

УДК 004.912

Коломоєць Г.П., к. ф.-м. н.

Запорізька державна інженерна академія

## ПРОГРАМА ПІДГОТОВКИ ДО ДРУКУ ДИПЛОМІВ ПРО ВИЩУ ОСВІТУ ЗА ДАНИМИ ЄДЕБО ІЗ ВИКОРИСТАННЯМ СУЧАСНИХ ТЕХНОЛОГІЙ JAVA

**Коломоєць Г.П. Програма підготовки до друку дипломів про вищу освіту за даними ЄДЕБО із використанням сучасних технологій Java.** Вирішено актуальну задачу автоматизації підготовки до друку дипломів про вищу освіту за даними здобувачів вищої освіти, експортованими з Єдиної Державної Електронної Бази з питань Освіти (ЄДЕБО). Мета роботи: виконати аналіз структури та форматів даних, які можуть бути експортовані модулем "Замовлення документів" ЄДЕБО, та знайти ефективні засоби платформи Java для розробки програмного додатку для генерації PDF-документів, що представляють собою дипломи про вищу освіту груп здобувачів вищої освіти. За даними групи здобувачів вищої освіти, експортованими з ЄДЕБО в форматі eXtensible Markup Language (XML), за допомогою засобів Java-бібліотеки iText 7 виконується генерація документу у форматі PDF, що містить сторінки дипломів з даними здобувачів із фоновими зображеннями та інформацією, які залежать від рівня вищої освіти та інших параметрів навчання здобувача. Використовувались технологія Java Architecture for XML Binding для відображення експортованих даних у відповідні об'єкти користувачьких класів, високорівневі та низькорівневі засоби бібліотеки iText версії 7.1.0 для побудови PDF-документу, засоби інтернаціоналізації Java для відображення інформації на українській та англійській мовах. Аналіз структури XML-файлу з даними здобувачів вищої освіти, експортованими з ЄДЕБО, виявив неточності, зокрема, пов'язані з неввірно вибраними назвами для тегів, що зберігають дати закінчення навчального закладу та дати видачі документа. В роботі запропонований спосіб додавання фонового рисунку до сторінок багатосторінкового PDF-документа без відповідного збільшення розміру файлу. Апробація влітку 2017 р при випуску здобувачів Запорізької державної інженерної академії рівнів вищої освіти "бакалавр" та "спеціаліст" та взимку 2018 р - при випуску здобувачів рівня вищої освіти "магістр" довела ефективність розробленого програмного забезпечення.

**Ключові слова:** ЄДЕБО, диплом про вищу освіту, XML, JAVA, PDF, iText.

**Коломоєць Г.П. Программа подготовки к печати дипломов о высшем образовании по данным ЕГЭБО с использованием современных технологий Java.** Решена актуальная задача автоматизации подготовки к печати дипломов о высшем образовании по данным соискателей высшего образования, экспортируемым из Единой Государственной Электронной Базы по вопросам Образования (ЕГЭБО). Цель работы: выполнить анализ структуры и форматов данных, которые могут быть экспортированы модулем "Заказ документов" ЕГЭБО, и найти эффективные средства платформы Java для разработки программного приложения для генерации PDF-документов, представляющих собой дипломы о высшем образовании групп соискателей высшего образования. По данным группы соискателей высшего образования, экспортируемым из ЕГЭБО в формате eXtensible Markup Language (XML), с помощью средств Java-библиотеки iText 7 выполняется генерация документа в формате PDF, содержащего страницы дипломов данным соискателей с фоновыми изображениями и информацией, которые зависят от уровня высшего образования и других параметров обучения соискателя. Использовались технология Java Architecture for XML Binding для отображения экспортируемых данных в соответствующие объекты пользовательских классов, высокоуровневые и низкоуровневые средства библиотеки iText версии 7.1.0 для построения PDF-документа, средства интернационализации Java для отображения информации на украинском и английском языках. Анализ структуры XML-файла с данными соискателей высшего образования, экспортируемыми из ЕГЭБО, обнаружил неточности, в частности, связанные с неверно выбранными названиями для тегов, сохраняющих даты окончания учебного заведения и даты выдачи документа. В работе предложен способ добавления фонового рисунка к страницам многостраничного PDF-документа без соответствующего увеличения размера файла. Апробация летом 2017 г при выпуске соискателей Запорожской государственной инженерной академии уровня высшего образования "бакалавр" и "специалист" и зимой 2018 г - при выпуске соискателей уровня высшего образования "магистр" доказала эффективность разработанного программного обеспечения.

