Отже, задачі з параметром є цінним засобом розвитку здібностей учнів до здійснення математичної діяльності. Оволодіння досвідом такої діяльності формує в учнів здібності до самостійного осмислення та пошуку розв'язків задач.

Список використаних джерел

7. Бігун Я. Й. Математичне моделювання екологічних, економічних і соціальних процесів: Навч. посібник / Я.Й. Бігун. - Чернівці : Рута, 2005. — 80с.
8. Бондаренко Т. Практичні роботи на уроках математики / Т. Бондаренко. – [Електронний ресурс] – Режим доступу: http://klass.ho.ua/index.php?job=100029
9. Возняк Г.М. Прикладна спрямованість шкільного курсу математики: Розв'язування екстремальних задач: Метод, посібник / Г.М. Возняк, М.П. Маланюк. - К.: Рад. шк., 1984.
10. Формування життєвих вмінь та навичок учнів на уроках математики шляхом використання прикладних задач. – [Електронний ресурс] – Режим доступу: http://schoolv.ucoz.ru/publ/formuvannja_zhittevikh_vmin_ta_navichok_uchniv_na_uroka kh_matematiki_shljakhom_vikoristannja_prikladnikh_zadach/1-1-0-1.

FORMATION IN THE PRACTICES OF MATHEMATICAL MODELING ADMINISTRATION IN THE TRAINING PROCESS OF THE SOLUTION OF FULL PROBLEMS WITH PARAMETERS

Abstract. *The article deals with the methodology of teaching students to solve scene problems with parameters using mathematical modeling methods.*

Keywords: *plot problem, task with parameter, methods of mathematical modeling, algorithmic way of thinking.*

Анатолій Яровенко

ТЕКСТОВИЙ РЕДАКТОР З ПОЗИЦІЙ ОБ'ЄКТНОЇ ПАРАДИГМИ

Анотація. *В статті пропонується до розгляду авторська методика навчання розділу інформатики «Текстові редактори» на базі об'єктної парадигми. Об'єктно-зорієнтований підхід до вивчення текстових редакторів полягає в тому, що текстовий редактор розглядається з позицій об'єктної парадигми – як об'єктно-зорієнтована система, в середовищі якої користувач має справу (працює) з об'єктною моделлю текстового документа. Об'єктно-зорієнтований підхід дозволяє усунути недоліки, властиві процедурному підходу, сприяє розвитку абстрактного мислення, забезпечує вирішення проблеми актуалізації знань при вивченні об'єктно-зорієнтованого програмування.*

Ключові слова: *текстовий редактор, парадигма, документ, об'єкт, клас, ієрархія, властивість, операція.*

Формулювання проблеми.

Парадигма об'єктно-зорієнтованого програмування (ОЗП) сьогодні є без сумніву найпоширенішою та найпопулярнішою в розробці застосунків у різних сферах інформатики – в середовищах програмування, системах керування базами даних, імітаційного моделювання, інтелектуального аналізу даних тощо.

Оскільки компетенції з ОЗП є обов'язковими компонентами професійної компетентності сучасного ІТ-фахівця, то розробка підходів та методик, які забезпечать (чи сприятимуть) ефективне формування таких компетенцій, є актуальною педагогічною проблемою.

В рамках парадигми ОЗП від студента вимагається вміння проектування програм та програмних систем як сукупності класів та об'єктів, що, в свою чергу, вимагає нового (на відміну від алгоритмічного) мислення в категоріях класів та об'єктів, вміння застосовувати на практиці принципи моделювання, зокрема, декомпозиції (для побудови ієрархій класів та об'єктів) та абстрагування (для ідентифікації властивостей класів та об'єктів).

На переконання автора розгляд текстових редакторів з позицій об'єктно-зорієнтованого підходу не тільки сприятиме ефективному засвоєнню цього розділу інформатики, але й забезпечить актуалізацію знань при вивченні об'єктно-зорієнтованого програмування.

Метою статті є представлення авторської методики навчання розділу інформатики «Текстові редактори», яка базується на використанні об'єктної парадигми.

Аналіз останніх досліджень і публікацій.

