Міністерство освіти і науки, молоді та спорту України

Державний заклад «Луганський національний університет   
імені Тараса Шевченка»

|  |  |
| --- | --- |
| Факультет (інститут) | Фізики, Математики та Інформаційних Технологій |
| *(повна назва)* |
| Кафедра | Інформаційних Технологій та Систем |
|  | *(повна назва)* |

|  |  |  |
| --- | --- | --- |
|  | **ЗАТВЕРДЖУЮ** | |
|  | Завідувач кафедри ІТС | |
|  |  | |
|  |  |  |
|  |  | *(ПІП)* |
|  | “\_\_\_\_ “ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ р. | |

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

**" Система навчального призначення «Лектор» "**

**ІТС.ПІ4.0512-03-ПЗ**

|  |  |
| --- | --- |
| **ПОГОДЖЕНО** | **ВИКОНАВЕЦЬ** |
| Керівник кваліфікаційної роботи  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Ю.Л. Тихонов  “\_\_\_\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_20\_\_ р | Студент групи \_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Г.А. Сичова  “\_\_\_\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_20\_\_ р |
| Нормоконтроль  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_І.П. Сіборов |  |

Луганськ 2016

ЗМІСТ

[ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ 5](#_Toc451427689)

[ВСТУП 6](#_Toc451427690)

[РОЗДІЛ 1 8](#_Toc451427691)

[1. Дистанційна освіта в Україні та світі 8](#_Toc451427692)

[1.1. Аналіз дистанційної освіти у світі 8](#_Toc451427693)

[1.2. Аналіз дистанційної освіти в Україні 9](#_Toc451427694)

[1.3. Міжнародні програми дистанційної освіти 10](#_Toc451427695)

[1.3.1. Coursera 11](#_Toc451427696)

[1.3.2. ЄШКО 12](#_Toc451427697)

[1.4. Висновки до розділу 13](#_Toc451427698)

[1.5. Постановка задачі 13](#_Toc451427699)

[РОЗДІЛ 2 15](#_Toc451427700)

[2. Алгоритмічне конструювання системи навчального призначення 15](#_Toc451427701)

[2.1. Алгоритм тестування 15](#_Toc451427702)

[2.2. Криптографічний алгоритм шифрування Rijndael 17](#_Toc451427703)

[2.3. Алгоритм завантаження та обробки навчальних матеріалів 17](#_Toc451427704)

[2.1. Алгоритм завантаження та обробки розкладу 19](#_Toc451427705)

[2.2. Алгоритм обробки персональних даних користувача 20](#_Toc451427706)

[2.3. Алгоритм пошуку навчальних матеріалів на локальному диску 22](#_Toc451427707)

[2.4. Висновки 24](#_Toc451427708)

[РОЗДІЛ 3 25](#_Toc451427709)

[3. Програмне конструювання системи навчального призначення 25](#_Toc451427710)

[3.1. Обґрунтування вибору засобів розробки 25](#_Toc451427711)

[3.1.1. Обґрунтування вибору мови програмування 25](#_Toc451427712)

[3.1.2. Обґрунтування вибору середовища розробки 25](#_Toc451427713)

[3.1.3. Обґрунтування вибору форматів файлів 27](#_Toc451427714)

[3.2. Основні модулі програмного засобу 29](#_Toc451427715)

[3.2.1. Діаграма класів 29](#_Toc451427716)

[3.2.2. Клас MainWindow 30](#_Toc451427717)

[3.2.3. Клас StartPage 30](#_Toc451427718)

[3.2.4. Клас CourcePage 33](#_Toc451427719)

[3.2.5. Клас SubjectPage 34](#_Toc451427720)

[3.2.6. Клас LectionPage 36](#_Toc451427721)

[3.2.7. Клас TestPage 37](#_Toc451427722)

[3.2.8. Клас UserDataPage 39](#_Toc451427723)

[3.2.9. Бібліотека класів Lecturer.PDFControl 40](#_Toc451427724)

[3.2.10. Бібліотека класів Lecturer.Data 42](#_Toc451427725)

[3.2.11. Клас StorageProcessor 42](#_Toc451427726)

[3.2.12. Клас ExcelFileProcessor 43](#_Toc451427727)

[3.2.13. Клас XMLProcessor 44](#_Toc451427728)

[3.2.14. Клас CryptoProcessor 46](#_Toc451427729)

[3.2.15. Класи-сутності 46](#_Toc451427730)

[3.2.15.1. Клас Cource 47](#_Toc451427731)

[3.2.15.2. Клас Subject 48](#_Toc451427732)

[3.2.15.3. Клас Topic 49](#_Toc451427733)

[3.2.15.4. Клас Quiz 50](#_Toc451427734)

[3.2.15.5. Клас QuizItem 50](#_Toc451427735)

[3.2.15.6. Класи Department та Specialty 51](#_Toc451427736)

[3.2.15.7. Клас UserData 52](#_Toc451427737)

[3.3. Висновки до розділу 53](#_Toc451427738)

[4. Тестування системи навчального призначення 54](#_Toc451427739)

[4.1. Висновки до розділу 54](#_Toc451427740)

[ВИСНОВКИ 55](#_Toc451427741)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 56](#_Toc451427742)

[ДОДАТОК А. Діаграми 59](#_Toc451427743)

[ДОДАТОК Б. Блок-схема алгоритму тестування 64](#_Toc451427744)

[ДОДАТОК В. Блок-схема алгоритму завантаження та обробки навчальних матеріалів 65](#_Toc451427745)

[ДОДАТОК Г. Програмні коди модульних тестів 66](#_Toc451427746)

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД - база даних

ІС - інформаційна система

ІТ - інформаційні технології

ООП - об’єктно-орієнтоване програмування

ОС - операційна система

ПЗ - програмне забезпечення

ДНЗ - державний навчальний заклад

МВОК - масові відкриті дистанційні онлайн-курси

ДНЗ - державний навчальний заклад

# ВСТУП

Актуальність роботи

Дистанційна освіта з кожним роком набуває популярності в усьому світі. Свідоцтвом цьому є велика кількість дистанційних курсів, онлайн-платформ, а також введення змішаної або виключно дистанційної форми освіти навчальними закладами в усьому світі.

Україна не стала виключенням: хоча й не так стрімко, як у США- або Великобританії, в нашій країні створюються та розвиваються центри дистанційної освіти, державні заклади пропонують абітурієнтам онлайн-курси, з’являються переклади популярних міжнародних дистанційних курсів українською мовою (наприклад, на платформі Coursera).

Тому зростає потреба в новітніх розробках у сфері онлайн-освіти, в першу чергу для державних навчальних закладів. Це зумовлюється спеціалізацією ДНЗ, рівнем розвитку ІС, підготовленістю кадрів, а також суворішими вимогами до контролю знань учнів.

Мета і задачі роботи

Метою бакалаврської роботи є розробка системи навчального призначення «Лектор». ПЗ повинно надавати користувачу ефективний інструмент для вивчення теоретичного матеріалу навчальних дисциплін.

Досягнення мети включало розв’язання таких задач:

* аналіз дистанційної освіти в Україні та світі, міжнародних програм;
* розробка оптимальних алгоритмів для обробки файлів та даних, із якими працюватиме система;
* опис основних класів, які повинні бути реалізовані у системі;
* підготування навчальних матеріалів;
* реалізація системи.

Об’єктом дослідження є дистанційна освіта.

Предметом дослідження є процес вивчення та закріплення теоретичного матеріалу при дистанційному навчанні.

Практичне значення одержаних результатів

Одержане ПЗ, що надає засоби для вивчення та закріплення теоретичного матеріалу в умовах дистанційної освіти, довготривалого збереження навчальних матеріалів на локальному комп’ютері користувача у структурованому вигляді. Таким чином користувач має можливість вивчати, закріплювати та повторювати теоретичний матеріал навіть в умовах відсутності надійного інтернет-з’єднання.

Основним результатом є розроблений ПЗ.

# РОЗДІЛ 1

# Дистанційна освіта в Україні та світі

Поняття «дистанційна освіта» включає широкий спектр освітніх програм, починаючи з курсів підвищення кваліфікації, які не мають акредитації, закінчуючи акредитованими програмами вищої освіти, які реалізують можливість тісного спілкування студентів з викладачами та однокурсниками, як це відбувається при очному навчанні. Щоб забезпечити ефективну взаємодію викладачів та студентів, для дистанційного навчання використовуються комп'ютерні програми, спеціалізовані веб-сайти, телефонний зв’язок, електронна та звичайна пошта, онлайн-чати та ін.

В силу своєї зручності, дистанційна освіта стає надзвичайно популярною формою навчання. Вона усуває основний бар'єр, що утримує багатьох професіоналів та ділових людей від продовження освіти, позбавляючи від необхідності відвідувати заняття за встановленим розкладом. Люди, які навчаються дистанційно, можуть вибирати зручний для себе темп та час занять згідно із власним розкладом; спосіб навчання; мають ширший вибір навчальних закладів.

Саме тому кількість програм дистанційної освіти невпинно збільшується. Розглянемо ситуацію із дистанційною освітою у світі та Україні, а також міжнародні освітні програми.

# Аналіз дистанційної освіти у світі

Після революційних подій в онлайн-навчанні в США, особливо з появою масових відкритих онлайн-курсів у 2010-х роках, популярність дистанційної освіти невпинно зростає і в інших частинах світу: в Азії, Європі та Австралії.

У порівнянні з традиційною освітою, дистанційна має немало переваг. Багато студентів зараз обирають онлайн-курси через їх доступність, гнучкість розкладу та низьку вартість. Також дистанційна освіта сприяє соціальній інтеграції людей з уразливих груп населення (таких, як жителі країн третього світу, інваліди та ін.).

Саме через це збільшується і обсяг, і типи дистанційного навчання: студент може обрати онлайн, відкрите або змішане навчання, запропоноване спеціалізованим закладом дистанційної освіти або традиційними університетами. А з розвитком інформаційних технологій інтерес як учнів, так і навчальних закладів до дистанційної освіти тільки зростає.

Наведемо декілька прикладів:

* у 2010 році в Австралії 12% всіх студентів навчалися дистанційно, в той час як 7% навчалися за змішаними програмами [12].;
* у 2012 році в університетах Європи кількість студентів, завдяки введенню дистанційних курсів збільшилася на 15-20% [12].

За словами Генерального Секретаря Європейської Асоціації Університетів Дистанційної Освіти Піта Хендерікса, «відкрита та дистанційна освіта стає важливою відповіддю сучасним освітнім викликам та істотно впливає на наш освітній ландшафт» [12].