**Ключевые слова:** ЕГЭБО, диплом о высшем образовании, XML, JAVA, PDF, iText.

**Kolomoiets G.P. Software of preparation for the printing of diplomas about higher education by data of the USEDEI with the use of modern Java technologies.** The actual task of automating of preparation for the printing of diplomas about higher education according to the data of higher education seekers exported from the Unified State Electronic Database on Education Issues (USEDEI) was solved. The purpose of the work is to analyze the structure and data formats that can be exported by the "Ordering Documents" module of USEDEI and to develop software for generating PDF documents, which are diplomas of higher education of groups of higher education seekers. By data exported from the USEDEI in the eXtensible Markup Language (XML) format of higher education seekers group, using the iText 7 Java library tools, a PDF document is generated that contains the diploma pages to seekers with background images and information that depend on the higher education level and other parameters of seeker education. We used the Java Architecture for XML Binding technology for mapping the exported data to the corresponding objects of the user classes, we used high-level and low-level tools of the iText library version 7.1.0 for the construction of the PDF document, Java internationalization tools for to display information in Ukrainian and English. Analysis of the structure of the XML file with the data of higher education seekers exported from the USEDEI found inaccuracies, in particular, related to incorrectly selected names for tags that contain the graduation dates and the issue date of the document. In this paper, a method is proposed for adding a background image to pages of a multipage PDF document without a corresponding increase in the file size. The effectiveness of the developed software was proved by approbation in the summer of 2017 for bachelors and specialists of Zaporizhzhia State Engineering Academy and in the winter of 2018 - for masters.

**Keywords:** USEDEI, diploma of higher education, XML, JAVA, PDF, iText.

**Постановка проблеми.** Відповідно до статті 7 Закону України "Про вищу освіту" [1], заклади вищої освіти отримали право на виготовлення документів про вищу освіту: дипломів та додатків до них, які повинні містити перелік інформації, затверджений наказом Міністерства освіти і науки України від 12 травня 2015 р. № 525 "Про затвердження форм документів про вищу освіту (наукові ступені) державного зразка та додатків до них, зразка академічної довідки" [2]. Оскільки вся інформація, яка відтворюється в дипломах про вищу освіту, відповідно до статті 8 Закону України "Про вищу освіту" знаходиться в Єдиній державній Базі з питань Освіти (ЄДЕБО) [1], можлива автоматизація підготовки дипломів при вищу освіту для друку із використанням можливостей експорту даних з ЄДЕБО.

І хоча послуги по виготовленню дипломів можуть виконувати інші організації [3], на наш погляд, для закладу вищої освіти актуальною є розробка власного програмного забезпечення для можливості оперативного друку документів та внесення змін до функціональності такого програмного забезпечення. Окрім того, представляє певний інтерес дослідження і використання сучасних засобів програмної інженерії, зокрема такої розвиненої платформи з вільно розповсюджуваним програмним кодом, як Java, для ефективної розробки програмного забезпечення.

Для вирішення проблеми було необхідно:

- виконати аналіз структури даних та порівняння форматів XML-файлів експорту даних документів про освіту з ЄДЕБО [4-5], обравши дані, актуальні для програмного забезпечення, що розробляється;
- виконати проектування документу у форматі PDF із інформацією дипломів про вищу освіту здобувачів вищої освіти відповідно до наказу МОНУ від 12 травня 2015 р. №525 [2];
- виконати проектування архітектури програмного забезпечення, провести конструювання програмного забезпечення та підготовку його до дистрибуції;
- виконати апробацію роботи програмного забезпечення для вихідних даних здобувачів різних ступенів вищої освіти, експортованих з ЄДЕБО.

**Аналіз досліджень.** Традиційно парсинг XML в Java виконується з використанням таких інтерфейсів прикладного програмування (API) Java, як простий API для XML (Simple API for XML - SAX) або об'єктна модель документа (Document Object Model - DOM). Засоби першого відрізняються послідовним зчитуванням розмітки XML-файла та передачею програмному додатку подій при зустрічі розпізнаних елементів XML, які зазвичай оброблюються за допомогою функцій зворотного виклику. SAX-парсери відрізняються гарною продуктивністю, але не дозволяють змінювати структуру XML-файлу. DOM API базується на попередньому аналізі всього XML-файла та створенні для кожного його тега об'єкту-вузла Node, який інкапсулює ім'я, атрибути та значення атрибутів тега. Цей API дозволяє не тільки зчитувати у довільному порядку, але й змінювати XML-документ [6].