Аналіз навчальної літератури з інформатики, виданої протягом останніх 10-ти років (проаналізовано більше 40 підручників), дозволяє констатувати, що при вивченні текстових редакторів застосовується традиційний процедурний підхід.

Основна увага при цьому приділяється вивченню процедурних технологій – як виконати ту чи іншу операцію (процедуру) над виділеними складовими текстового документу – фрагментами тексту, таблицями, списками тощо. Такий підхід запозичений із парадигми процедурного програмування, коли програма розроблялась як набір алгоритмічних процедур чи інструкцій комп'ютеру.

Особливих труднощів при цьому не виникає, оскільки операції над елементами текстового документу згруповані в пунктах системного меню текстового редактора, а також можуть відображатись у вигляді піктограм (з підказкою) на панелях інструментів. Тут і в подальшому для демонстрації використовувались текстові редактори Microsoft Word з пакету Microsoft Office (версія MS® Word 2016 MSO 64-біт) та LibreOffice Writer (версія 5.2.3.3 64-біт) з офісного пакету LibreOffice.

Але такий процедурний підхід не забезпечує необхідної ефективності освоєння текстових редакторів, не дозволяє застосовувати на практиці принципи моделювання, не дозволяє побудувати об'єктну модель текстового документа, не сприяє розвитку абстрактного мислення, не забезпечує навіть знайомства з основними елементами ОЗП (класи, об'єкти, властивості тощо), що було б дуже доречним і корисним в сенсі актуалізації знань при вивченні ОЗП.

Виклад основного матеріалу дослідження.

Для обґрунтування можливості й доцільності об'єктно-зорієнтованого підходу при вивченні текстових редакторів, розглянемо основні поняття парадигми ОЗП. Багато науковців у своїх працях намагалися і намагаються сформулювати основні принципи ОЗП. На думку автора найкраще (найлаконічніше і найточніше) це вдалося видатному американському вченому в галузі інформаційних технологій і програмування, автору класичних праць з об'єктно-орієнтованого аналізу Граді Бучу [1], Деборі Армстронг [2], яка дослідила комп'ютерну літературу з цієї теми, видану протягом останніх 40 років, та Алану Кею – розробнику мови Smalltalk, якого вважають одним з «батьків-засновників» ОЗП. Звичайно, серед усіх формулювань і тверджень вищезгаданих авторів ми використаємо лише ті, які необхідні для нашого дослідження.

Для того, щоб ефективно використовувати об'єктний підхід, необхідно насамперед відповісти на запитання [1, с. 104]:

- Що таке класи та об'єкти?
- Як правильно ідентифікувати класи та об'єкти, які відносяться до конкретного застосунку?

За означенням Граді Буча «ОЗП – це метод програмування, який базується на представленні програми як сукупності взаємодіючих об'єктів, кожен з яких є екземпляром певного класу, а класи є членами певної ієрархії успадкування» [1, с. 69]. Підкреслюючи, що об'єктом може бути матеріальна або абстрактна сутність, Граді Буч дає наступне «емпіричне означення об'єкта: Об'єкт володіє станом, поведінкою та індивідуальністю. Структура і поведінка схожих об'єктів визначається у спільному для них класі» [1, с. 109].

В контексті об’єктно-зорієнтованого аналізу Граді Буч визначає клас «як множину об’єктів, які володіють спільною структурою, поведінкою і семантикою. Окремий об’єкт є просто екземпляром класу» [1, с. 123].

Дебора Армстронг: «Клас визначає абстрактні характеристики деякої сутності, включаючи характеристики самої сутності та дії, які вона здатна виконувати (її поведінка, методи або можливості)» [2, с. 125].

Алан Кей: «Кожен об’єкт є представником (екземпляром) класу, який виражає загальні властивості об’єктів. У класі задається поведінка (функціональність) об’єкту. Таким чином всі об’єкти, які є екземплярами одного класу, можуть виконати одні й ті ж самі дії. Класи організовані у єдину деревовидну структуру з загальним корінням, яка називається ієрархією успадкування».

Принципово новим і важливим підходом методології ОЗП є включення в структуру об’єкта його поведінки – так званих методів (операцій обробки даних). Метод – це процедура або функція обробки даних, яка належить класу об’єктів.

