УДК 681.142.2

В. Амеліна, В. Семотюк
Національний університет “Львівська політехніка”,
кафедра програмного забезпечення

ІНСТРУМЕНТАЛЬНА СИСТЕМА ДИНАМІЧНОГО ГЕНЕРУВАННЯ МОБІЛЬНИХ НАВЧАЛЬНИХ ПРОГРАМ

Наведено результати дослідження принципів побудови та розроблення інструментальної системи, яка дає змогу створювати навчальні програми для мобільних пристроїв; проаналізовано розроблений метод динамічного генерування мобільних програм. Подано основні елементи архітектури спроектованої інструментальної системи.

The results of research of principles of construction and development of the instrumental system are result. It allows to create on-line tutorials for the mobile devices. Tthe analysis of the developed method of dynamic generation of the mobile programs is conducted. The basic elements of architecture of the projected instrumental system are given.

Вступ

У роботі запропоновано підходи до розв’язання проблем, що виникають під час проектування та реалізації інструментальної системи динамічного генерування навчальних програм для мобільних систем, досліджуються шляхи їх вирішення та наводяться результати проектування. UML-діаграми представляють основні елементи архітектури спроектованої інструментальної системи.

Програмування для мобільних пристроїв – важлива галузь інформаційних технологій. Для цього сегмента прогнозують швидкий ріст у міру того, наскільки мобільні програми будуть підтримані технічними можливостями мобільних пристроїв [1]. Останнє покоління мобільних пристроїв має кольорові дисплеї високої роздільної здатності, велику пам’ять тощо, що дає змогу зробити навчання за допомогою мобільних технологій привабливішим, знизити вартість розробки, наблизити його до навчальних систем на «традиційних» платформах. Використання мультимедійних засобів спростить запам’ятовування навчального матеріалу та сприятиме підвищенню якості знань. Однак навчальні програми для мобільних пристроїв можуть розглядатись лише як доповнення до повноцінного навчання із викладачем, отже, найбільшого ефекту можна досягти, узгодивши зміст програм із навчальним планом та методикою викладання конкретного викладача. Це завдання може бути вирішене за допомогою створення комп’ютерної системи, яка дасть змогу педагогам самостійно формувати навчальні програми та їх наповнення.

На ринку зараз існують дві ОС, адаптовані під мобільні телефони: Windows Mobile та Symbian. Операційна система Symbian є відкритою і висуває значно менше порівняно з Windows Mobile, вимог до апаратного забезпечення [2]. Крім операційних систем важливу роль відіграють базові платформи прикладного програмного забезпечення. Ці платформи працюють на великій кількості різних мобільних пристроїв та надають покращені набори інструментальних програмних засобів та базових можливостей. Прикладами таких платформ є WAP microbrowsers, Java та .NET Compact Framework [3].Велика кількість нових моделей, особливо телефони на основі S60 (або смартфони), мають великий обсяг внутрішньої пам'яті, яку можна додатково розширити за допомогою MMC-карт [4]. Всі найбільші виробники мобільних пристроїв, зокрема Nokia, Motorola, Siemens, Samsung, Fujitsu, Invetec, LG Electronics, Mitsubishi, Nec, Panasonic, Psion, RIM, Sharp та Sony прийняли Java як частину своєї головної стратегії для майбутніх інтелектуальних пристроїв [5].

Широкі відомості про застосування мобільних телефонів у навчанні містяться у [6], де опиано систему КИП-М. Це потужна навчальна система, яка містить спеціально виконані підручники на паперових носіях, мобільний телефон (із WAP та GPRS) для учнів та комп’ютер і Інтернет – для

«віртуального тренера». Учень працює лише з двома елементами – книгою, у якій викладено весь навчальний, довідковий матеріал та вправи, а також мобільним телефоном, за яким він отримує номери вправ і завдань та відповідає на них, обираючи та відсилаючи варіанти своїх відповідей.