Але обидва класичних підходи до парсинга XML вимагають досить великого кодування для отримання значень тегів та їх атрибутів, які потім, як правило, інкапсулюються в об'єкти користувачьких класів, що надалі використовуються у програмі. Наступний рівень абстрагування - безпосереднє зчитування даних XML-файла та збереження їх у об'єктах користувачьких класів забезпечує технологія Java Architecture for XML Binding (JAXB) [7]. Для кожного XML-документа за допомогою мови описання структури XML-документа XML Schema може бути побудована відповідна схема. У разі попередньої підготовки користувачького класу, що інкапсулює необхідні у програмі властивості, які заповнюватимуться значеннями тегів та атрибутів XML-документу, така схема може бути виведена засобами JAXB автоматично за допомогою відповідних анотацій властивостей класів. Таке анотування забезпечує процедуру відображення (bind) XML-схеми на користувачький клас (рис. 1). Після такого відображення можливе використання технології демаршалінга (unmarshaling), яка дозволяє отримувати об'єкти користувачьких класів з властивостями зі значеннями із відповідних тегів та атрибутів XML-документу. Також можлива зворотна генерація XML-документу за даними об'єктів - технологія маршалінга (marshaling) [7].

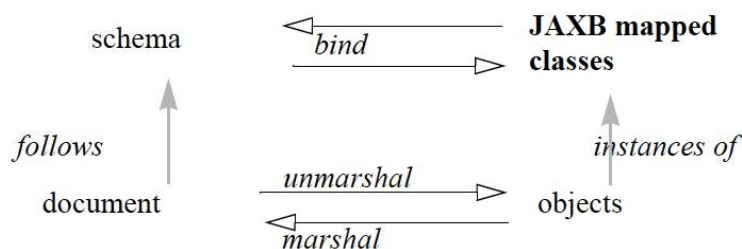


Рис. 1 - Відображення XML-файлу на об'єкти [7]

В роботі у якості формату вихідного документу з дипломами здобувачів вищої освіти обраний формат PDF, який дозволяє відображати та друкувати незмінний документ із даними різного типу на різних платформах та пристроях. Оскільки вбудованих до API платформи Java засобів генерації PDF-документів не існує, потрібно було обрати сторонню бібліотеку для забезпечення цього функціоналу. Дослідження в Інтернет показали, що найбільшою популярністю користуються бібліотеки Apache PDF та iText, а серед них користувачі більше застосовують iText [8]. Перевагами цієї бібліотеки є добра документованість, наявність книг [9-11] та сайту підтримки з великою кількістю прикладів використання засобів бібліотеки, розвинений функціонал, що дозволяє задовольнити усі потреби програмістів при роботі з PDF-файлами, на час написання статті актуальною була версія 7.1.0 бібліотеки.

Бібліотека iText надає як високорівневі засоби роботи з PDF-документом за допомогою об'єктів класів Document, які можуть містити об'єкти класів Paragraph, List, Image та інших, так і низькорівневі засоби - об'єкти класів PdfWriter, PdfDocument та PdfCanvas, які дозволяють, наприклад, програмно виводити графіки та додавати блоки тексту у довільні області сторінки.

**Виклад основного матеріалу і обґрунтування отриманих результатів.** Аналіз структури XML-файлу, експортованого з ЄДЕБО [4] та порівняння його зі структурою аналогічного файлу, експортованого попередньою версією програмного забезпечення ЄДЕБО [5] дозволяє заключити:

1) розробники змінили підхід до структури XML-файлу, а саме, якщо раніше дані для кожного здобувача зберігались як атрибути тегу <documentdata>, то зараз вони зберігаються у окремих тегах, окрім того, використовується CamelCase-нотація, що спрощує сприйняття назв XML-елементів (рис. 2);

2) новий формат містить більше даних, аніж попередній (74 проти 46), але відрізняється, на наш погляд, надлишковістю: для деяких даних зберігається і ідентифікатор, і назва, наприклад, "Ідентифікатор відзнаки" <AwardTypeId> і "Назва відзнаки" <AwardTypeName> (0 - без відзнаки, 3 - з відзнакою – незрозуміло, чому не 1). Теж саме відноситься до ідентифікатора і назви джерела даних, ідентифікатора і назви типу вихідного документа для дубліката, ідентифікатора і назви типу фінансування, ідентифікатора і назви форми навчання, ідентифікатора і назви курсу, ідентифікатора і назви статі. Однак явно надлишкові елементи - прізвище, ім'я та по-батькові випускника разом (fio та fioen) видалені.

3) на наш погляд, присутня помилка у виборі назви для дати закінчення навчального закладу <IssueDate> та дати видачі документа <GraduateDate> - з точки зору англійської мови потрібно було б назвати ці теги навпаки. Як результат, для випадку дублікату диплому, коли не співпадають зазначені дати, ми отримуємо:

```
<GraduateDate>16.10.2017 0:00:00</GraduateDate>
<IssueDate>30.06.1988 0:00:00</IssueDate>
```

що без документації може сприйматись як дата закінчення пізніша ніж дата видачі диплому.