Таким чином, щоб побудувати об’єктну модель, треба:

- ідентифікувати класи та об’єкти, за допомогою яких можна достатньо повно описати моделюючу систему;
- ідентифікувати їх властивості, істотні для даної задачі;
- ідентифікувати методи (поведінку, можливі дії або команди, які можуть виконати об’єкти).

Задача ідентифікації об’єктів і класів є складовою загальнонаукової проблеми класифікації. Ця задача настільки важлива і складна, що Граді Буч присвятив їй у своїй фундаментальній праці цілий розділ, в якому не тільки проаналізував труднощі та проблеми, але й розглянув підходи і методи її розв’язання [1, с. 150-174].

На щастя проблеми ідентифікації вирішені розробниками текстових редакторів, які не тільки ідентифікували об’єкти текстового документа, їх властивості та допустимі операції, але й розробили інструменти визначення та зміни властивостей об’єктів (засоби форматування) та механізми виконання операцій над ними з використанням новітніх технологій опрацювання даних (Point&Click, Drag&Drop, Clipboard, OLE тощо).

Завдяки цьому користувач має можливість самостійно побудувати об’єктну модель будь-якого текстового документа, яка дозволить програмам та скриптам динамічно отримувати доступ до вмісту, структури та стилю цього документа.

З того, що об’єктна парадигма використовує в якості базових елементів об’єкти, а не алгоритми, слідує що при розгляді текстових редакторів (текстового документа) з позицій об’єктно-зорієнтованого підходу (об’єктної парадигми) доцільно зосередитись на ієрархічній об’єктній моделі документа, тобто, вивчати класи об’єктів документа, але не операції (процедури) над фрагментами документа.

Відзначимо, що об’єктна парадигма в нашому підході до вивчення текстових редакторів використовується не формально. Ми також розглядатимемо класи об’єктів, які мають цілком визначені властивості, але, на відміну від парадигми ОЗП, для класу визначатимемо не методи (дії, які можуть виконувати об’єкти класу), а операції, які можна виконувати *над* (*виділено мною*) об’єктами класу.

Відзначимо, що об’єктна парадигма в нашому підході до вивчення текстових редакторів використовується не формально. Ми також розглядатимемо класи об’єктів, які мають цілком визначені властивості, але, на відміну від парадигми ОЗП, для класу визначатимемо не методи (дії, які можуть виконувати об’єкти класу), а операції, які можна виконувати *над* (*виділено мною*) об’єктами класу.

Сукупність властивостей класу будемо називати форматом, а під форматуванням розумітимемо задання чи зміну властивостей класу. На відміну від редагування (зміни

внутрішнього вмісту об’єкта), при форматуванні задаються чи змінюються «зовнішні» характеристики об’єкта, не змінюючи його «внутрішній» вміст.

Очевидно, що кореневим в ієрархії класів об’єктів текстового редактора є клас «Документ». Відзначимо надзвичайну потужність цього класу, що забезпечує користувача практично необмеженими можливостями і свободою дій. Тільки на основних 8-ми сторінках бібліотеки міститься більше тисячі (1057) шаблонів для створення нового документа.

Формат подання об’єктів класу «Документ» задається вибором відповідної опції на першій зліва панелі інструментів пункту «Подання» системного меню редактора MS Word чи відповідної опції з контекстного меню пункту «Перегляд» системного меню редактора LibreOffice Writer.

Основні ж властивості (формат) об’єктів класу «Документ» у редакторі MS Word задаються, змінюються та переглядаються на сторінці «Відомості» пункту системного меню «Файл» в розділі «Властивості».

В цьому ж розділі можна відкрити (лівий Click на заголовку «Властивості») діалогове вікно, в якому на 5-ти вкладках можна задати, змінити чи переглянути весь набір властивостей об’єкту класу «Документ».

Зауважимо, що це ж вікно можна відкрити послідовністю команд **Меню → Файл → Переглянути властивості документу**.