Запропонована система має на меті забезпечити можливість динамічного вибору схеми навчання та формування навчального матеріалу комп'ютерної навчальної системи для мобільних систем, зокрема, мобільних телефонів. Це дасть змогу створювати навчальні програми того рівня і тієї структури, яка найбільше відповідає потребам індивідуального користувача. За допомогою динамічного формування коду можна легко та швидко сформувати із запропонованих модулів програму, яка допомагає у вивченні, наприклад, іноземної мови. При цьому користувач не повинен обов'язково володіти навичками програмування – ним може бути, наприклад, викладач іноземної мови.

1. Вибір архітектури системи

Розглядалися два можливі варіанти архітектури проектованої інструментальної системи:

– створення незмінної мобільної частини і передавання на мобільний пристрій навчально-тестувального наповнення, створеного за допомогою комп'ютерної частини;

– динамічне генерування мобільної частини комп'ютерною на основі обраних користувачем модулів та наповнення.

У першому випадку доцільним було б використання мови C++ для розробки, застосування якої з врахуванням API залежить від операційної системи, а це, своєю чергою, знижує універсальність розробленої системи і можливості застосування її тільки для певних модифікацій мобільних телефонів. У зв'язку з цим було вибрано інший варіант архітектури, при якому для проектування доцільним є середовище Java 2ME, що забезпечує кросплатформеність системи. Використовуючи Java 2ME, можна створити архітектуру з динамічним генеруванням комп'ютером мобільної частини на основі обраних користувачем модулів, тобто структури та наповнення. Цим забезпечуються широкі можливості конфігурування вихідних навчальних програм; можливість мінімізації обсягу пам'яті для вихідних навчальних програм; універсальність системи щодо вибору наповнення. На рис. 1 представлено архітектурне вирішення системи. Такий підхід дає змогу, крім можливості динамічно генерувати код вихідних програм за допомогою комп'ютерної системи, передбачити можливість зміни не лише наповнення вихідних програм, але й набору модулів, які потрібно ввести до системи. Також необхідно передбачити можливість повторного використання як параметрів, заданих щодо конкретного навчального проекту (вихідної програми), так і набору наповнення.

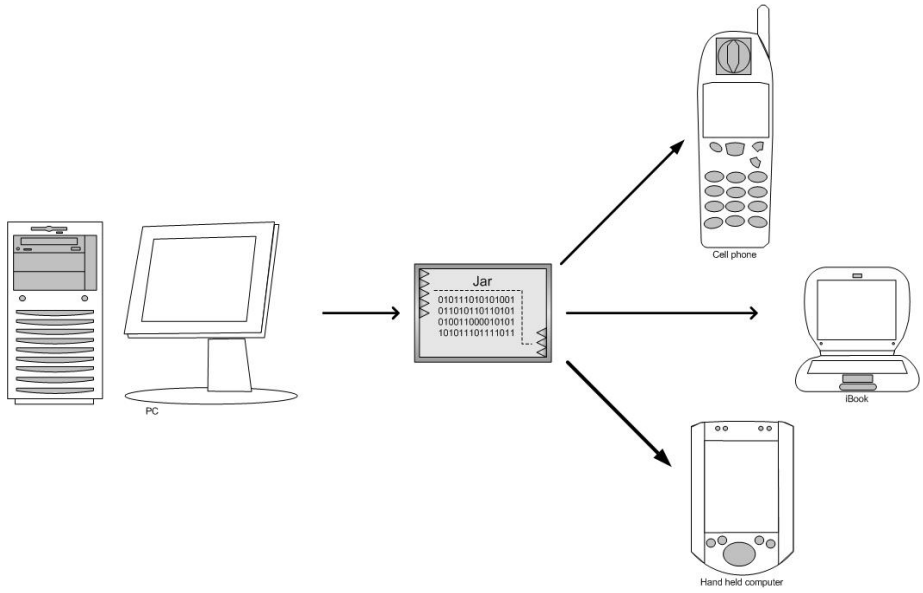

Рис. 1. Архітектура системи

2. Опис методу та проектування системи

Суть методу полягає у динамічному генеруванні коду на основі попередньо заданих шаблонів та контенту (наповнення), отриманого від користувача. Розглянемо типовий модуль для мобільної системи вивчення іноземної мови. Модуль реалізує одну з методик вивчення іноземної мови за допомогою малюнків. Учень має вказати відповідний малюнок для заданого слова. Було розроблено мідлет (різновид Java-аплікацій, призначений для роботи на різних мобільних пристроях), який реалізує цей модуль. Для динамічної заміни контенту у вихідному коді мідлета розроблена система тегів, якими позначаються місця, де знаходиться інформація, що стосується контенту, або ж індивідуалізує модуль (наприклад, назву). Отриманій шаблон вставляється у відповідний модуль desktop-частини, реалізованої на основі технології .Net. У цій частині користувач – формувач мобільної навчальної програми – має можливість сформувати контент (у даному випадку це набір слів та відповідних малюнків).