4) перевагами нового формату є наявність додаткових тегів, що дозволяють зберігати назву освітньої програми здобувача двома мовами, а також теги для збереження даних назв та розрядів професій;

5) фактичний формат дати та часу тегів <GraduateDate> та <IssueDate> не відповідає документації (там вказаний формат без часу: ДД.ММ.РРРР [4]).

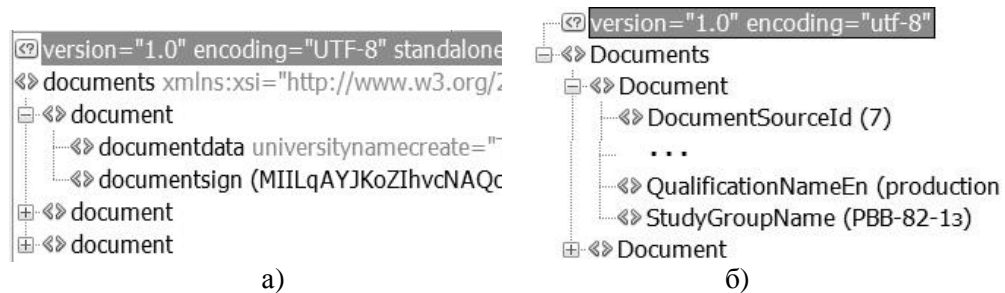


Рис. 2 - Структура XML-файлу з даними здобувачів вищої освіти, експортованими з ЄДЕБО: а) у попередній версії програмного забезпечення ЄДЕБО, б) у поточній версії програмного забезпечення ЄДЕБО

Для підготовки до друку така велика кількість даних була надлишковою, тому у спроектованому на архітектурному рівні моделі класі `Diploma` було обрано лише 22 елементи даних, які відображені у структурі класу на рис. 3. Відображення XML-документу на об'єкти цього класу конфігурується JAXB-анотаціями [7].

Для організації демаршалінга довільних класів архітектурного рівня моделі з використанням технології узагальнень (Java Generics) був розроблений універсальний парсер з методом демаршалінга XML-файлу у об'єкти аргументу узагальненого типу, вихідний код класу якого наведений нижче:

```
public class JAXBParser<T> {  
  
    final Class<T> typeParameterClass;  
  
    public JAXBParser(Class<T> typeParameterClass) {  
        this.typeParameterClass = typeParameterClass;  
    }  
  
    public T getObject(File file) throws JAXBException {  
  
        JAXBContext context =  
            JAXBContext.newInstance(typeParameterClass);  
        Unmarshaller unmarshaller =  
            context.createUnmarshaller();  
        T object = (T) unmarshaller.unmarshal(file);  
        return object;  
    }  
}
```

Такий клас розміщений на рівні сервісів (рис. 4). Об'єкт цього класу ініціалізується об'єктом `Class` для кореневого класу моделі XML-документу, у нашому випадку:

```
JAXBParser<Diplomas> parser = new JAXBParser<>(Diplomas.class);
```

Виклик метода `getObject(File file)` цього об'єкта з передачею як аргументу об'єкта `File`, пов'язаного з XML-файлом з даними, повертає об'єкт `Diplomas` зі списком об'єктів `Diploma` з даними здобувачів вищої освіти.