# Аналіз дистанційної освіти в Україні

Як зазначено на офіційному сайті Міністерства Освіти і Науки України, «дистанційно в Україні можуть навчатися громадяни, які мають середню, професійну, вищу освіту, а також ті, що мають можливість виконувати дистанційно необхідні завдання за допомогою освітніх технологій. Процес навчання побудований на використанні різних комунікаційних засобів. По закінченню такого навчання, студенти отримують відповідні сертифікати.

Термін навчання на базі середньої вищої освіти становить 6 років. Середньої професійної освіти – 4,5 роки. На базі вищої неюридичної освіти – 3 роки [9]»

Наразі державних закладів, які мають центри дистанційної освіти, в Україні не менш десяти (за даними офіційного сайту МОН України). До їх числа входять: Київський Національний Університет імені Тараса Шевченка, Івано-Франківський національний технічний університет нафти та газу, Львівський Національний Університет «Львівська Політехніка» та ін.

З кожним роком кількість навчальних закладів в Україні (як державних, так і комерційних), що мають центри дистанційної освіти, росте. Так, наприклад, у списку центрів дистанційної освіти МОН України поки не вказано Луганський Національний Університет імені Тараса Шевченка.

Перші в Україні МВОК проходили на базі КНУ імені Тараса Шевченка у 2013 році. У 2014 році було засновано два проекти онлайн-освіти: EdEra та Prometheus.

Prometheus є громадським проектом МВОК, що надає доступ до онлайн-курсів, розроблених в українських університетах, а також курси підготовки до зовнішнього незалежного оцінювання. На офіційному сайті зазначено, що з 2015 року проект співпрацює з провідними українськими університетами, а саме: КНУ ім. Тараса Шевченка, Києво-Могилянською академією, Українським католицьким університетом та Львівською IT школою (LITS).

EdEra – це освітній проект, що створює повноцінні онлайн-курси та супроводжувальні матеріали широкого профілю. Метою проекту є намір зробити освіту в Україні доступною та якісною на зразок західних найкращих освітніх програм. У якості прикладу для рівняння зазначено онлайн-курс «Circuits and electronics» Массачусетського Технологічного Університету. Наразі проект має дев’ять безкоштовних онлайн-курсів та бібліотеку електронних книжок до них.

# Міжнародні програми дистанційної освіти

Серед великого розмаїття міжнародних програм розглянемо найбільш популярну у світі англомовну освітню платформу Coursera та відому в Україні та Європі ЄШКО.

# Coursera

Coursera – це освітня платформа, яка пропонує онлайн-курси від провідних університетів та організацій світу, у тому числі Вашингтонського та Єльського університетів (США), Лондонського та Манчестерського університетів (Великобританія), Токійського університету (Японія), Барселонського університету (Іспанія), корпорації IBM та спільноти National Geographic.

Згідно даних з офіційного сайту, Coursera є партнером 142 організацій та навчальних закладів з 28 країн світу [6].

Навчальні програми на Coursera побудовано на чотирьох концепціях:

* ефективне дистанційне навчання у формі відео-уроків;
* поетапне оволодіння предметом, яке допомагає студентам повністю зрозуміти матеріал, перш ніж переходити до вивчення більш складного (цей підхід засновано на розробках педагога-психолога Бенджаміна Блума);
* завдання для взаємної оцінки. Багато курсів включають у себе завдання, які не піддаються автоматичному оцінюванню (наприклад, запис відео із відповіддю на питання, написання есе). Тому студенти мають можливість оцінювати і рецензувати роботи одне одного. Численні дослідження довели, що такий підхід забезпечує учням не тільки точну оцінку їх робіт, а й цінний досвід перевіряючого.
* змішане навчання. Багато університетів, з якими співпрацює освітня платформа, використовують її для розширення можливостей очного навчання студентів. Дослідження показали, що змішана модель освіти підвищує залученість студентів у навчальний процес, збільшує їх відвідуваність і покращує успішність [7].

Перевагами навчання на платформі Coursera є:

* розмаїття курсів та організацій, які їх надають;
* велика кількість безкоштовних курсів;
* можливість отримати сертифікат про проходження курсу;
* матеріал викладається у вигляді коротких відео (у середньому 10 хвилин), що не дозволяє студенту втрачати концентрацію;
* наявність різноманітних засобів оцінювання знань студента: тестування, есе, контрольні питання, проекти, запис відео-відповіді та ін.

Серед недоліків можна відмітити:

* для вивчення курсів потрібне стабільне інтернет-підключення;
* курси викладаються різними мовами і потребують субтитрів або перекладу, що підвищує вартість та складність навчання;
* для більшості курсів не надається теоретичний матеріал у текстовому вигляді.

# ЄШКО

ЄШКО (абр. Європейська Школа Кореспондентської Освіти) – це міжнародна дистанційна школа, яка має філіали у наступних європейських країнах: Голландії, Польщі, Росії, Україні, Білорусі, Казахстані, Румунії, Іспанії та Угорщині. ЄШКО-Україна пропонує 86 дистанційних курсів, більшість з яких (більше 30) – іноземні мови [8].

Усі курси надаються у вигляді журналів, до деяких (в особливості, мовних) додаються аудіозаписи. Журнали можна отримувати в електронному вигляді, завантажуючи з сайту, або у друкованому, на пошті. Тривалість курсу визначає студент: є можливість отримувати по одному журналу на місяць, або увесь курс за раз.

У кожному журналі розміщено два уроки, в яких викладається і теорія, і практика. Для кожного уроку додається контрольне завдання, яке необхідно відправити викладачеві онлайн або поштою.

До переваг дистанційного навчання у ЄШКО можна віднести:

* наявність пробних журналів до кожного курсу;
* можливість обрати, в якому вигляді отримувати матеріал – електронному або друкованому;
* теоретичний та практичний матеріал в кожному уроці;
* проходження курсу перевіряється викладачем;
* по завершенні курсу студент отримує сертифікат.

Недоліками є:

* вартість журналу (на момент проведення аналізу складає близько 200 гривень);
* стислість викладеного теоретичного матеріалу;
* потреба у додаткових навчальних матеріалах: словниках, довідниках, програмах, курсах іншого рівню складності та ін., посилання на які зустрічаються на полях журналу.

# Висновки до розділу

Таким чином можна зробити висновок, що дистанційна освіта набуває розвитку у багатьох країнах світу, у тому числі в Україні. Покращується якість та кількість міжнародних курсів, з’являються нові методи навчання (наприклад, змішане навчання). Але існують недоліки, які ще потрібно допрацювати.

Тому зростає потреба у засобах дистанційної освіти, які компенсують недоліки вже існуючих систем. Загальним недоліком описаних вище освітніх програм є недостатність теоретичного матеріалу.

# Постановка задачі

Метою даної роботи є створення системи навчального призначення «Лектор» для вивчення теоретичного матеріалу дисциплін та контролю знань користувача.

На основі поставленої мети можна виділити задачі, які потрібно вирішити:

* обробка даних користувача. Сюди входять дані про підрозділ, у якому навчається студент, спеціальність, курс та поточний семестр, ім’я та прізвище користувача (користувач повинен мати змогу змінити введену інформацію);
* завантаження та обробка навчальних матеріалів та даних про них з файлового сервера на локальний диск. Мається на увазі завантаження та обробка архіву з файлами, які завантажуються з файлового сервера; завантаження та обробка файлу з розкладом дисциплін на поточний семестр;
* пошук навчальних матеріалів на локальному диску, обробка та представлення лекційного матеріалу. Дана задача потребує взаємодії зі стороннім програмним забезпеченням для роботи з файлом формату PDF;
* оцінювання знань користувача з вивченої дисципліни, що полягає у пошуку файлу тестування з теми, його дешифрування (якщо потрібно) та представлення у графічному вікні, перевірки проходження тестування та занесення даних (за потребою) до файлу користувача;

# РОЗДІЛ 2

# Алгоритмічне конструювання системи навчального призначення

Алгоритм – це набір інструкцій, що описують порядок дій, необхідних для досягнення певного результату. Незалежні інструкції можуть виконуватися в довільному порядку.

Далі у розділі буде описано основні алгоритми, які використовувалися під час розробки системи «Лектор».

# Алгоритм тестування

Тестування є невід’ємним в сучасних системах освіти та найпростішим засобом перевірки здобутих знань. Тестування з вивченого матеріалу допомагає запобігти ілюзії компетентності навчання – явищу, яке виникає, коли ученик вчить та повторює теоретичний матеріал без перевірки засвоєння матеріалу або його відтворення.

Файл тестування повинен містити наступну інформацію: назва тесту, порогове значення, питання та варіанти відповідей до них. Порогове значення – це цілочислове значення, мінімальна кількість балів, яку потрібно набрати, щоб визначити тест як пройдений.

Кожне питання обов’язково повинно мати мінімум два варіанти відповіді, хоча б один з варіантів має бути вірним. З метою захисту даних вся інформація повинна зберігатися у файлі в зашифрованому виді. Для шифрування да дешифрування даних використовується алгоритм Rijndael, який буде описано далі у розділі.

Дешифрування даних виконується під час обробки файлу тестування. Дешифровані дані не зберігаються у файл, користувач може вивчати їх тільки під час роботи системи.

У наступному алгоритмі описано процес обробки відповідей користувача на питання тесту. Блок-схему алгоритму тестування розташовано у Додатку Б.

* Встановлюємо значення поля Points, яке зберігає кількість набраних у тесті балів, на 0. Створюємо змінну цілочислового типу, що дозволяє значення null, isAllHaveAnswer для контролю кількості відповідей на питання та встановлюємо значення null.
* Перебираємо у циклі всі питання тесту.
* Створюємо змінну-лічильник цілочислового типу та встановлюємо у значення 0.
* Перебираємо у циклі усі відповіді на поточне питання.
* Якщо відповідь на питання співпадає з очікуваним (правильним) значенням, збільшуємо лічильник, інакше – дій не виконуємо.
* Перевіряємо, чи вибрав користувач поточний варіант відповіді: якщо варіант не обрано, збільшуємо значення змінної isAllHaveAnswers, інакше – дій не виконуємо.
* По закінченні циклу по варіантам відповіді на поточне питання перевіряємо кількість та якість відповідей користувача. Якщо значення змінної isAllHaveAnswer співпадає з кількістю варіантів відповіді на поточне питання, показати повідомлення про те, що користувач не відповів на усі питання. Переходимо до кроку 9.   
  Інакше – перевірити правильність відповіді. Якщо значення лічильника співпадає з кількістю варіантів відповіді на поточне питання, збільшуємо значення поля Points. Інакше – дій не проводимо.
* По завершенні циклу по питанням підраховуємо зароблені бали. Якщо кількість балів більше або дорівнює пороговому значенню, повідомляємо користувачу, що він пройшов тестування та успішно вивчив тему; встановлюємо, що дану тему вивчено. Інакше – повідомляємо, що тест не пройдено.
* Завершити тестування.