Для генерації мобільної навчальної програми вставлений шаблон аналізується, і всі теги заміняються на інформацію, що відповідає параметрам та контенту, який ввів чи обрав користувач. Після цього отримані вихідні тексти (програмний код мовою Java) мобільної навчальної програми зберігаються на жорсткому диску (за замовчуванням або у директорію, обрану користувачем за допомогою функцій меню) та запускається процес компіляції. У результаті на виході отримуємо навчальну програму, готову для встановлення та використання на мобільному телефоні.

На рис. 2 зображено компонентну UML-діаграму інструментальної системи. Комп'ютерна частина побудована з використанням механізму плагінів (plug-in). Навчальні модулі, які можуть бути введені користувачем до вихідної навчальної програми для мобільних пристроїв, реалізовані у вигляді окремих бібліотек. Ці бібліотеки динамічно підключаються до комп'ютерної частини за допомогою розробленого класу PluginManager. Цим досягається:

- гнучкість;
- можливість розширення функціональності системи;
- можливість конфігурування.

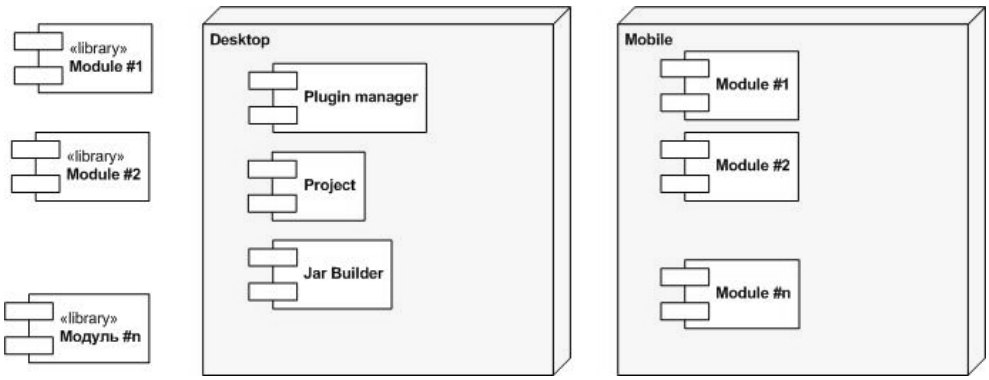

Рис. 2. Компонентна діаграма проектованої системи

На UML-діаграмі (рис. 3) зображено операції, які можуть здійснюватися користувачем:

- визначення переметрів навчальної системи:

– вибір модулів, які будуть введені до навчальної програми;

– вибір наповнення для обраних модулів.

- запис обраних параметрів у файл. Цим досягається можливість повторного використання як усього проекту загалом, так і певної його частини;
- зчитування параметрів проекту з файла.