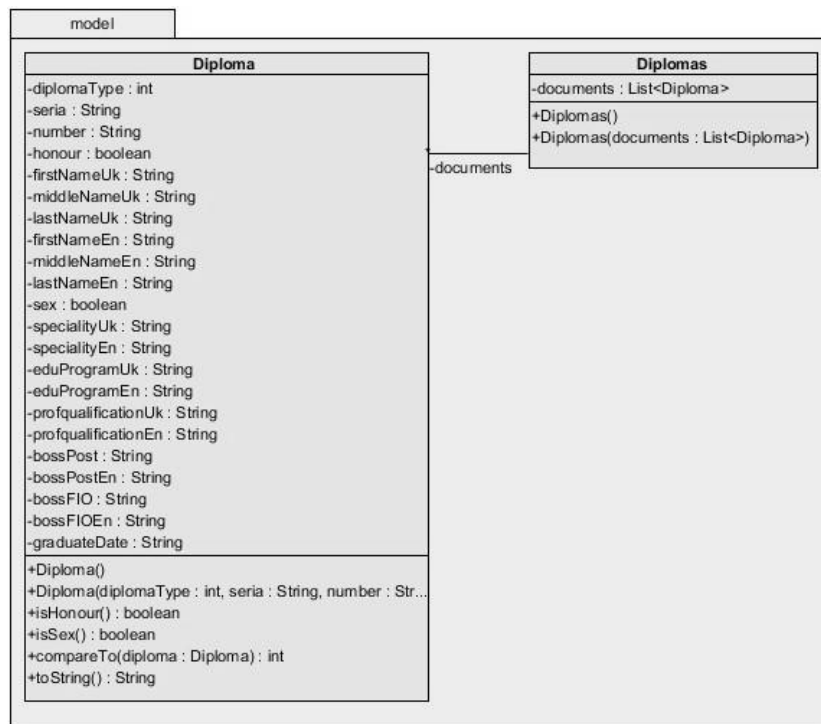


Рис. 3 - Класи архітектурного рівня моделі

Після отримання об'єктів з даними дипломів здобувачів вищої освіти настає етап генерування PDF-документу із дипломами, готовими до друку, за допомогою засобів бібліотеки iText. Перелік jar-файлів з засобами бібліотеки iText, який вказується у [10], для реалізації програмного забезпечення, що описується, був надлишковим. Дослідженням встановлена необхідність тільки таких файлів бібліотеки: itext7-kernel-7.0.2.jar, itext7-io-7.0.2.jar, itext7-layout-7.0.2.jar. Також знадобилися файли додаткових бібліотек slf4j-api-1.7.25.jar та slf4j-simple-1.7.25.jar - для системи організації журналів (логів) та commons-io-2.5.jar - для завантаження байтів фонових зображень при запуску програми із jar-файлу.

На рис.5 наведений приклад сторінки згенерованого PDF-документа із зазначенням методів класу PdfGenerator, відповідальних за побудову частин сторінки.

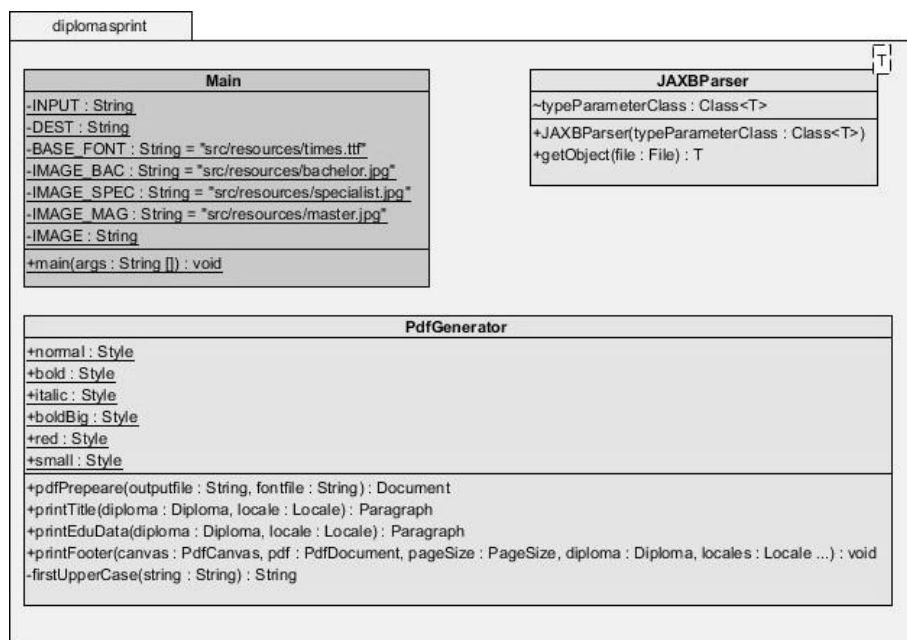


Рис. 4 - Класи архітектурного рівня сервісів



Рис. 5 – Зображення прикладу сторінки згенерованого PDF-документа із зазначенням методів PdfGenerator, відповідальних за побудову частин сторінки

Фоновий рисунок відрізняється кольором захисних елементів для дипломів різних рівнів вищої освіти: бакалавр, спеціаліст та магістр, та автоматично підставляється у залежності від значення властивості `diplomaType` об'єкту `Diploma`. Звертаємо увагу на використання об'єктів утилітарного класу `Java Locale` для організації двомовного виведення інформації. Метод `Document pdfPrepare(String outputfile, String fontfile)` класу `PdfGenerator` за допомогою засобів бібліотеки `iText` виконує підготовку до виведення інформації у PDF-документ [11]:

- визначає розмір, орієнтацію та відступи від краю сторінки диплому;
- створює об'єкт `PdfFont` з підтримкою символів кирилиці;
- визначає об'єкти стилів шрифту різного розміру та написання, також передбачений стиль шрифту червоного кольору для виведення назви диплому з відзнакою;
- конфігурує виведення інформації у дві колонки та визначає розміри колонок.

Цей метод приймає посилання на місце розташування результуючого PDF-файлу та на файл шрифту (можливо використовувати звичайний `True Type` шрифт з символами кирилиці). Повертає метод об'єкт високорівневого класу `Document`, який і є результуючим PDF-документом.