# Криптографічний алгоритм шифрування Rijndael

Rijndael (англ. Рендал) – це симетричний алгоритм блочного шифрування, прийнятий в якості стандарту шифрування урядом США за результатами конкурсу AES. Цей алгоритм добре проаналізовано та зараз широко використовується. Даний алгоритм часто називають просто AES (абр. Advanced Encryption Standard).

Блок – це сукупність біт фіксованої довжини. Таким чином, рядок даних розбивається на блоки, після чого блоки шифруються. Даний алгоритм використовує блоки та ключі розміру 128, 192 або 256 біт. У цьому полягає різниця між безпосередньо алгоритмом Rijndael та стандартом AES: останній дозволяє шифрувати тільки 128-бітні блоки.

Даний алгоритм обрано за надійність і те, що він добре справляться з даними великих розмірів (на відміну, наприклад, від алгоритму RSA). Ключем шифрування є назва тесту, яка зберігається у файлі в незашифрованому вигляді.

Детальний опис алгоритму не наводиться через те, що планується використовувати існуючий інструмент.

# Алгоритм завантаження та обробки навчальних матеріалів

Навчальні матеріали розташовано на файловому сервері. Усі матеріали, потрібні для вивчення у поточному семестрі, зберігаються в архіві формату zip.

Файл архіву завантажується з сервера за протоколом FTP.

FTP (англ. File Transfer Protocol – протокол передачі файлів) — це протокол, що надає абоненту можливість обмінюватися файлами з будь-яким комп'ютером мережі, що підтримує протокол FTP. Установивши зв'язок з віддаленим комп'ютером або сервером, користувач може скопіювати файл з нього на свій, та навпаки.

Алгоритм завантаження та обробки архіву з навчальними матеріалами описано далі. Блок-схему алгоритму розташовано у Додатку В.

1. Перевірити наявність навчальних матеріалів на локальному диску. Якщо директорію із номером поточного семестру не знайдено, перейти до кроку 2, інакше – до кроку 10.
2. Зчитати з файлу University.xml дані про адресу сервера, пароль та логін.
3. Задати формат архіву з навчальними матеріалами, його розташування на віддаленому сервері та адресу на локальному диску, куди буде завантажено архів. Адреса на сервері, за якою розташовується файл, має наступний формат:

ftp://адреса\_сервера/код\_підрозділу/код \_спеціалності/семестр

Де:

* адреса\_сервера – адреса віддаленого сервера;
* код\_підрозділу – код інституту або факультету, на якому навчається студент; задається автоматично при реєстрації студента у системі;
* код\_спеціальності – код обраної спеціальності; задається автоматично при реєстрації студента у системі;
* семестр – поточний семестр; визначається автоматично при реєстрації студента у системі (за допомогою системного календаря та номеру курсу) або зміні курсу на сторінці UserDataPage.

1. Відправити запит до сервера на пошук файлів. У разі успішного виконання запиту система отримує список файлів, який містить потрібне значення. Інакше – перехід до кроку 10.
2. В отриманому списку файлів знайти перший файл потрібного формату, додати його назву та формат до адреси на сервері. Якщо файл не знайдено – перехід до кроку 10.
3. Відправити запит до сервера на завантаження файлу. У разі успішного виконання запиту перейти до кроку 7, інакше – до кроку 10.
4. Завантажити файл на локальний диск.
5. Розпакувати файл архіву програмно. У разі успішного виконання операції перейти до кроку 9, інакше – до кроку 10.
6. Видалити файл архіву.
7. Завершити операцію.

# Алгоритм завантаження та обробки розкладу

Файл розкладу розташовано на віддаленому сервері. Формат файлу може бути xls або xlst. Файл із розкладом завантажується з сервера за протоколом FTP. Алгоритм завантаження та обробки файлу описано далі. Блок-схему алгоритму розташовано у Додатку Г.

1. Перевірити наявність списку дисциплін на поточний семестр у файлі користувача. Якщо списку немає, перейти до кроку 2, інакше – до кроку 13.
2. Задати список форматів файлів, які треба шукати.
3. Зчитати з файлу University.xml дані про адресу сервера, пароль та логін, задати адресу розташування файлу на сервері.   
   Адреса розташування файлу на сервері має наступний формат:

ftp://адреса\_сервера/код\_підрозділу/

Де:

* адреса\_сервера – адреса віддаленого сервера;
* код\_підрозділу – код інституту або факультету, на якому навчається студент;

1. Відправити запит до сервера на пошук файлів. У разі успішного виконання запиту система отримує список файлів, який містить потрібне значення. Інакше – перехід до кроку 13.
2. В отриманому списку файлів знайти перший файл потрібного формату, додати його назву та формат до адреси на сервері. Якщо файл не знайдено – перехід до кроку 13.
3. Відправити запит до сервера на завантаження файлу. У разі успішного виконання запиту перейти до кроку 7, інакше – до кроку 13.
4. Завантажити файл на локальний диск.
5. Відкрити файл як базу даних. У разі успішного виконання операції перейти до кроку 9, інакше – 13.
6. Знайти у базі даних лист, назва якого співпадає з рядком наступного формату:

«номер\_курсу код\_спеціальності»

Де:

* номер\_курсу – курс, на якому навчається студент; задається при реєстрації студента у системі або при зміні курсу на сторінці UserDataPage.
* код\_спеціальності – код спеціальності; задається автоматично при реєстрації студента у системі.

Між номером курсу та кодом спеціальності ставиться знак пропуску.

У разі вдалого виконання операції перейти до наступного кроку, інакше – до кроку 13.

1. Розібрати лист як таблицю у циклі. Потрібно знайти наступні дані: назва дисципліни, кількість годин на вивчення теоретичного матеріалу, ПІБ викладача. Якщо у курсі вивчення дисципліни не викладається теоретичний матеріал, завершити ітерацію не зберігаючи дані.
2. Закрити базу даних та зберегти отримані дані у файл користувача на локальному диску.
3. Видалити файл бази даних.
4. Завершити операцію.

# Алгоритм обробки персональних даних користувача

Персональні дані користувача зберігаються на локальному диску у файлі settings.xml. Зберігаються наступні дані:

* ім’я та прізвище студента;
* адреса, за якою буде розташовано навчальні матеріали;
* назва та код спеціальності;
* код підрозділу;
* номер курсу;
* номер семестру;
* список дисциплін, які вивчаються у семестрі;
* список тем для кожної дисципліни та відмітки про їх проходження.

Персональні дані користувача задаються, обробляються та зберігаються локально. Дані про підрозділ, спеціальність, курс, адреса локального розташування учбових матеріалі задаються користувачем при першому вході у систему. Номер семестру визначається автоматично, за системним календарем та номером курсу. Список тем та дисциплін задається після обробки файлу з розкладом на семестр та навчальних матеріалів, завантажених з віддаленого сервера.

Дані про курс, адресу локального розташування навчальних матеріалів, прізвище та ім’я користувача можна змінити (або задати) на сторінці UserDataPage.

Наступний алгоритм описує процес обробки персональних даних користувача при першому вході у систему. Блок-схему алгоритму розташовано у Додатку Д.

1. Почати пошук файлу з персональними даними користувача settings.xml на локальному диску. Якщо файл не знайдено, перейти до кроку 2, інакше – до кроку 10.
2. Перейти на сторінку StartPage, отримати від користувача дані про підрозділ (інститут або університет), спеціальність та курс, адресу на локальному диску, куди буде завантажено файли.
3. Занести дані у колекцію елементів типу string.
4. Згенерувати файл settings.xml
5. Занести до файлу колекцію даних.
6. Зберегти файл.
7. Завантажити та обробити розклад дисциплін з сервера. Алгоритм операції завантаження то обробки файлу розкладу описано у пункті 2.1 даного документу.
8. Завантажити та обробити учбові матеріали. Алгоритм операції завантаження то обробки учбових матеріалів описано у пункті 2.2 даного документу.
9. Зберегти змінений файл з персональними даними.
10. Завершити операцію.

# Алгоритм пошуку навчальних матеріалів на локальному диску

Завантажені з віддаленого сервера та оброблені учбові матеріали зберігаються на локальному диску. Розташування кореневої папки додатку, у якій зберігаються файли, визначається користувачем при першому запуску системи. Також користувач має можливість програмно змінити розташування папки з файлами.

Структуру кореневої папки системи, у якій зберігаються навчальні матеріали, схематично зображено на Рис. 2.1:

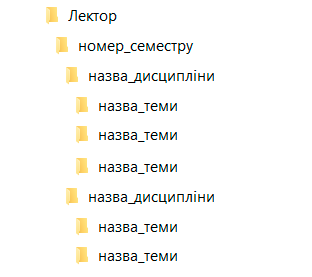


Рис. 2.1. Схематичне зображення структури кореневої папки

Де:

* номер\_семестру – семестр;
* назва\_дисципліни – назва дисципліни, яка вивчається у даному семестрі;
* назва\_теми – назва теми, яка вивчається у рамках дисципліни

У наступному алгоритмі описано пошук файлу, розташованого у папці назва\_теми. В даній папці знаходяться файли із лекційним матеріалом та тестом за темою, кожен файл – у єдиному екземплярі. Для пошуку файлів обох форматів використовується єдиний алгоритм, який описано нижче. Блок-схему алгоритму розташовано у Додатку Ж.

1. Визначити адресу розташування папки з навчальними матеріалами, номер поточного семестру, назву дисципліни та теми, яку обрав користувач.
2. Сформувати адресу розташування файлу, який потрібно завантажити з локального диску, використовуючи отримані дані. Адреса буде мати наступний формат:

кор\_папка//ном\_семестру//назва\_дисципліни//назва\_теми//

Де:

* кор\_папка – розташування кореневої папки додатку, у якому зберігаються файли (папка обов’язково має назву «Лектор»);
* ном\_семестру – номер поточного семестру;
* назва\_дисципліни – назва обраної користувачем дисципліни;
* назва\_теми – назва обраної користувачем теми, яка вивчається в рамках дисципліни.

1. Задати формат файлу, який потрібно завантажити: для лекційного матеріалу – pdf, для тесту – xml.
2. Отримати список усіх файлів, які знаходяться за вказаною адресою. У разі успішного виконання операції перейти до кроку 5, інакше – до кроку 8.
3. Знайти серед списку файлів перший, формат якого відповідає вказаному на кроці 3 значенню, запам’ятати його назву. У разі успішного виконання операції, утворити повне ім’я файлу, яке містить адресу розташування на локальному диску (сформовану на кроку 2), назву та формат. Інакше – перейти до кроку 8.
4. Завантажити файл у систему.
5. Обробити файл чинним образом та відобразити.
6. Завершити операцію.

# Висновки

В даному розділі було наведено найбільш важливі алгоритми, які буде реалізовано у системі «Лектор». Для двох описаних алгоритмів побудовано блок-схеми.Також у розділі було наведено загальну інформацію про криптографічний алгоритм Rijndael, який буде використовуватися для дешифрування даних з файлу тестування.

