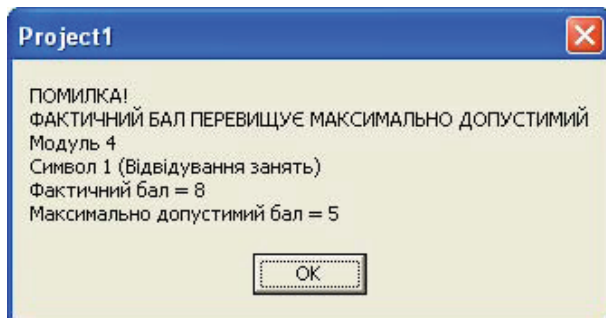


Рис. 6. Реакція на спробу ввести помилкове значення бала

У цьому прикладі було розроблено прикладний додаток для відображення моніторингової інформації для 4 студентів. Відповідна форма представлена на рис. 7.

Відповідні текстові файли було попередньо скопійовано у ту ж саму папку, де знаходиться додаток. Файли мають назву ND1.txt, ND2.txt, ND3.txt, ND4.txt.

Після запуску цього додатку було відображено дані по першому студенту (рис. 8).

Приклад використання класу TStud

Студент: **IbPIB**

Група: **IbGrupa**

Навчальні досягнення за дисципліною

**IbDis**



Попередній студент

Наступний студент

Рис. 7. Форма для прикладного додатку

Як бачимо, загальний бал студента Петренко І. М. по дисципліні «Інформатика та обчислювальна техніка» складає 72 бали, що відповідно до системи ECTS відповідає оцінці «задовільно» (але вже досить близько до «добре»). Його сильна сторона — виконання лабораторних робіт (23 бали). Слабка сторона — відвідування занять. Таким чином, студент не без здібностей, але недисциплінований.

Після натискання на кнопку «Попередній студент» нічого не змінюється, оскільки попереднього студента не існує. Після натискання на кнопку «Наступний студент» на формі відображається інша інформація (рис. 9).

Наступна перевірка показує, що наступні натискання на кнопки переходу призводять до правильного переходу від одного студента до іншого.

В роботі отримано такі основні результати.

1. Засобами середовища Delphi розроблено новий клас TStud («Студент») для підсистеми моніторингу якості знань при кредитно-модульній системі навчання.

2. Проведено тестування цього класу, яке показало правильність роботи властивостей та методів.

3. Розроблено прикладний додаток з використанням нового класу. Показано, що простота в програмуванні цього додат-

Приклад використання класу TStud

Студент: **Петренко Іван Миколайович**

Група: **ЗРЕ-К6-1**

Навчальні досягнення за дисципліною

**Інформатика та обчислювальна техніка**

	Відвідування занять	Аудиторний контроль	Лабораторні роботи	Домашні завдання	Разом
Модуль 1	5	6	4	6	21
Модуль 2	2	3	8	6	19
Модуль 3	2	4	6	4	16
Модуль 4	2	6	5	3	16
Разом	11	19	23	19	72

Попередній студент

Наступний студент

Рис. 8. Відображення інформації про першого студента

Приклад використання класу TStud

Студент: **Рибкін Микола Іванович**

Група: **ЗРЕ-К6-1**

Навчальні досягнення за дисципліною  
**Інформатика та обчислювальна техніка**

	Відвідування занять	Аудиторний контроль	Лабораторні роботи	Домашні завдання	Разом
Модуль 1	3	1	2	4	10
Модуль 2	2	3	8	6	19
Модуль 3	3	4	6	4	17
Модуль 4	5	6	5	3	19
Разом	13	14	21	17	65

Попередній студент      Наступний студент

Рис. 9. Відображення другої сторінки

ка досягнута саме за рахунок використання нового класу, тобто тих переваг, які дає об'єктно орієнтоване програмування.

## Література

1. *Бондаренко, М.А.* Основні напрямки розвитку в комп'ютерній галузі. / *М.А. Бондаренко.* — Харків : «ФОП Лібуркіна Л. М.», 2009. — 696 с.
2. *Бондаренко, М.А.* Основи інформаційних технологій та програмування. / *М.А. Бондаренко.* — Харків : «ФОП Павленко О.Г.», 2010. — 536 с.