Метод `Paragraph printTitle(Diploma diploma, Locale locale)` виводить інформацію про рівень вищої освіти диплому, признак відзнаки, якщо є (у цьому випадку назва диплому виводиться червоним кольором та додається рядок про відзнаку), серію та номер диплому, прізвище, ім'я та по-батькові здобувача та назву навчального закладу, який він закінчив і рік завершення навчання. Особливістю методу є використання засобів інтернаціоналізації `Java`, включаючи використання пакетів ресурсів (`ResourceBundle`) разом з файлами властивостей та форматування повідомлень за допомогою засобів класу `MessageFormat`. Ці засоби дозволили повторно використовувати код методу та адаптувати вивід інформації в залежності від об'єктів `Locale`, які передаються методу як аргумент. Також за допомогою вказаних технологій коректуються закінчення україномовних слів у залежності від статі здобувача.

Метод `Paragraph printEduData(Diploma diploma, Locale locale)` виводить інформацію про параметри здобутої випускником кваліфікації: рівень вищої освіти, спеціальність (або напрям для бакалавра), назву освітньої програми та професійну кваліфікацію (за наявності). Використовуються ті ж самі засоби, як і в методі `printTitle`.

Додання нижньої частини диплому з посадою, прізвищем та підписом керівника та іншими даними на високому рівні виявилось проблематичним, оскільки внаслідок різного розміру даних, що виводяться описаними вище методами, його положення б змінювалось. Тому було прийнято

рішення в методі `void printFooter(PdfCanvas canvas, PdfDocument pdf, PageSize pageSize, Diploma diploma, Locale... locales)` використати об'єкти низькорівневого класу `PdfCanvas` та класу `Canvas`, що зв'язує низькорівневі та високорівневі засоби бібліотеки `iText` і дозволяє абсолютне позиціонування інформації, що виводиться [11]. Окрім низькорівневих об'єктів `PdfCanvas` та `PdfDocument`, метод приймає об'єкт розміру сторінки, документ з даними добувача та масив локалей, що використовуються (передаються як аргументи змінної довжини `varargs`). Оскільки тег дати видачі диплома `<GraduateDate>` зберігає окрім дати і час, і потрібно виводити дату у повному форматі з урахуванням правил виведення для української та англійської локалі, в методі були застосовані засоби `Java Date and Time API` – клас `DateTimeFormatter`.

Клас `Main` містить константи, що визначають шляхи до файлів-ресурсів: файлу шрифту та файлів фонових зображень для трьох рівнів вищої освіти. Назва вихідного XML-файлу з даними користувачів передається при запуску програми як аргумент командного рядку, а результуючий PDF-файл з назвою, як у XML-файла, але з розширенням `.pdf`, зберігається у підкаталозі `result` каталогу, з якого виконувався запуск програми. Головний метод додатку `void main(String[] args)`, розміщений в цьому класі, за допомогою об'єкту класу `JAXBParser`, параметризованого класом архітектурного рівня моделі `Diplomas`, отримує список об'єктів `Diploma`, після чого список відсортовується за прізвищами здобувачів.

Цікавою особливістю програми є спосіб додання фонового рисунку до сторінок PDF-документу. Річ у тому, що багатократне додання звичайного рисунку – об'єкту `Image` до документу не збільшує розмір PDF-файлу, тобто байти рисунку у документі зберігаються тільки один раз, а дублюються посилання на них [11]. Але для додання фонового рисунку доводиться використовувати низькорівневий об'єкт `PdfCanvas`, специфічний для кожної сторінки PDF-документу. При цьому з'являється проблема необгрунтованого збільшення розміру результуючого PDF-файлу пропорційно кількості сторінок з однаковим фоновим зображенням в ньому [12]. Вказану проблему вдалося вирішити доданням до `PdfCanvas` об'єкту `PdfImageXObject`, що є "обгорткою" над фоновим рисунком, в обох випадках об'єкти зображень створюються через об'єкт `ImageData`:

```
ImageData imgData = ImageDataFactory.create(IMAGE);
```

Але у першому випадку об'єкт `ImageData` додається безпосередньо:

```
canvas.addImage(imgData, pageSize, false);
```

А у разі оптимізації – через об'єкт `PdfImageXObject`:

```
PdfImageXObject imgObject = new PdfImageXObject(imgData);  
canvas.addXObject(imgObject, rectangle);
```

Аргумент `rectangle` визначає область сторінки для додавання зображення, при розтягненні зображення на всю сторінку він може визначатись через розмір сторінки:

```
Rectangle rectangle = new Rectangle(pageSize.getWidth(),  
                                   pageSize.getHeight());
```