Наступним кроком після алгоритмічного конструювання системи є вибір інструментів та програмне конструювання системи.

# РОЗДІЛ 3

# Програмне конструювання системи навчального призначення

У даному розділі буде розглянуто основні засоби, які було використано для створення системи навчального призначення «Лектор».

# Обґрунтування вибору засобів розробки

# Обґрунтування вибору мови програмування

C# – об'єктно-орієнтована мова програмування, розроблена у 1998-2001 роках групою інженерів під керівництвом Андерса Хейлсберга в компанії Microsoft як мова розробки додатків для платформи .NET Framework, та згодом була стандартизована як ECMA-334 і ISO/IEC 23270.

C# відноситься до сімейства мов із C-подібним синтаксисом. Мова має явну статичну сильну типізацію, підтримує поліморфізм, перевантаження операторів, делегати, атрибути, події, властивості, узагальнені типи та методи, ітератори, анонімні функції з підтримкою замикань, інтегровану мову запитів LINQ, виключення, коментарі в форматі XML.

Цю мову програмування було обрано для розробки системи через те, що вона є одним із найбільш потужних та популярних інструментів розробки настільних додатків, має велику кількість безкоштовних стандартних та сторонніх бібліотек, а також добре взаємодіє із сторонніми продуктами (наприклад, Adobe Acrobat Reader v11). Таке розмаїття інструментів дозволяє сильніше концентруватися на продукті, який виготовляється, а не на деталях розробки окремих компонентів. І як наслідок – значно прискорюється процес розробки, створюються більш якісні програмні засоби.

# Обґрунтування вибору середовища розробки

Через те, що мовою розробки обрано C#, найбільш раціональним вибором середовища розробки є Visual Studio 2015, створеного та підтримуваного компанією-розробником обраної мови програмування.

Microsoft Visual Studio – це лінійка продуктів компанії Microsoft, що включають інтегроване середовище розробки програмного забезпечення та ряд інших інструментальних засобів. Дані продукти дозволяють розробляти консольні додатки, додатки з графічним інтерфейсом (Windows Forms, WPF); веб-сайти, веб-додатки, веб-служби за допомогою технології ASP MVC; додатки для платформ Windows 8/8.1 та Windows Universal Platform, у тому числі додатки для мобільних ОС та ігрових консолей; засоби для Microsoft Office, ігри та багато іншого.

Visual Studio включає в себе редактор вихідного коду з підтримкою технології IntelliSense, можливістю рефакторінгу коду. Вбудований відладчик може працювати як відладчик рівня вихідного коду, так і в якості відладчика машинного рівня. Решта вбудованих інструментів включають редактор, веб-редактор, дизайнер класів і дизайнер схеми бази даних. Visual Studio дозволяє створювати і підключати сторонні додатки для розширення функціональності практично на кожному рівні розробки, включаючи додавання підтримки систем контролю версій вихідного коду (наприклад, Team Foundation Server або Git), додавання нових наборів інструментів (наприклад, для редагування і візуального проектування коду на об’єктно-орієнтованих мовах програмування) або інструментів для інших аспектів процесу розробки програмного забезпечення. Ще однією перевагою Visual Studio є можливість швидкої розробки модульних тестів.

Visual Studio 2015 є крайньою версією продукту на момент розробки системи. Середовище було представлено 20 червня 2015 року, під кодовим ім'ям Dev14. Суттєвою зміною стала підтримка багатьох цільових платформ: окрім базової Windows з'явилась можливість створювати проекти для IOS та Android. Для розробників комп'ютерних ігор була додана підтримка фреймворку Unity. Оновлено і механізм автентифікації: під час запуску Visual Studio користувач синхронізується з єдиним аккаунтом Microsoft. Версія включає в себе .NET Framework 4.6 та підтримку універсальної платформи Windows 10.

Крайнім оновленням на момент розробки системи є Update 2 від 30 березня 2016-го року, у якому багато уваги приділено стабільності, та продовжено роботу у напрямку підтримки нових стандартів мови С++.

# Обґрунтування вибору форматів файлів

Система навчального призначення «Лектор» працює із п’ятьма форматами файлів. Опис обраних форматів, способу використання та інструментів обробки розташовано у Таблиці 3.1.

Таблиця 3.1.

|  |  |  |
| --- | --- | --- |
| Формат | Для чого використовується | Інструмент обробки |
| xml | Файл даних користувача; файли тестування | Класи просторів імен  System.Xml та  System.Xml.Linq |
| pdf | Лекційний матеріал | Adobe Reader v11 |
| xls, xlst | Розклад занять на семестр | Класи простору імен System.Data.OleDb |
| zip | Архів файлів, завантажуваних із сервера | Клас ZipFile простору імен System.IO.Compression |

Xml (англ. Extensible Markup Language – роширювана мова розмітки) – це простий та гнучкий текстовий формат. Мова розмітки була розроблена комітетом Консорціуму Всесвітньої павутини на чолі з Джоном Босаком. Її основне призначення — спростити мову SGML, фокусуючись на певному завданні — документах Інтернету. XML лишається метамовою, яка дозволяє користувачам створювати будь-які потрібні теги (впровадження розширюваності), а потім описувати ці теги та їх дозволене використання.

Zip – це формат архівації файлів і стиснення даних без втрат, створений в. 1989 році Філом Кацем і реалізований в програмі PKZIP компанії PKWARE, Inc. в якості заміни формату архівів. Формат ZIP підтримується багатьма програмами, в тому числі операційними системами Microsoft Windows (з 1998 року) і Apple Mac OS X (з версії 10.3). Багато вільних операційні системи також мають вбудовану підтримку ZIP-архівів. Найбільш часто в ZIP використовується алгоритм стиснення Deflate.

Формати xml та zip було обрано через те, що .NET Framework має зручні інтегровані інструменти для роботи з ними. Детальніше про засоби обробки описано у пунктах 3.2.11 та 3.2.11 даного документу.

Розклад занять – це невелика база даних, яка складається із таблиць (листів). У кожній таблиці зберігаються дані про дисципліни, які вивчаються у поточному семестрі для обраної групи, викладачів, кількість годин для кожного виду занять, виду контролю, тижнів проведення занять та ін. Для розміщення розкладу занять було обрано файли форматів xls та xlst, адже вони є найбільш зручними та популярними для створення, обробки та зберігання невеликих баз даних.

PDF (абревіатура від англ. Portable Document Format – портативний формат документів) – це розроблений компанією Adobe Systems з використанням низки можливостей мови PostScript незалежний від платформи формат подання в електронному вигляді поліграфічної продукції, різної електронної документації та презентацій.

Для перегляду можна використовувати офіційну безкоштовну програму Adobe Acrobat Reader (далі – Adobe Reader), а також програми сторонніх розробників.

У процесі розробки ПЗ виникла необхідність у використанні стороннього додатку для перегляду файлу обраного формату, адже основна мета системи навчального призначення «Лектор» – це створення зручного інструменту для вивчення теоретичного матеріалу, а не для перегляду файлів формату PDF. Найбільш зручним з економічної та функціональної точок зору інструментом є програма Adobe Reader, яку і було використано для перегляду файлів з лекційним матеріалом. Через те, що для роботи системи потребується специфічна версія Adobe Reader (а саме – Adobe Reader v11), дистрибутив програми включено до набору файлів, необхідних для роботи розроблюваної системи. Таким чином користувач може працювати із системою, навіть якщо він не встановив Adobe Reader потрібної версії наперед.

# Основні модулі програмного засобу

# Діаграма класів

В процесі розробки ПС, було побудовано UML діаграму основних класів, яку зображено на Рис. А.1 Додатку А.

Класи, назви яких закінчуються на слово «Page», є нащадками класу Page, що інкапсулює сторінку з розміщеними на ній елементами. Сторінка знаходиться у навігаційному вікні.

Діаграму послідовності дій, а саме переміщення користувача (на діаграмі – «Студент») між класами-нащадками класу Page, зображено на Рис. А.2. Додатку А. Діаграму кооперації зображено на Рис. А.4. Додатку А.

Класи, призначені для обробки файлів та даних, а також структурованого зберігання даних в оперативній пам’яті, розташовано у бібліотеці Lecturer.Data. Класи, призначені для обробки даних, знаходяться у просторі імен Lecturer.Data.Processor, класи для зберігання даних – у Lecturer.Data.Entities.

В окрему бібліотеку винесено елемент керування WpfAcrobatCtrl, який використовується для відображення файлів формату PDF.

Далі у розділі описано класи та бібліотеки класів ПЗ.

# Клас MainWindow

Клас MainWindow являє собою головне вікно додатку, на якому розташовано сторінки із елементами управління. Він є нащадком класу Window простору імен System.Windows, який надає можливість створювати, налаштовувати, показувати та керувати часом існування вікон та діалогових вікон.

У Таблиці 3.2. наведено методи, реалізовані безпосередньо у класі MainWindow.

Таблиця 3.2.

|  |  |  |  |
| --- | --- | --- | --- |
| Ім’я метода | Вхідні параметри та їх опис | Опис метода | Значення, яке повертає метод |
| TryLoadUserData | - | Визначити, чи можна завантажити файли та дані | Флаг завантаження |
| mainFrame\_ContentRendered | Об’єкт-відправник, подія | Вимкнути стандартний спосіб навігації між сторінками | - |

# Клас StartPage

Клас StartPage є нащадком класу Page та являє собою сторінку додатку, призначену для введення даних користувача, а саме: назва підрозділу, спеціальність, курс, семестр, адресу розміщення папки з учбовими матеріалами на локальному диску. На дану сторінку користувач переводиться автоматично, якщо не вдалося знайти файл даними.