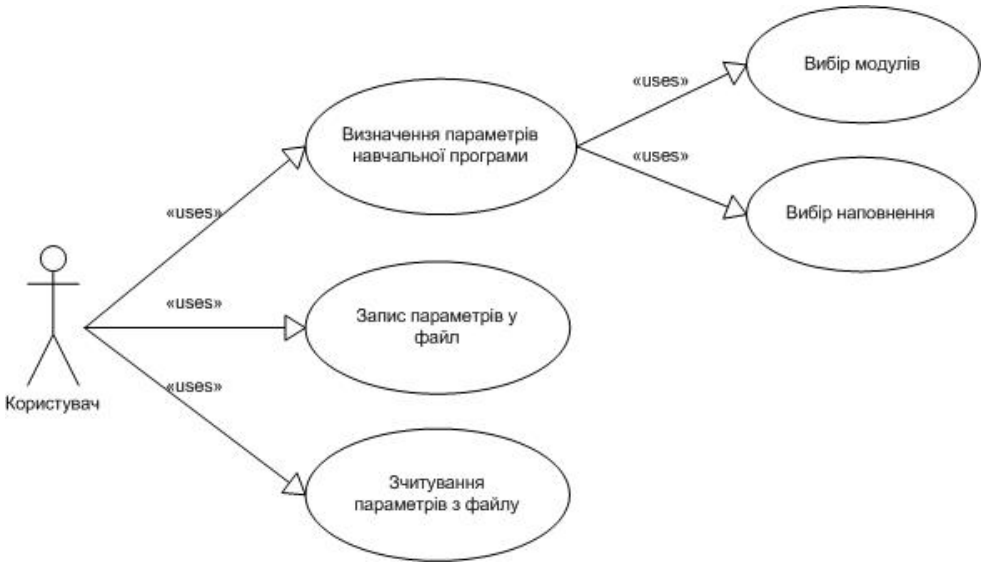

Рис. 3. Use-case діаграма дій користувача

Наповнення можна вибрати двома шляхами – користувач має можливість:

- набрати на клавіатурі комп'ютера слова та/або речення іноземної мови, граматичні правила тощо, обрати зображення та/або звукові файли (залежно від того, який модуль наповнюється);
- обрати наперед визначений набір даних (який до того був збережений самим користувачем). Такі файли мають розширення .set.

На UML-діаграмі (рис. 4) зображено етапи генерування мобільної частини:

- система аналізує шаблони обраних користувачем модулів та заміняє знайдені теги на відповідне наповнення за параметрами, отриманими від користувача;
- записує сформований програмний код для мобільної частини на диск;
- виконує компіляцію мобільної частини.

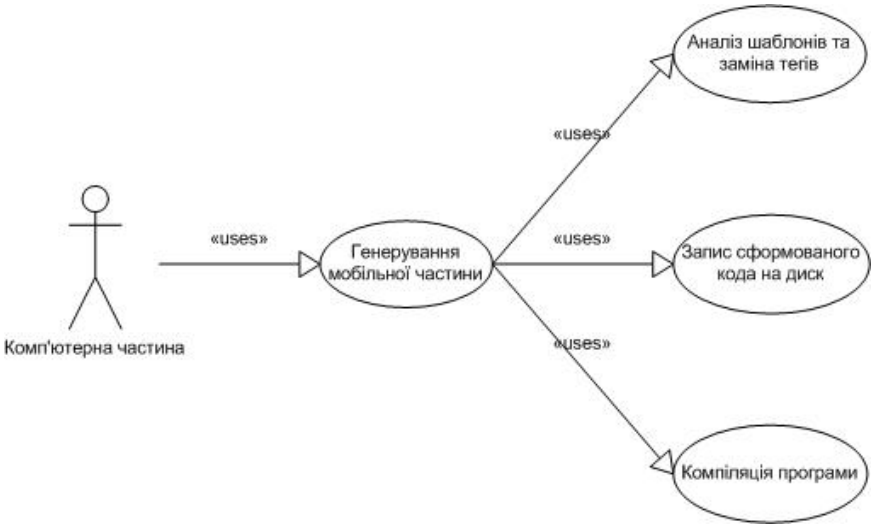

Рис. 4. Use-case діаграма генерування мобільної частини

На рис. 5 наведена UML діаграмі класів, на якій зображено класи, що репрезентують динамічну генерацію та компіляцію коду та внутрішнє представлення навчальної системи, а саме назву, список модулів, з котрих вона складається, наповнення для кожного модуля тощо.