Дослідженням результуючих PDF-файлів у програмі `PDFXplorer` [13] було з'ясовано, що безпосереднє додання фонових зображень до об'єкту `PdfCanvas` призводить до наявності декількох об'єктів `PDF XObject`. Це зумовлено тим, що метод `PdfXObject addImage(ImageData image, Rectangle rect, boolean asInline)` класу `PdfCanvas` створює кожен раз окремий об'єкт `PdfXObject`, навіть для одного і того ж об'єкту `ImageData`. Натомість, використання методу `PdfCanvas addXObject(PdfXObject xObject, Rectangle rect)` того ж класу у якості аргументу приймає об'єкт `PdfXObject`, який відповідно до специфікації PDF - є зовнішнім до основного контенту об'єктом [14]. Вочевидь, тому останній метод використовує посилання на один і той же об'єкт, який "обгортає" фоновий рисунок. Було виконано порівняння розміру згенерованих представленим програмним забезпеченням багатосторінкових файлів з однаковими фоновими рисунками, що були додані до канви безпосередньо та з використанням `PdfImageXObject`. Для файлу формату `JPG` розміром 1981250 байт 5-сторінковий PDF-документ з

безпосереднім доданням фонового рисунку до канви мав розмір 9909084 байта, у той час як 5-сторінковий PDF-документ з доданням того ж фонового рисунку до канви через об'єкт PdfImageXObject мав розмір 1983364 байти, тобто у 4,996 разів менше.

Після додання фонового зображення у методі void main(String[] args) до документу додаються об'єкти параграфів, що створюються вище описаними методами об'єкту PdfGenerator. Ці операції виконуються у циклі з кількістю ітерацій, що дорівнює кількості дипломів здобувачів вищої освіти у вихідному XML-файлі. При цьому кінець першої колонки організовується додаванням до PDF-документу об'єкта AreaBreak, а кінець сторінки – об'єкта AreaBreak з передачею його конструктору значення AreaBreakType.NEXT\_PAGE як аргументу.

Оскільки взаємодія програми з користувачем дуже проста: запуск програми з командного рядка з зазначенням імені XML-файлу як параметра, було вирішено підготувати дистрибутив програми у вигляді jar-файлу. При цьому було вирішено за доцільне упакувати разом з файлами програми файли ресурсів: файл шрифту, файли фонових зображень, та файли властивостей для української та англійської локалей. Завантаження файлів шрифту та зображень було організоване за допомогою методу static byte[] toByteArray(InputStream input) класу IOUtils бібліотеки Apache Commons IO [15]:

```
byte[] fontBytes = IOUtils.toByteArray(  
    Main.class.getResource(fontfile).openStream());
```

та створенням з масиву байтів об'єкту PdfFont (та об'єктів ImageData):

```
PdfFont font = PdfFontFactory.createFont(fontBytes,  
    PdfEncodings.IDENTITY_H);
```

або

```
ImageData imgData = ImageDataFactory.create(imageBytes);
```

Для запуску програми на комп'ютері користувача повинно бути встановлено середовище для запуску програм Java (JRE) або комплект розробника програм на Java (JDK) версії не нижче 7. При цьому команда запуску програми буде

```
java -jar DiplomasPrint.jar MET-13-1.xml
```

У разі встановлення на комп'ютері користувача останньої 9 версії JRE або JDK команда запуску додатку зміниться внаслідок переходу Java до концепції модулів та виключення зі стандартного модуля java.se засобів Java EE, у тому числі JAXB API [16]. У цьому випадку потрібно підключати модуль з засобами JAXB:

```
java -jar --add-modules java.xml.bind DiplomasPrint.jar MET-13-1.xml
```

Представлене програмне забезпечення було апробоване при підготовці до друку дипломів здобувачів вищої освіти у Запорізькій державній інженерній академії: бакалаврів та спеціалістів у червні 2017 р та дипломів магістрів у січні 2018 р. Апробація довела ефективність програми за рахунок зменшення часу на друк та зменшення помилкових роздруківок, оскільки раніше виконувалася окремих друк фонового рисунка та інформації диплому.

**Висновки та перспективи подальшого дослідження.** Аналіз структури XML-файлу з даними здобувачів вищої освіти, експортованими з ЄДЕБО, виявив неточності, зокрема, пов'язані з невірно вибраними назвами для тегів, що зберігають дати закінчення навчального закладу та дати видачі документа.

Використання технології відображення елементів XML-файлів на об'єкти користувацьких класів (JAXB) суттєво зменшили обсяг кодування для реалізації парсингу та підвищили надійність роботи програми. Розроблений типізований JAXB парсер обмежує використання типів аргументом типу, що надає відомі переваги [17]. Застосування засобів технології інтернаціоналізації додатків Java дозволило повторно використовувати методи генерування даних дипломів на українській та англійській мовах. Використання технології форматування повідомлень дозволило забезпечити необхідну зміну порядку слів та корекцією закінчень для різних локалей.