Зовнішній вигляд сторінки StartPage наведено на Рис. 3.1. У Таблиці 3.3. наведено основні методи класу StartPage:

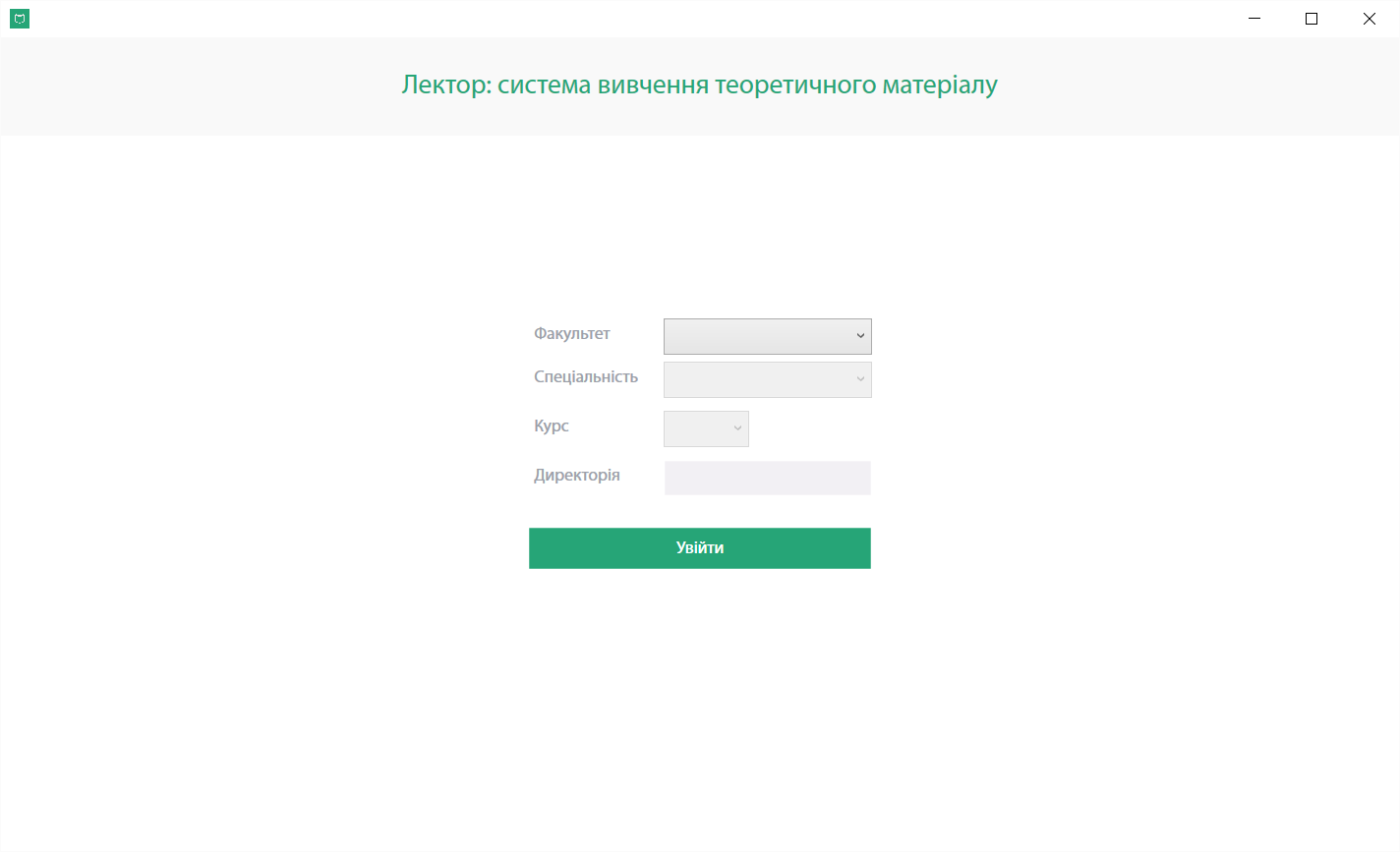


Рис. 3.1. Сторінка StartPage.

Таблиця 3.3.

|  |  |  |  |
| --- | --- | --- | --- |
| Ім’я метода | Вхідні параметри та їх опис | Опис метода | Значення, яке повертає метод |
| Done\_Click | Об’єкт-відправник, подія | Слухач кнопки Done, яка обробляє введені дані | - |
| Folder\_Focus | Об’єкт-відправник, подія | Слухач текстового поля, при наведенні на яке з’являється діалогове вікно для вибору папки | - |
| comboIns\_SelectionChanged | Об’єкт-відправник, подія | Слухач випадаючого списку; при зміні вибраного елементу змінює значення та видимість інших | - |
| comboSpec\_SelectionChanged | Об’єкт-відправник, подія | Слухач випадаючого списку; при зміні вибраного елементу змінює значення та видимість інших | - |
| comboCource\_SelectionChanged | Об’єкт-відправник, подія | Слухач випадаючого списку; при зміні вибраного елементу змінює значення та видимість інших | - |
| GetUniversityList | - | Отримує список підрозділів з файлу University.xml | - |
| ProcessUserFile | - | Підготовка введених даних до запису у файл settings.xml | - |

# Клас CourcePage

Клас CourcePage є нащадком класу Page та являє собою сторінку додатку, призначену для відображення списку дисциплін, кількості годин та імен викладачів. Також на даній сторінці знаходиться кнопка для переходу на сторінку UserDataPage, опис якої розташовано у пункті 3.8 даного документу.

Зовнішній вигляд сторінки CourcePage наведено на Рис. 3.2. У Таблиці 3.4 наведено основні методи класу CourcePage:

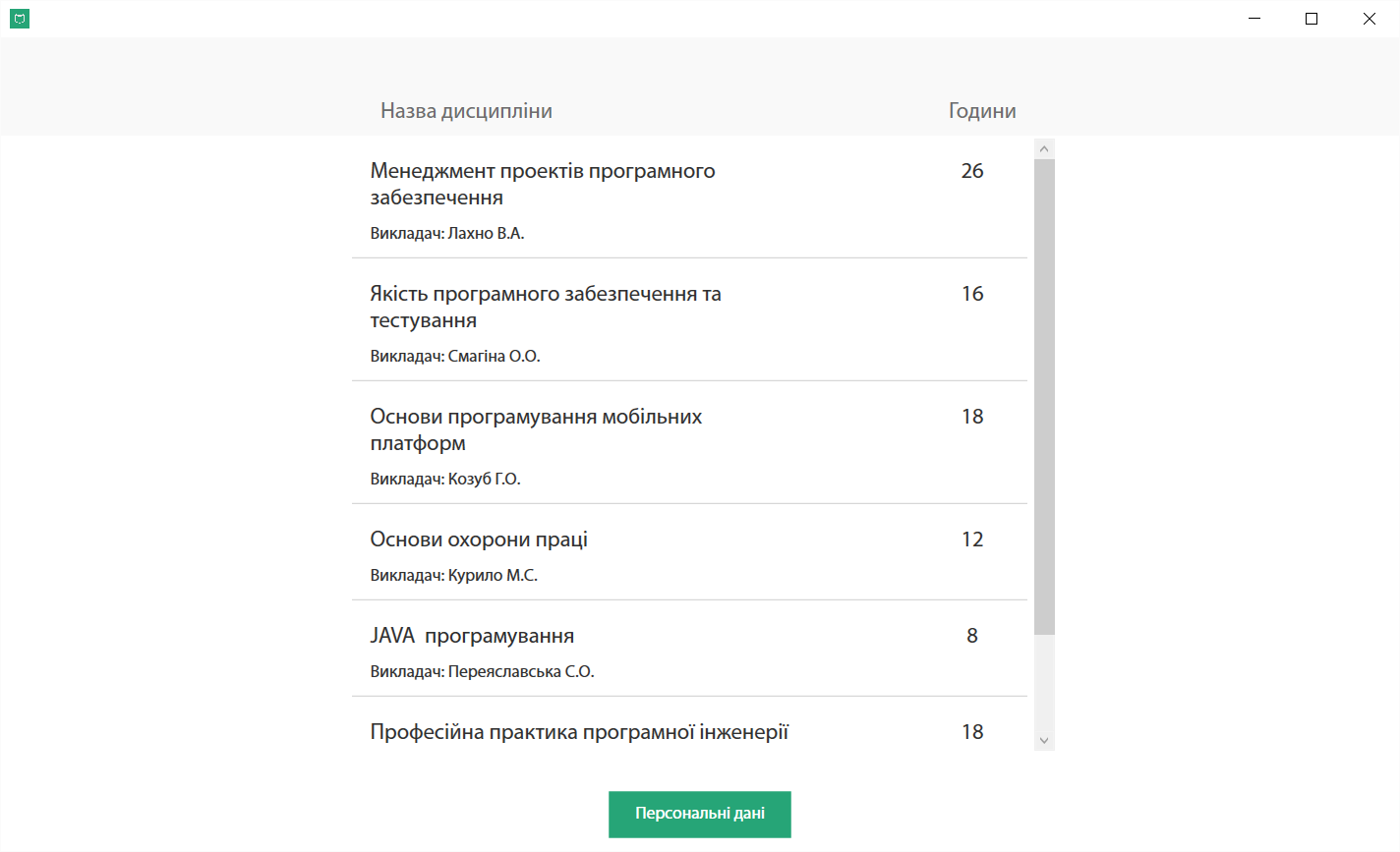


Рис. 3.2. Сторінка CourcePage

Таблиця 3.4.

|  |  |  |  |
| --- | --- | --- | --- |
| Ім’я метода | Вхідні параметри та їх опис | Опис метода | Значення, яке повертає метод |
| myList\_SelectionChanged | Об’єкт-відправник, подія | Слухач натискання на елемент керування ListView; перехід на сторінку SubjectPage, заповнену відповідними даними (якщо існують) | - |
| Button\_Click | Об’єкт-відправник, подія | Слухач натискання на кнопку; перехід на сторінку UserDataPage | - |
| PrepareData | - | Підготовка даних про дисципліни до виведення на екран | - |

Таблиця 3.4. Основні методи класу CourcePage

# Клас SubjectPage

Клас SubjectPage є нащадком класу Page та являє собою сторінку додатку, призначену для виводу даних про теми, які вивчаються у рамках обраної користувачем дисципліни, а саме: назви теми, дані про її вивчення та доступність.

Зовнішній вигляд сторінки SubjectPage наведено на Рис. 3.3. У Таблиці 3.5 наведено основні методи класу SubjectPage.