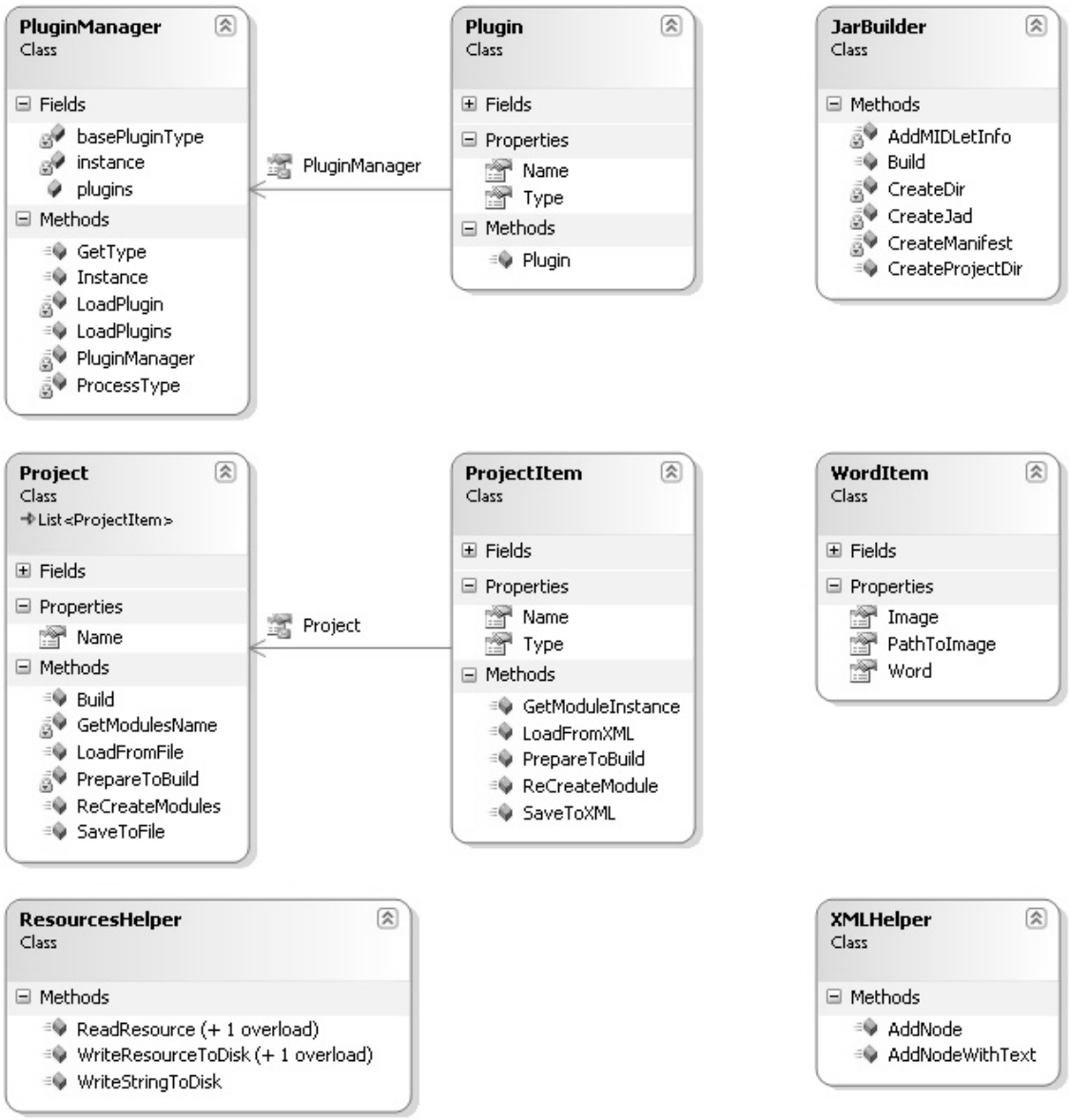

Рис. 5. Клас-діаграма внутрішніх та допоміжних класів

- JarBuilder – це клас, що відповідає за формування вихідних кодів навчальної системи на диску та їхню компіляцію;
- Project – зберігає інформацію про навчальну програму та інкапсулює всі пов’язані з нею операції, містить список обраних користувачем модулів тощо;
- ProjetItem – представляє окремий модуль навчальної програми та містить інформацію про назву, тип та сам модуль;
- PluginManager – клас для роботи з плагінами;
- Plugin – клас для збереження інформації про плагін;
- ResourceHelper – допоміжний клас для роботи з ресурсами;
- XMLHelper – допоміжний клас для роботи з XML.

Структуру модулів та форм подано у вигляді UML діаграми (рис. 6).

