

Список використаних джерел:

1. Карл Крейчі-Граф, «Основні питання нафтогазової геології». – Ленінград, Головна редакція паливно-гірничої літератури, 1934р. – С. 263.
2. Kosko B.: Neural Networks and Fuzzy Systems, A Dynamical Systems Approach to Machine Intelligence, Prentice Hall, Englewood Cliffs, NJ 07632, 1992.
3. Mohaghegh S., Arefi, R., Ameri, S., and Rose, D.: «Design and Development of an Artificial Neural Network for Estimation of Formation Permeability,» SPE 28237, Proceedings, 1994 SPE Computer Conference, July 31 – August 3, Dallas, Texas.
4. Mohaghegh S., Arefi R., Ameri S., and Hefner M. H.: «A Methodological Approach for Reservoir Heterogeneity Characterization Using Artificial Neural Networks», SPE 28394, Proceedings, 1994 SPE Annual Technical Conference and Exhibition, 25-28 September, New Orleans, Louisiana.

Депутат А.Б.

студент;

Вовк Р.Б.

кандидат технічних наук, доцент,

Івано-Франківський національний технічний університет нафти і газу

**РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
АВТОМАТИЧНОЇ ГЕНЕРАЦІЇ ДОКУМЕНТІВ**

В даний час робота з документами забирає левову частку часу багатьох фахівців офісних спеціальностей. Особливо це стосується співробітників тих організацій, яким доводиться займатися великими обсягами різної документації: страхових компаній, банків, нотаріальних бюро тощо. Комп'ютери, в даному випадку, надають неоціненну користь істотно полегшуючи роботу. Одним з перспективних рішень до автоматизації процедури створення однотипних документів є розробка програмного забезпечення, яке дозволить користувачам створити необхідні документи з мінімальними затратами часу та зусиль.

Архітектура програмного забезпечення (Software architecture) – сукупність найважливіших рішень щодо організації програмної системи. Архітектура включає в себе вибір структурних елементів та їх інтерфейсів, за допомогою яких складена система, а також їх поведінку в рамках співпраці структурних елементів; з'єднання обраних елементів структури і поведінки у все більш крупні системи; архітектурний стиль, який спрямовує всю організацію елементів, їх інтерфейси, взаємодію та з'єднання. У програмній інженерії Unified Modeling Language (UML) є типом діаграми статичної структури, яка описує структуру системи, показуючи системи класів, їх атрибути, операції (або методи), і відносини між об'єктами. Діаграма класів є основним блоком об'єктно-орієнтованого моделювання. Вона використовується для загального концептуального моделювання систематичного застосування і для детального моделювання перекладу моделей в програмний код. Як правило, діаграми

класів використовуються для охоплення вимог системи, тобто для того, що саме система повинна робити [1].

При проектуванні системи кількість класів ідентифікується і групується в діаграмі класів (рис. 1), що допомагає визначити статичні відносини між ними. При детальному моделюванні, класи концептуального дизайну часто поділяються на декілька підкласів. Для подальшого опису поведінки систем ці діаграми класів можуть бути доповнені діаграмою станів або автоматом станів UML.

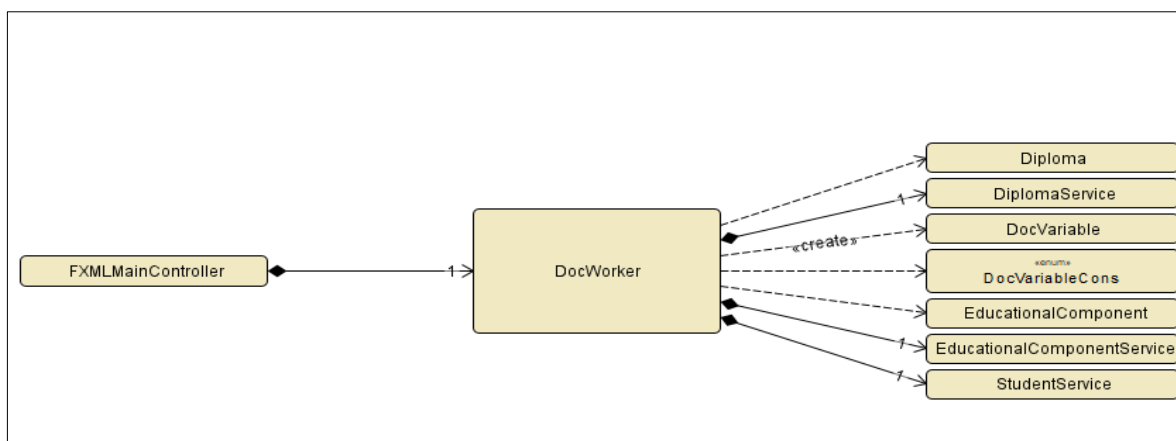
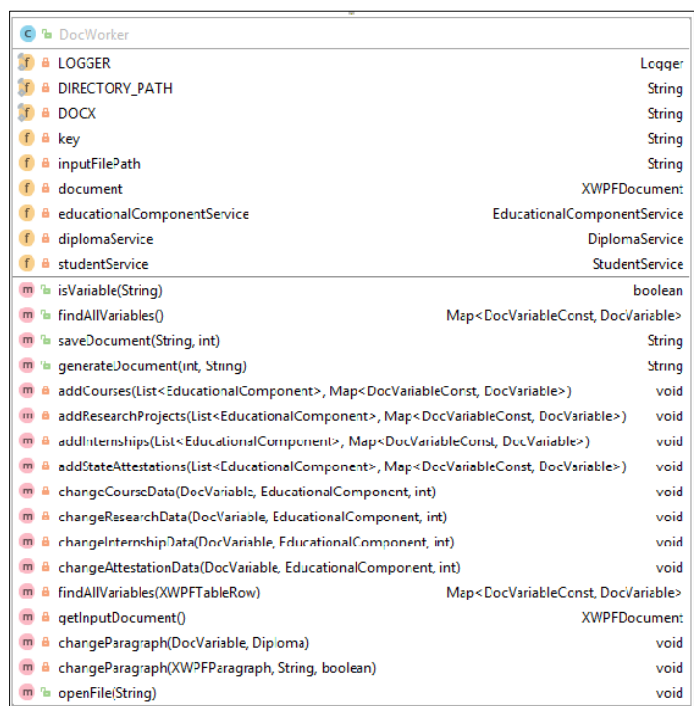