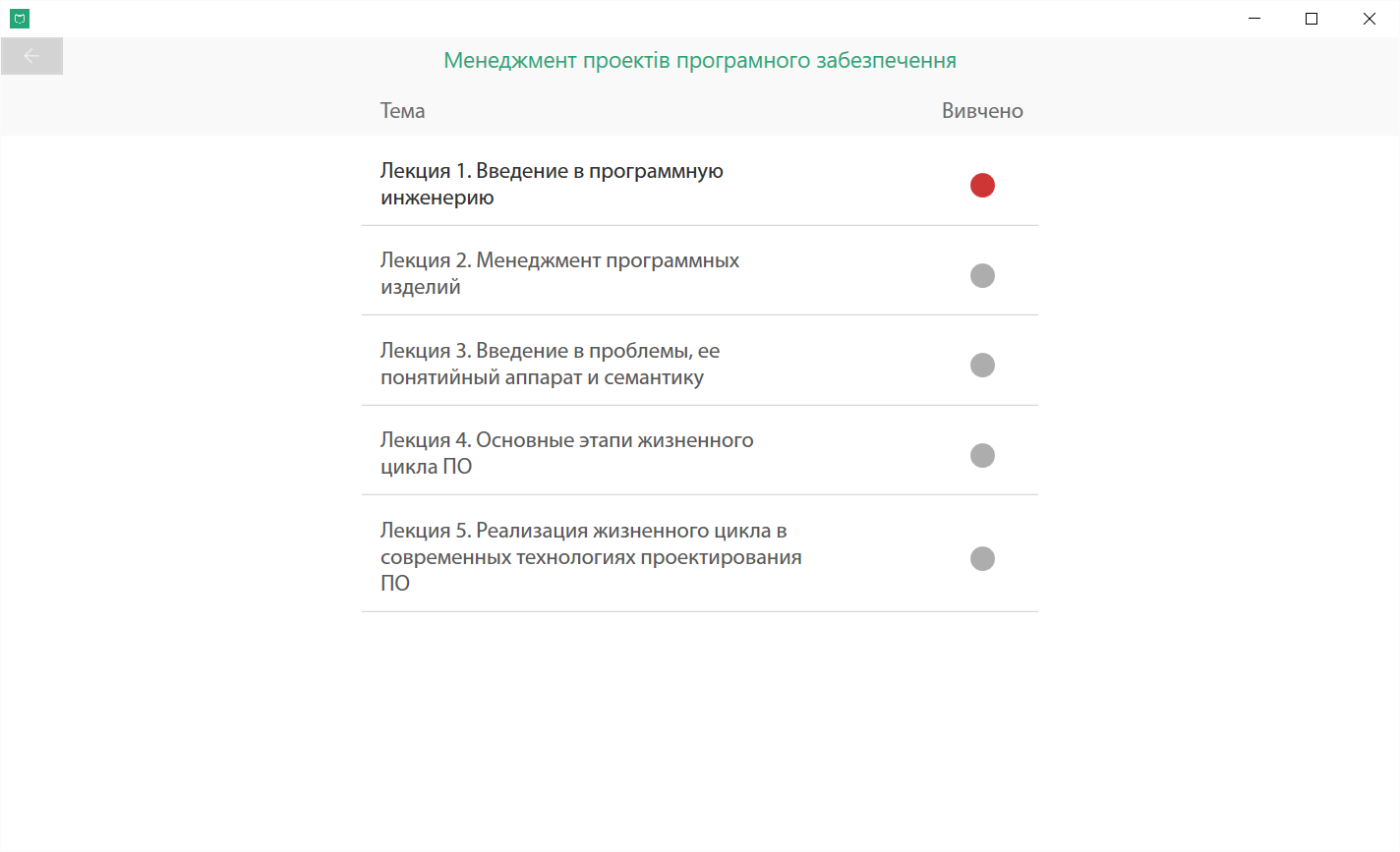


Рис. 3.3. Сторінка SubjectPage

Таблиця 3.5.

|  |  |  |  |
| --- | --- | --- | --- |
| Ім’я метода | Вхідні параметри та їх опис | Опис метода | Значення, яке повертає метод |
| myList\_SelectionChanged | Об’єкт-відправник, подія | Слухач натискання на елемент керування ListView; відповідає за переведення на сторінку LectionPage, якщо існує файл | - |
| InfoMessage | Текст повідомлення, заголовок | Показ повідомлення із заданими параметрами | - |
| btnLink\_Click | Об’єкт-відправник, подія | Слухач натискання на кнопку; перехід на сторінку CoursePage | - |
| CheckAcrobatInstallation | - | Перевірити, чи встановлено Acrobat Reader v11 | Флаг результату перевірки |
| ShowInstallationMessage | - | Показ повідомлення про те, що потрібну програму не встановлено | - |

# Клас LectionPage

Клас LectionPage є нащадком класу Page та являє собою сторінку додатку, призначену для виведення на екран файлу із лекцією. Для демонстрації файлу використовується елемент керування WpfAcrobatCtrl бібліотеки класів Lecturer.PDFControl, яку описано у пункті 3.2.9 даного документу.

Окрім конструктора без параметрів, клас містить закритий метод Next\_Click, параметрами якого є об’єкт-відправник та подія. Даний метод є слухачем натискання на кнопку Next та відповідає за переведення користувача на сторінку тестування TestPage або внесення даних про вивчення теми у файл користувача.

Зовнішній вигляд сторінки LectionPage наведено на Рис. 3.4.

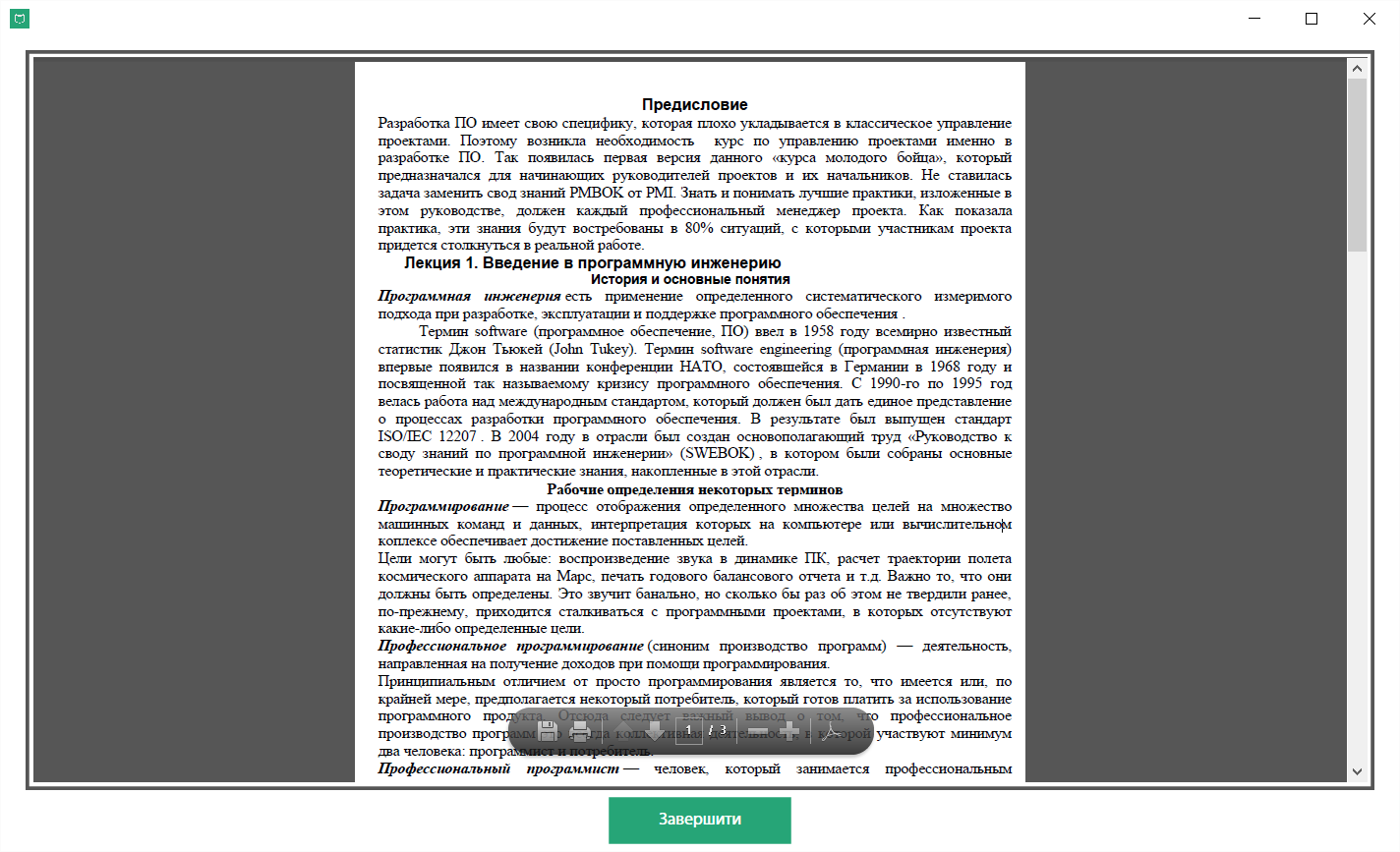


Рис. 3.4. Сторінка LectionPage

# Клас TestPage

Клас TestPage є нащадком класу Page та являє собою сторінку додатку, призначену для тестування користувача з вивченої теми.

Зовнішній вигляд сторінки TestPage зображено на Рис. 3.5. У Таблиці 3.6 наведено основні методи класу TestPage.

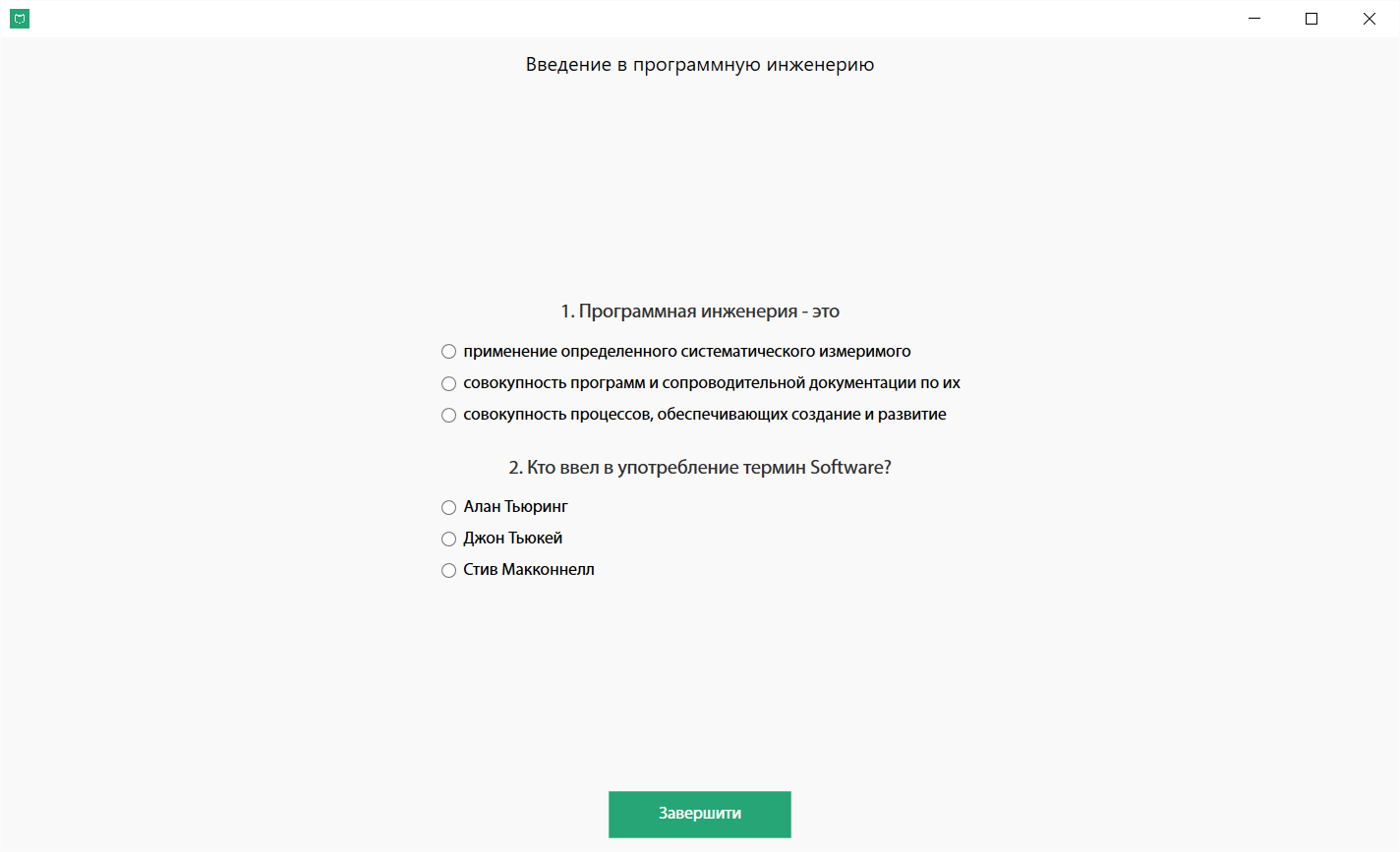


Рис. 3.5. Сторінка TestPage

Таблиця 3.6.