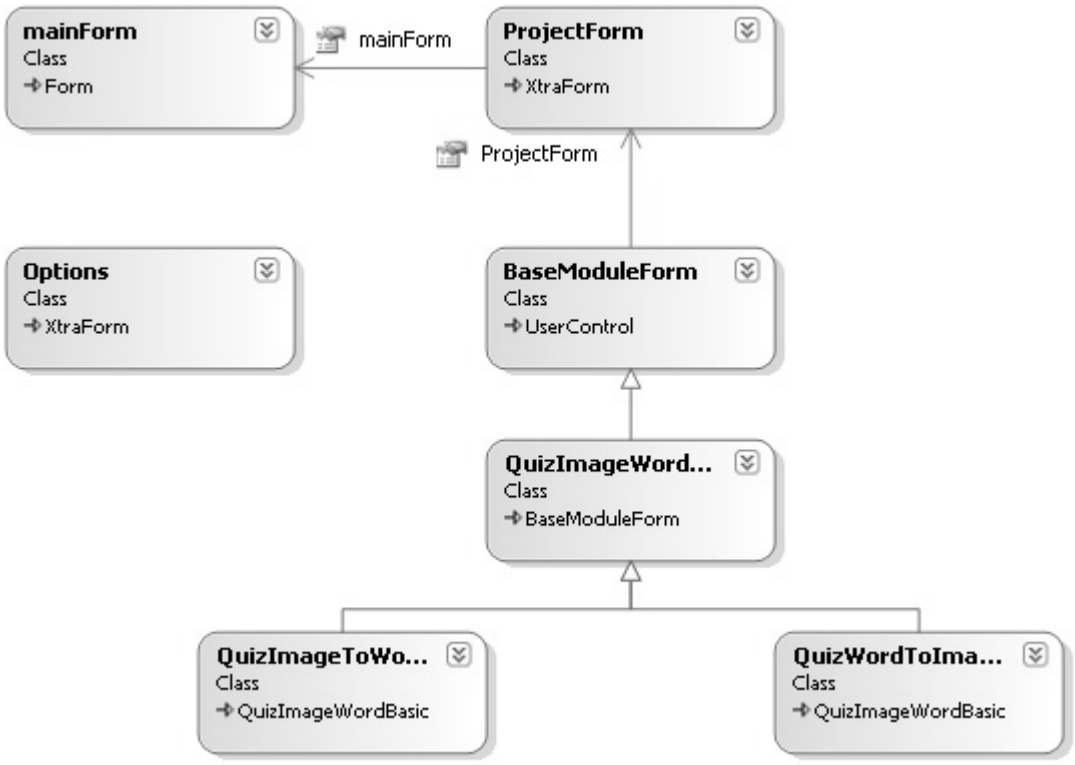

Рис. 6. Клас-діаграма модулів

3. Формат зберігання готових програм та інтерфейс користувача

У результаті проектування було виявлено необхідність розробити власний формат зберігання отриманих від користувача даних. Необхідно вирішити такі завдання:

- зберігання всього проекту для повторного використання;
- зберігання наборів данних (наповнення) для окремих модулів;
- можливість використання одного набору даних для схожих за функціональністю модулів.

За основу обрано XML – універсальну гнучку мову, яка дає змогу розробляти власні формати. Головними тегами розробленого формату є:

- <Project> – всередині цих тегів міститься сам проект;
- <Name> – використовується для зберігання назви проекту та модулів;
- <Modules> – всередині цих тегів міститься список модулів;
- <Module> – тег для зберігання модуля;
- <Set> – тег для зберігання набору даних (наповнення).

Обраний формат дає змогу зберігати всі дані про створений за допомогою інструментальної системи проект (навчальну програму). Завдяки цьому проект може бути повторно використаний для генерації програми після внесення деяких змін користувачем.

Розроблена програмна система побудована з використанням MDI Child інтерфейсу і складається з головної форми (mainForm) та MDI Child форм (ProjectForm), кожна з яких є окремою навчальною програмою. ProjectForm форма містить TabControl. На першій сторінці користувач формує список модулів, з яких має складатись вихідна навчальна програма. На наступних сторінках розміщено комп'ютерні частини вибраних модулів (класи, породжені від BaseModuleForm), на яких користувач задає наповнення для модуля – вводить з клавіатури або обирає готовий набір даних. Вигляд головної форми з декількома MDI Child формами наведено на рис. 7.

