# АЛГОРИТМ ПЕРЕНЕСЕННЯ ДАНИХ З ФАЙЛІВ *DOC*-ФОРМАТУ ТАБЛИЧНОЇ СТРУКТУРИ ДО БАЗИ ДАНИХ

Дунько Ю.С.

Науковий керівник – доц. каф. “Системного програмного забезпечення”,

канд. техн. наук Блажко О.А.

На даний момент в організаціях існує велика кількість документів, які зберігаються в різних електронних форматах, а також існують інформаційні системи, які зберігають дані у формі реляційної бази даних (БД). Тому часто виникає необхідність внесення змістової частини документу до БД та навпаки — генерувати документ зі змістом з БД .

Метою роботи є автоматизація процесу перенесення змісту таблиць документу *DOC*-формату до БД, яка включає: опис документу з використанням мови розмітки *XML*; отримання даних з документу, їх розподіл на лексеми*;* виділення змістової інформації; формування таблиці, придатної для занесення до БД; занесення інформації до БД.

Пропонується в шаблоні документу описувати наступні його елементи : опис розділів, опис підрозділів на різних рівнях ієрархії, шапка таблиці, структура таблиці. Прикладом шаблону документу може бути: *<doc> ”назва” <head0> “Зарахувати студентів, що вивчаються за $форма_навчання$, до ОНПУ” </head0> <info> <head1> Спеціальність $спеціальність$ </head1> <tab_info> $ПІБ$ </tab_info> </info> <conclusion> Проект наказу внесено ... </conclusion> </doc>*, де *<doc>* - тег початку опису документу з його назвою, *<head0>* - тег опису заголовка документу, *<info>* - тег змістової частини документу, *<head1>* - тег опису заголовка параграфу, *<tab_info>* - тег опису заголовку таблиці, *<conclusion>* - тег опису кінцевої частини документу. Зміст елементів тегів може включати константи у вигляді словосполучень, які повинні однозначно ідентифікувати тег, та змінні з умовними позначками *$*, які в подальшому стають атрибутами таблиці БД.

Враховуючі структуру шаблону, запропоновано алгоритм розбору змісту документу, який містить наступні кроки. Крок 1. Пошук в документах строк-констант, які містяться в тегах *<head0>* та *<head1>* між змінними з позначками *$*. Крок 2. Отримання значень змінних в тегах та зберігання їх в масиві. Крок 3. Пошук в документах таблиць, які знаходяться після строк, описаних в тегах *<head1>*. Крок 4. Формування масивів зі змістом таблиць у відповідності з описом тегів *<tab_info>*. Крок 5. Формування масивів назв атрибутів таблиць на основі аналізу опису тегів *<tab_info>* та змінних з тегів *<head0>, <head1>*. Крок 6. Створення запитів по внесенню структури та змісту таблиць до БД.

При створенні програмного забезпечення була використана бібліотека *org.apache.poi* для аналізу документу *DOC*-формату та СУБД *PostgreSQL* для збереження таблиці в БД.