|  |  |  |  |
| --- | --- | --- | --- |
| Ім’я метода | Вхідні параметри та їх опис | Опис метода | Значення, яке повертає метод |
| btnDone\_Click | Об’єкт-відправник, подія | Слухач натискання на | - |
| PrepareTest | - | Підготовка тесту | - |
| CreateRadioButton | Текст надпису, тег | Створення елементу керування для тесту | Об’єкт RadioButton |
| CreateCheckBox | Текст надпису, тег | Створення елементу керування для тесту | Об’єкт CheckBox |

# Клас UserDataPage

Клас UserDataPage є нащадком класу Page та являє собою сторінку додатку, призначену для огляду та змінення даних користувача.

Зовнішній вигляд сторінки UserDataPage зображено на Рис. 3.6. У Таблиці 3.7 наведено основні методи класу UserDataPage.

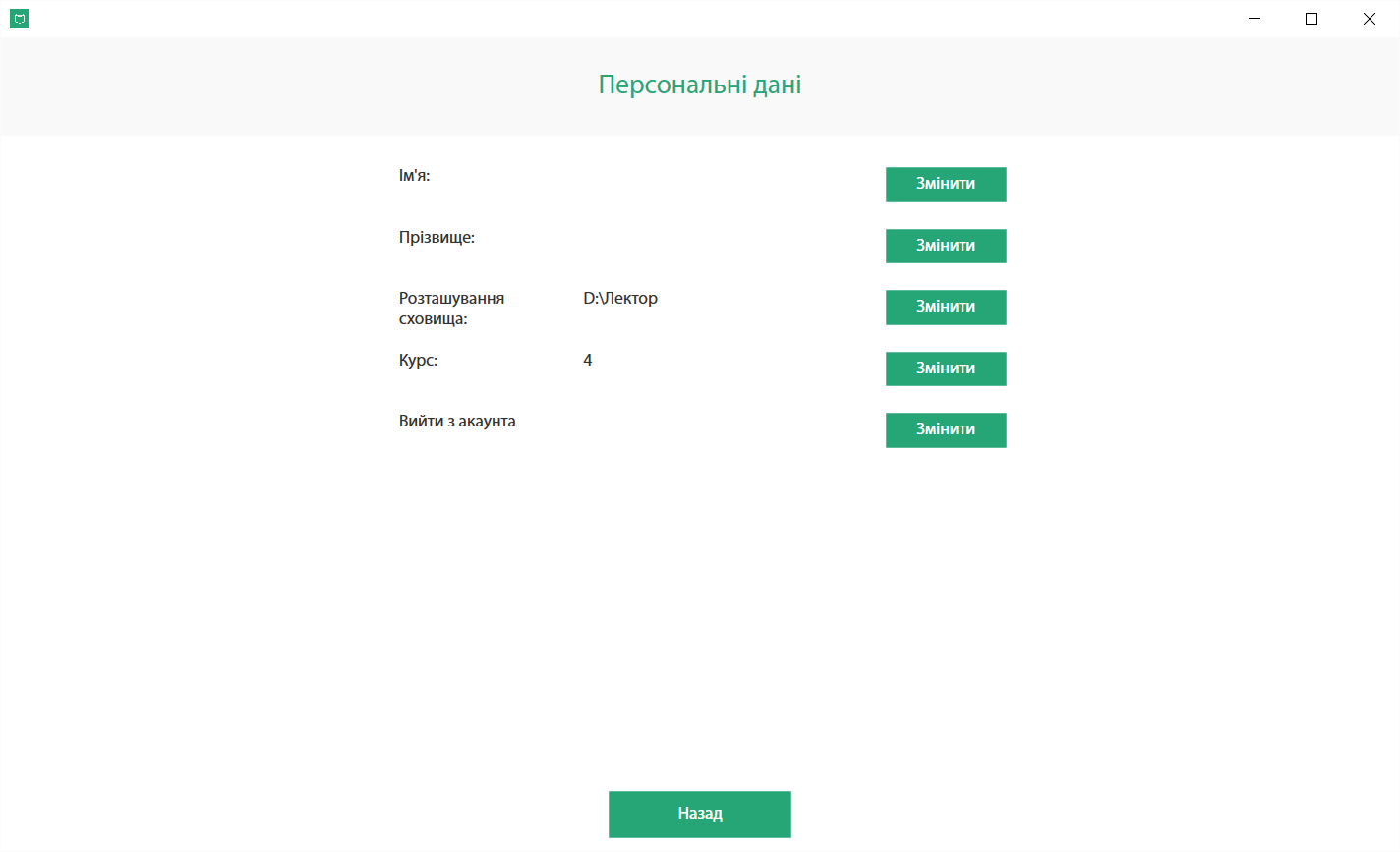


Рис. 3.6. Сторінка UserDataPage

Таблиця 3.7.

|  |  |  |  |
| --- | --- | --- | --- |
| Ім’я метода | Вхідні параметри та їх опис | Опис метода | Значення, яке повертає метод |
| Button\_Click | Об’єкт-відправник, подія | Обробка натискання на кнопку; | - |
| btnOnListView\_Click | Об’єкт-відправник, подія | Обробка натискання на елемент на ListView; змінення даних | - |
| LoadFilesFromServer | Номер курсу | Завантаження даних з сервера | - |
| WriteDataIntoFile | Змінене значення | Запис даних у файл | - |
| CopyDirectory | Адреса знаходження папки, адреса копіювання | Копіювання папки | - |

# Бібліотека класів Lecturer.PDFControl

Бібліотека класів Lecturer.PDFControl містить два класи – PdfControl та WpfAcrobatCtrl. Клас WpfAcrobatCtrl – це «огортка» до другого класу бібліотеки та є елементом керування додатку, побудованого за системою WPF, для відображення файлу формату PDF.

Клас-огортка – це об'єктний тип даних, покликаний зберігати значення об'єктного типу. Клас-огортка проектується за шаблоном «Адаптер».

Адаптер (англ. Adapter) – структурний шаблон проектування, призначений для організації використання функцій об'єкта, недоступного для модифікації, через спеціально створений інтерфейс.

В даному випадку використовується для адаптації елементу керування PdfControl, який у свою чергу є «огорткою» до програми, яка дозволяє працювати із файлом формату PDF. Цей елемент керування призначено для використання у додатках Windows Forms. Через те, що із елементом керування не можна працювати напряму із додатку WPF, і було створено подвійну огортку.

Програмою, яка використовується для роботи із файлом PDF-формату, є Adobe Acrobat Reader v11. Опис програми наведено у пункті 3.1.3. даного документу.

Схематичне зображення елементу керування WpfAcrobatCtrl зображено на Рис. 3.7:

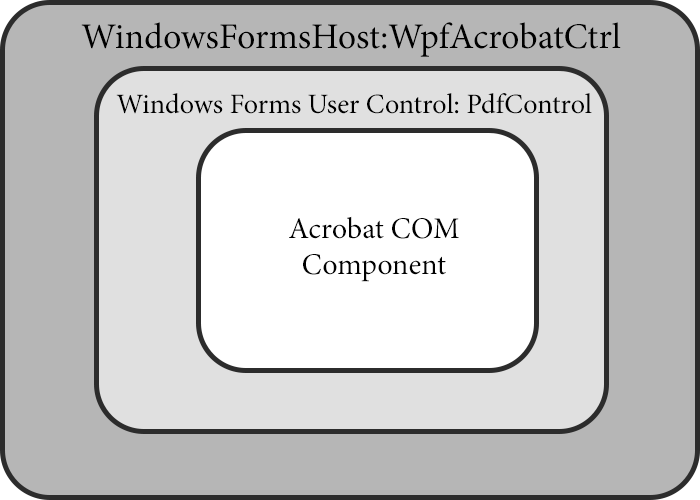


Рис 3.7. Схематичне зображення внутрішньої організації елементу керування WpfAcrobatCtrl.

У таблиці 3.8. наведено опис методів, реалізованих у класі WpfAcrobatCtrl.

Таблиця 3.8.

|  |  |  |  |
| --- | --- | --- | --- |
| Ім’я метода | Вхідні параметри та їх опис | Опис метода | Значення, яке повертає метод |
| FilePathChanged | Об’єкт-відправник, подія | Обробка події змінення адреси файлу | - |
| FilePathChanged | Адреса файлу, який використовується; адреса нового файлу | Обробка події змінення адреси файлу | - |

# Бібліотека класів Lecturer.Data

Бібліотека класів Lecturer.Data надає засоби для завантаження та обробки файлів з віддаленого сервера та локального диску, роботи з даними користувача, обробки файлів тестування, розкладу дисциплін; структурованого зберігання даних в оперативній пам’яті.

Далі описано класи, які містяться в бібліотеці.

# Клас StorageProcessor

Клас StorageProcessor містить методи для роботи із файлами на локальному диску та віддаленому сервері (окрім файлів формату xml). Клас не є статичним, однак усі методи класу є статичними.