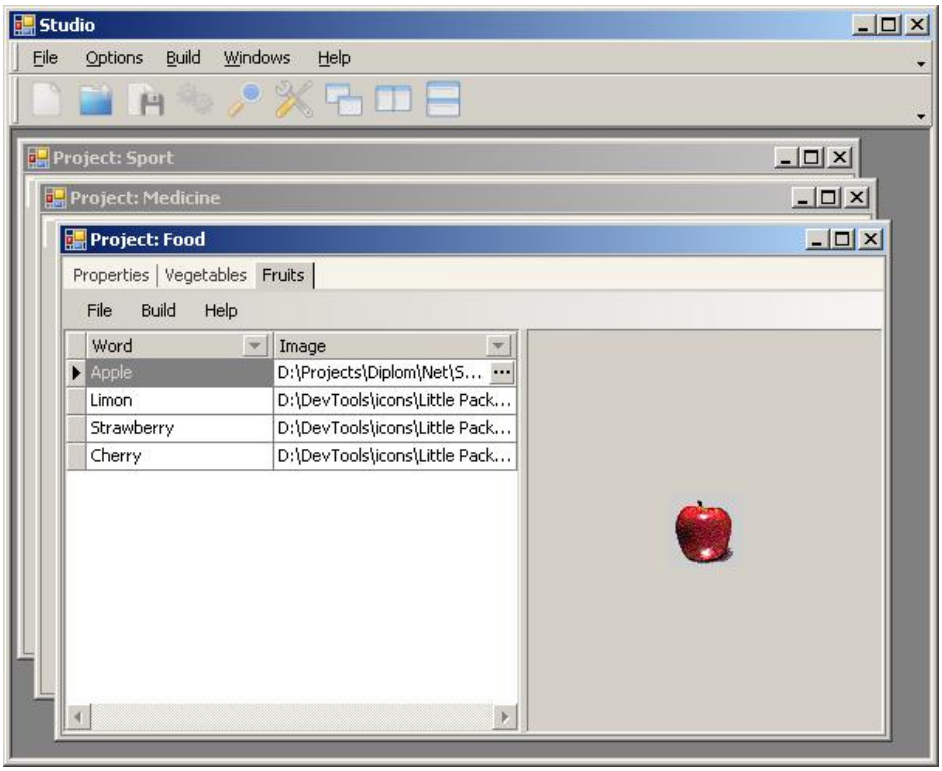

Рис. 7. Форма для введення контента

На рис.8 наведено інтерфейс мобільної частини з відкритим модулем для вивчення слів за малюнком.

Рис. 8. Вигляд інтерфейсу
навчальнго модуля вихідної програми

4. Програмна реалізація

На UML діаграмі (рис. 9) зображено класи вихідної мобільної частини. Програма складається з головного мідлета, представленого класом MMain. Для кожного навчального модуля існує окремий клас, породжений від Canvas. Ці класи інкапсулюють усю логіку, необхідну для роботи модуля.

Рис. 9. Клас-діаграма вихідної мобільної частини

Клас ModulesHolder призначений для зберігання інформації про модулі, з яких будується вихідна навчальна програма. Ця інформація використовується для побудови меню вибору модуля. Також цей клас виконує функцію “фабрики” – він створює екземпляри модулів. Клас ModulesHolder побудований за шаблоном проектування Singleton, оскільки нам завжди потрібен один і той самий екземпляр цього класу. Для відображення меню вибору модуля використовується клас CanvasMenu, породжений від класу Canvas.

Для побудови інтерфейсу користувача використовується допоміжний клас GUIHelper. Він же забезпечує уніфікацію інтерфейсу для різних модулів, оскільки вся робота ведеться тільки через нього. За потреби можна легко змінити інтерфейс користувача, змінюючи тільки цей клас.

Мобільна частина реалізована на мові **Java** і по суті являє собою мідлет (різновид Java-аплікацій, призначений для роботи на різних мобільних пристроях), а отже, призначена для виконання у середовищі віртуальної машини **J2ME**.