В редакторі LibreOffice Writer всі властивості об’єкту класу «Документ» згруповані на 7-ми вкладках діалогового вікна, яке відкривається послідовністю команд **Файл → Властивості…**

Використовуючи вкладки діалогового вікна «Властивості» користувач має можливості для задання або зміни широкого спектру властивостей конкретного об’єкту класу «Документ» та переглянути його статистичні характеристики.

Принципово невірним є твердження, що формат документа визначається форматом його об’єктів. Адже, як було зазначено вище, формат документа визначає його «зовнішні» властивості, тобто, властивості власне документа, але не його «внутрішніх» складових. Як буде показано в наступних розділах дослідження, властивості різних об’єктів навіть в межах одного класу можуть бути абсолютно різними.

Перелік допустимих операцій над документом відображається у вигляді відповідних команд (опцій) контекстного меню пункту «Файл» системного меню обох редакторів, що розглядаються. До знайомих і зрозумілих навіть інтуїтивно операцій «Відкрити», «Закрити», «Друк», «Зберегти» та «Зберегти як» (зберегти під новим іменем та/або з перетворенням типу) в сучасних версіях редакторів додані нові операції над документами.

За допомогою команд «Експорт» (MS Word) та «Експорт у PDF» (LibreOffice Writer) користувач може перетворити активний текстовий документ у документ формату PDF/XPS, параметри якого можна задати у відповідному діалоговому вікні.

Широкі можливості щодо публікації документа користувач отримує завдяки засобам розділу «Спільний доступ» контекстного меню пункту «Файл» системного меню редактора MS Word.

За допомогою цих засобів текстовий документ можна опублікувати (зберегти) у хмарі, надіслати за вказаними адресами електронної пошти у форматах Word, PDF чи XPS через MS Outlook 2016 або як факс через відповідну службу факсів Інтернету.

Слід також відзначити дві нові операції в дусі часу та сучасних Інтернет-технологій – онлайнове презентування та публікація в блозі, для виконання яких у MS Word передбачені відповідні засоби.

У редакторі LibreOffice Writer текстовий документ можна надіслати електронною поштою та перетворити у HTML-документ або складений документ з розширенням odf.

Наостанок зауважимо, що при аналізі операцій над об'єктом класу «Документ», останні інколи плутають з операціями над об'єктом файлової системи – файлом, який містить текстовий документ. Такої плутанини легко уникнути, якщо пам'ятати, що до множини допустимих операцій над об'єктом класу «Документ» належать (відносяться) тільки ті операції, які можна виконати над об'єктом в *середовищі* і *засобами* (підкреслено мною) тестового редактора. Наприклад, видалити текстовий документ можна тільки як файл, який містить цей документ.

Висновки та перспективи подальших розвідок.

Обґрунтовано доцільність розгляду текстового редактора з позицій об'єктної парадигми – як об'єктно-зорієнтованої системи, в середовищі якої користувач має справу (працює) з об'єктною моделлю текстового документа.

Завданням подальших досліджень є створення об'єктної моделі текстового документа, яка дозволить програмам та скриптам динамічно отримувати доступ до вмісту, структури та стилю документів, що забезпечить ефективне опрацювання останніх.

Список використаних джерел

1. Буч Г., Максимчук Р.А., Энгл М.У и другие. Объектно-ориентированный анализ и проектирование с примерами приложений, 3-е изд.: Пер. с англ. – М.: 000 «И.Д. Вильямс», 2008. – 720 с.

2. Armstrong D.J. The Quarks of Object-Oriented Development. – Communications of the ACM, 2006, 49 (2). – pp.123–128.

TEXT EDITOR WITH THE POSITIONS OF THE OBJECT PARADIGM

Abstract. *In the article proposes to consider the author's methodology of teaching the section of computer science «Text editors» on the basis of the object paradigm. Object-oriented approach to the study of text editors is that the text editor is considered from the standpoint of the object paradigm - as an object-oriented system, in the environment in which the user is dealing (working) with the object model of a text document. Object-oriented approach allows to eliminate the disadvantages inherent in the procedural approach, contributes to the development of abstract thinking, provides a solution to the problem of updating knowledge in the study of object-oriented programming.*

Keywords: *text editor, paradigm, document, object, class, hierarchy, property, operation.*