В роботі запропонований спосіб додання фонового рисунку до сторінок багатосторінкового PDF-документу без відповідного збільшення розміру файлу.



Виконана адаптація програмного коду для формування JAR-дистрибутиву програмного забезпечення та досліджені особливості його запуску під найновішою 9 версією Java.

Результати аналізу структури XML-файлу, з даними експортованими з ЄДЕБО, надіслані до розробників програмного забезпечення ЄДЕБО зі сподіванням їх врахування при модернізації ЄДЕБО.

1. "Про вищу освіту" Закон України від 1 липня 2014 р № 1556-VII // Відомості Верховної Ради (ВВР). – К.: ПАРЛАМЕНТСЬКЕ ВИДАВНИЦТВО, 2014, № 37-38, – С. 2004.
2. "Про затвердження форм документів про вищу освіту (наукові ступені) державного зразка та додатків до них, зразка академічної довідки" наказ Міністерства освіти і науки України від 12 травня 2015 р. № 525 [Електронний ресурс] / Сайт "Законодавство України". – Режим доступу: <http://zakon3.rada.gov.ua/laws/show/z0551-15>.
3. Виготовлення документів про вищу освіту [Електронний ресурс] / Сайт інформаційно-виробничої системи "Освіта". – Режим доступу: <https://osvita.net/ua/vigotovlennya-dokumentiv-pro-vishhu-osvitu>.
4. Опис XML-файлу для експорту даних документів про освіту з ЄДЕБО [Електронний ресурс] / Сайт Державного Підприємства "Інфоресурс". – Режим доступу: <https://www.inforesurs.gov.ua/uploads/files/EDEBOFormat-XML-Zamovl-2017.pdf>.
5. Опис XML файлу для експорту даних документів про освіту (з використанням попередньої версії програмного забезпечення ЄДЕБО) [Електронний ресурс] / Сайт Державного Підприємства "Інфоресурс". – Режим доступу: [https://www.inforesurs.gov.ua/uploads/files/1434111418\\_edebo.format-xml-zamovl.dvo.pdf](https://www.inforesurs.gov.ua/uploads/files/1434111418_edebo.format-xml-zamovl.dvo.pdf).
6. McLaughlin B. Java and XML. Solutions to Real-World Problems, 3rd Edition / B. McLaughlin, J. Edelson. – Sebastopol : O'Reilly Media, 2009. – 480 p.
7. JSR-000222 Specification. The Java™ Architecture for XML Binding (JAXB) 2.3 Final Release July 11, 2017 [Electronic resource] / – Access mode: [http://download.oracle.com/otndocs/jcp/jaxb-2\\_3-mrel3-eval-spec/index.html](http://download.oracle.com/otndocs/jcp/jaxb-2_3-mrel3-eval-spec/index.html).
8. Compare Apache PDFBox and iText's popularity and activity [Electronic resource] – Site Awesome Java / – Access mode: <https://java.libhunt.com/project/apache-pdfbox/vs/itext>.
9. Lowagie B. iText in Action. Second Edition. / B. Lowagie. – Stamford : Manning Publications Co., 2011. – 618 p.
10. Lowagie B. iText 7: Jump-Start Tutorial. / B. Lowagie. – Lean Publishing, 2016. – 100 p.
11. Lowagie B. iText 7: Building Blocks. / B. Lowagie. – Lean Publishing, 2016. – 224 p.
12. Kolomoets G. iText 7 read background image from jar file [Electronic resource] – Site Stack overflow / – Access mode: <https://stackoverflow.com/questions/44871451/itext-7-read-background-image-from-jar-file>.
13. PDFXplorer [Electronic resource] / Site O2 Solutions – Access mode: <http://www.o2sol.com/pdfexplorer/overview.htm>
14. PDF 32000-1:2008. Document management - Portable Document Format - Part I: PDF 1.7, First Edition [Electronic resource] / Site Adobe Systems Incorporated – Access mode: [http://www.adobe.com/content/dam/acom/en/devnet/acrobat/pdfs/PDF32000\\_2008.pdf](http://www.adobe.com/content/dam/acom/en/devnet/acrobat/pdfs/PDF32000_2008.pdf).
15. Apache Commons IO [Electronic resource] / Site Apache Commons – Access mode: <https://commons.apache.org/proper/commons-io>.
16. Sander M. Java 9 Modularity. / M. Sander, P. Bakker – Sebastopol : O'Reilly Media, 2017. – 366 p.
17. Naftalin M. Java Generics and Collections / M. Naftalin, Ph. Wadler – Sebastopol : O'Reilly Media, 2007. – 286 p.