Мідлет для своего запуску потребує пристрій, який підтримує **J2ME** та **MIDP.** Як і будь-яка **Java**-программа, один раз скомпільований мідлет може бути запущений на будь-якому пристрої із підтримкою Java. Найпоширеніші засоби розроблення мідлетів — **Sun Java Wireless Toolkit** та **NetBeans** разом **NetBeans Mobility Pack**. Було використано **Sun Java Wireless Toolkit**. У дистрибутиві мидлета, крім файла **.jar**, міститься також файл **.jad**, в якому описується вміст файла .jar-а.

Мідлет має задовольняти такі вимоги для того, щоб його можна було запускати на мобільному пристрої:

- головний клас повинен бути потомком класу javax.microedition.midlet.MIDlet;
- мідлет повинен бути упакований у **.jar**-файл (наприклад, за допомогою jar-tool).

Вихідні навчальні програми задовольняють ці дві вимоги.

У мобільній частині реалізовано шаблон проектування Singleton, який на рівні JVM гарантує, що не буде створено більше одного екземпляру заданого класу, дозволяючи іншим класам отримати доступ до цього екземпляра. Для того, щоб забезпечити контроль над створенням екземплярів класу, достатньо встановити конструктор класу private.

Для дослідження ефективності методу генерування навчальних програм та обраних підходів до архітектури системи реалізовано чотири модулі:

- QuizImageToWord. Модуль працює в трьох режимах «Навчання», «Тренування» та «Тестування». У режимі «Навчання» учень переглядає слова іноземною мовою разом із їх графічним зображенням. Після цього він може обрати наступний режим – «Тренування» або «Тестування». У режимі «Тренування» учень повинен згідно з зображенням на дисплеї мобільного пристрою ввести відповідне слово у поле для введення. У режимі «Тренування» модуль працює, доки користувач не вийде з нього за власним бажанням. Під час роботи в режимі «Тестування» підраховується кількість правильних відповідей; робота модуля припиняється, коли буде отримано відповіді на всі питання, і учень зможе побачити кількість правильних відповідей.
- QuizWordToImage. Також працює у трьох режимах. Режим «Тренування» та «Тестування» побудовані за іншим принципом. Учню не потрібно вводити слово з клавіатури – йому показують слово, а він має обрати графічне зображення, що відповідає цьому слову, і натиснути клавішу Select.
- QuizSentence. Гра, яка полягає у складанні речень із запропонованих слів.
- Grammatic. Це текстовий модуль, який містить граматичні правила і взагалі будь-яку текстову інформацію, яку захоче вкласти в неї користувач інструментальної системи.

Висновки

Запропонована архітектура та розроблене програмне забезпечення інструментальної системи динамічного генерування навчальних систем для мобільних пристроїв. Навчальні програми, згенеровані інструментальною системою, працюють на усіх моделях мобільних пристроїв (телефонах, смартфонах, комунікаторах, КПК) з підтримкою Java MIDP 2.0 (конфігурація CLDC 1.0).

1. Семенець С.В. Розробка методів та програмного інструментарію створення прикладних програм для мобільних телефонів: Автореф. дис. ... канд. техн. наук: 05.13.06/ Інс-т проблем мат. машин і систем НАН України. – К., 2006. – 22с. 2 Горнаков С.Г. Symbian OS. Программирование мобильных телефонов на C++ и Java 2 ME. – М.: ДМК Пресс, 2005. – 448с.: ил. 3. Семенец С.В., Гавсиевич И.Б. Обзор программных архитектур ОС мобильных платформ Математичні машини і системи. – 2002. – №1. 4. Горнаков С.Г. Программирование мобильных телефонов на Java 2 Micro Edition. – М.: ДМК Пресс, 2006. – 336с.: ил. 5. Yuan M.Y. Enterprise J2ME: Developing Mobile Java Applications – Lafflin, Pennsylvania: "Prentice Hall PTR", 2006. 6. Система обучения КИП-М / Р. Гареев, М. Зиновкина // Высшее образование в России. – 2005. – №6. – С. 151–153.