Рис. 1. Діаграма класів для генерації документів

Для централізації доступу до даних і функцій використовується Service Layer. Service Layer визначає межу між додатком і шаром сервісів, що утворює набір доступних операцій і управляє відповіддю додатку в кожній операції.

Основний функціонал виконує клас DocWorker (рис. 2), а саме це пошук змінних, запис даних і створення документів. Перед створенням документів, в об'єкт класу, бібліотеки Apache POI, XWPFDocument записується вхідний шаблон (файл формату docx) [2]. Спочатку виконується метод generateDocument() в параметри якого записується id користувача та назва файлу, який потрібно створити. Після цього викликається метод findAllVariables(), який шукає всі змінні в тексті та таблицях, і записує в колекцію, згодом витягує необхідні дані за допомогою об'єктів сервісів studentService, diplomaService і educationalComponentService. Після цього замінює змінні на необхідну інформацію і створює рядки в таблиці для потрібних компонентів і записує в них дані.



Field/Method	Type
LOGGER	Logger
DIRECTORY_PATH	String
DOCX	String
key	String
inputFilePath	String
document	XWPFDocument
educationalComponentService	EducationalComponentService
diplomaService	DiplomaService
studentService	StudentService
isVariable(String)	boolean
findAllVariables()	Map<DocVariableConst, DocVariable>
saveDocument(String, int)	String
generateDocument(int, String)	String
addCourses(List<EducationalComponent>, Map<DocVariableConst, DocVariable>)	void
addResearchProjects(List<EducationalComponent>, Map<DocVariableConst, DocVariable>)	void
addInternships(List<EducationalComponent>, Map<DocVariableConst, DocVariable>)	void
addStateAttestations(List<EducationalComponent>, Map<DocVariableConst, DocVariable>)	void
changeCourseData(DocVariable, EducationalComponent, int)	void
changeResearchData(DocVariable, EducationalComponent, int)	void
changeInternshipData(DocVariable, EducationalComponent, int)	void
changeAttestationData(DocVariable, EducationalComponent, int)	void
findAllVariables(XWPFTableRow)	Map<DocVariableConst, DocVariable>
getInputDocument()	XWPFDocument
changeParagraph(DocVariable, Diploma)	void
changeParagraph(XWPFParagraph, String, boolean)	void
openFile(String)	void

Рис. 2. Структура класу DocWorker

Отже, в цьому дослідженні було розглянуто архітектуру програмного продукту автоматичної генерації документів, наведено діаграму класів генерації документів, описано базовий функціонал.

Список використаних джерел:

1. Журнал «Інформаційні технології», UML: історія, специфікація [Електронний ресурс]. – Режим доступу: <http://it.ridne.net/node/265>.
2. Розширення Word у форматі файлів Office Open XML (.docx) [Електронний ресурс]. – Режим доступу: <https://msdn.microsoft.com/>.

Кастранець Ю.М.

студентка,

Національний технічний університет України

«Київський політехнічний інститут

імені Ігоря Сікорського»

СУТЬ ТА ПРИНЦИП РОБОТИ МАСИВНИХ СИСТЕМ МІМО

Багатоантенні системи зв'язку, відомі як multiple-input multiple-output (MIMO), інтенсивно досліджуються протягом останнього десятиліття. Перші роботи з постановкою задачі з'явилися вже в 1996 році [1; 2]. Згодом даному напрямку було присвячено низку наукових праць [3-5]. Завдяки істотному виграшу у швидкості передачі інформації технологія MIMO розглядається як