У Таблиці 3.9 наведено основні методи класу StartPage:

Таблиця 3.9.

|  |  |  |  |
| --- | --- | --- | --- |
| Ім’я метода | Вхідні параметри та їх опис | Опис метода | Значення, яке повертає метод |
| ReplaceCharacters | Адреса папки; флаг | Заміна некоректних символів | Рядок із заміненими символами |
| ListDirectoriesOnServer | Адреса папки на віддаленому сервері, логін та пароль | Підготування потоку до пошуку папки | потік |
| LoadFileFromPath | Адреса папки на віддаленому сервері, логін та пароль | Підготування потоку до завантаження файлу | потік |
| GetSemesterFilesAsync | - | Асинхронне завантаження файлів з сервера | Задача |
| TryGetFileByFTP | Адреса файлу на сервері; адреса завантаження; список форматів | Завантаження файлу | Путь до завантаженого файлу |
| ProcessZipFile | Путь до файла; путь для розпакування | Обробка zip-архіву | Флаг розпакування |
| CreateDirectory | Папка, у якій буде створено нову; назва створюваної папки | Створення директорії | Путь до створеної папки |
| ProcessSchedule | Код підрозділу | Обробка файлу із розкладом | - |
| GetFilePath | Путь до папки, формат файлу | Пошук файлу заданого формату у заданій папці | Путь до папки |
| GetFolderNames | Путь до директорії дисципліни | Отримує список тем | Список дисциплін |

Даний клас не містить засобів для обробки файлу розкладу занять та використовує для цього клас ExcelFileProcessor

# Клас ExcelFileProcessor

Даний клас містить методи для підключення до файлу бази даних та обробки інформації. Для роботи з фалами форматів xls та xlst використовуються класи простору імен System.Data.OleDb.

У Таблиці 3.10 наведено опис методів, реалізованих у класі.

Таблиця 3.10.

|  |  |  |  |
| --- | --- | --- | --- |
| Ім’я метода | Вхідні параметри та їх опис | Опис метода | Значення, яке повертає метод |
| GetConnectionString | Адреса файлу, заголовок | Отримання рядка підключення до бази даних | Рядок підключення до файлу бази даних |
| GetDataSet | - | Отримання таблиці, у якій знаходиться інформація про курс дисциплін для потрібної групи | Таблиця (об’єкт типу DataSet) |
| FillSource | - | Отримання списку дисциплін з таблиці | Список дисциплін (колекція елементів List<Subject>) |

# Клас XMLProcessor

Клас XMLProcessor відповідає за роботу із файлами формату xml. Клас містить методи для обробки файлу користувача та файлів тестування.

У Таблиці 3.11 наведено основні методи класу StartPage.

Таблиця 3.11.

|  |  |  |  |
| --- | --- | --- | --- |
| Ім’я метода | Вхідні параметри та їх опис | Опис метода | Значення, яке повертає метод |
| GetSubjectList | - | Отримання списку дисциплін | Список дисциплін |
| GetTopicList | Назва дисципліни | Пошук списку тем для заданої дисципліни | Список тем |
| ReadQuizFile | - | Читання та дешифрування файлу тестування | Екземпляр класу Quiz (дані для тестування) |
| PrepareUserData | Посилання на список даних із файлу користувача | Читання файлу із даними користувача | - |
| SaveDocument | - | Збереження файлу із даними користувача | - |
| CreateSettingsFile | Список даних користувача для запису у файл | Створення файлу із даними користувача | ­- |
| SetTopicStudied | - | Позначити, що тему вивчено | - |
| WriteSemester | - | Заповнення файлу з особистими даними | - |
| FillTopicList | Екземпляр класу Subject | Збереження списку тем до файлу | - |
| GenerateElement | Потік, назва тегу, вміст | Створення тегу | - |
| GenerateAttribute | Потік, назва атрибуту, назва тегу | Створення атрибуту | - |

Даний клас не має засобів для дешифрування файлів тестування, та використовує для цього засоби класу CryptoProcessor.

# Клас CryptoProcessor

Даний клас реалізує методи для шифрування та дешифрування даних за алгоритмом AES. У класі не реалізовано вказаний алгоритм шифрування – для цього використовуються класи простору імен System.Security.Cryptography, в особливості – клас Rijndael.

Опис методів, реалізованих у класі, наведено у Таблиці 3.12.

Таблиця 3.12.

|  |  |  |  |
| --- | --- | --- | --- |
| Ім’я метода | Вхідні параметри та їх опис | Опис метода | Значення, яке повертає метод |
| Encrypt | Дані, які потрібно зашифрувати; ключ шифрування | Шифрування даних | Рядок зашифрованих даних |
| Decrypt | Дані, які потрібно дешифрувати; ключ шифрування | Дешифрування даних | Рядок дешифрованих даних |
| ToString | масив байт | Перетворення типу | Рядок даних |
| ToByteArray | Рядок даниз | Перетворення типу | Масив байт |

# Класи-сутності

Класи-сутності – це класи, призначені для структурованого зберігання даних в оперативній пам’яті. До класів-сутностей відносяться усі класи, розташовані в просторі імен Lecturer.Data.Entities. Всі класи, окрім Course, містять тільки відкриті нестатичні поля.

# Клас Cource

Клас Course зберігає дані про курс дисциплін, які вивчає користувач. Він реалізує шаблон проектування Singleton.

Singleton (англ. Одинак) – це породжуючий шаблон проектування, який гарантує, що в однопотоковому додатку буде єдиний екземпляр класу з глобальною точкою доступу. Архітектура шаблону Singleton має такі важливі властивості:

* змінна доступна завжди, час її існування – від запуску програми до її завершення.
* надає глобальний доступ, тобто, може бути доступна з будь-якої частини програми.

Клас має статичне поле MyCourse типу Course, через яке можна отримати доступ до єдиного екземпляру класу. Опис полів класу розташовано у Таблиці 3.13.

Таблиця 3.13.

|  |  |  |
| --- | --- | --- |
| Назва поля | Тип | Опис поля |
| MyCource | Cource | Курс дисциплін та відомості про нього (реалізація шаблону проектування Singleton) |
| GetServerSubpath | string | Адреса розташування файлу з даними на сервері |
| InstituteCode | string | Код підрозділу |
| RootFolderPath | string | Адреса папки розташування навчальних матеріалів на локальному диску |
| CourceNumber | string | Номер курсу |
| Semester | string | Номер семестру |
| GroupName | string | Код групи |
| SpecialityCode | string | Код спеціальності |
| SpecialityName | string | Назва спецальності |
| SelectedSubject | Subject | Обрана дисципліна |
| Subjects | List<Subject> | Список дисциплін, які вивчаються у курсі |

# Клас Subject

Поля класу Subject зберігають дані про дисципліну, яка вивчається у поточному семестрі. Опис полів класу розташовано у Таблиці 3.14.

Таблиця 3.14

|  |  |  |
| --- | --- | --- |
| Назва поля | Тип | Опис поля |
| Name | string | Назва дисципліни |
| Hours | string | Кількість годин на вивчення теоретичного матеріалу |
| Teacher | string | ПІБ викладача |
| SelectedTopic | Topic | Обрана тема |
| Topics | List<Topic> | Список тем, які вивчаються |

# Клас Topic

Клас Topic призначено для зберігання даних про тему, яку обрав для вивчення користувач. Опис полів класу розташовано у Таблиці 3.15.

Таблиця 3.15.

|  |  |  |
| --- | --- | --- |
| Назва поля | Тип | Опис поля |
| Name | string | Назва теми |
| LectionUri | string | Адреса розташування файлу з матеріалом лекції на локальному диску |
| TestUri | string | Адреса розташування файлу з матеріалом тестування лекції на локальному диску |
| Opacity | double | Прозорість елементу у списку тем (визначає, чи можна перейти до вивчення теми) |
| CircleColor | Brush | Колір відмітки про вивчення теми (зелений- вивчено, червоний – не вивчено, сірий – не доступно) |

# Клас Quiz

Клас Quiz призначено для зберігання дешифрованих даних файлу тестування. Опис полів класу розташовано у Таблиці 3.16.

Таблиця 3.16.

|  |  |  |
| --- | --- | --- |
| Назва поля | Тип | Опис поля |
| MinPoints | int | Мінімальна кількість балів (порогове значення), яку потрібно набрати для проходження тесту |
| TestName | string | Назва тесту |
| Questions | List<QuizItem> | Список питань |

# Клас QuizItem

Екземпляр класу QuizItem зберігає інформацію про питання у тесті. Опис полів класу розташовано у Таблиці 3.17.

Таблиця 3.17.

|  |  |  |
| --- | --- | --- |
| Назва поля | Тип | Опис поля |
| Text | string | Текст питання |
| Answers | List<string> | Список варіантів відповідей |
| Values | List<string> | Флаги відповідей (у порядку їх розташування у колекції Answers) |
| IsOneTrue | bool | Кількість правильних відповідей на питання |

Поле IsOneTrue використовується під час визначення типу елементу керування на сторінці тестування: якщо поле приймає значення «істина», кожен варіант відповіді розташовується у елементі керування RadioButton, інакше – Checkbox.

# Класи Department та Specialty

Класи Department та Specialty використовуються для зберігання даних про структуру навчального закладу, який використовую систему «Лектор» для навчання студентів. Дані класи використовуються лише при першому вході користувача у систему та містять дані, які зберігаються у файлі University.xml. Опис полів класу Department розташовано у Таблиці 3.18, класу Specialty – у Таблиці 3.19.

Таблиця 3.18.

|  |  |  |
| --- | --- | --- |
| Назва поля | Тип | Опис поля |
| Name | string | Назва підрозділу |
| FolderName | string | Код підрозділу (також ім’я папки підрозділу на сервері латинськими літерами) |
| Specialities | List<Speciality> | Список спеціальностей підрозділу |

Таблиця 3.19.

|  |  |  |
| --- | --- | --- |
| Назва поля | Тип | Опис поля |
| Name | string | Назва спеціальності |
| Code | string | Код спеціальності |
| FolderName | string | Назва папки спеціальності на сервері |
| Cources | List<int> | Список курсів |
| IsEnabled | bool | Флаг доступності |

# Клас UserData

Клас UserData призначено для зберігання особистих даних користувача, які буде відображено на сторінці UserDataPage.xaml. Кожен екземпляр класу UserData зберігає дані тільки про один елемент персональних даних користувача (прізвище, ім’я, назву спеціальності і т. п.).

Поля класу UserData описано у Таблиці 3.20.

Таблиця 3.20.

|  |  |  |
| --- | --- | --- |
| Назва поля | Тип | Опис поля |
| Key | string | Назва тегу у файлі користувача, значення якого можна змінити |
| Value | string | Значення |
| Title | string | Назва значення (елементу) |
| Tag | int | Тег |

# Висновки до розділу

В даному розділі було розглянуто основні засоби та інструменти розробки, які використовуються для розробки ПЗ. Вказано, що найбільш оптимальною мовою програмування для розробки системи є C#, а середовищем – Visual Studio 2015. Також наведено опис використаних сторонніх та стандартних інструментів.

Також описано формати файлів, із якими працюватиме система. побудовано та описано основні модулі системи навчального призначення «Лектор».

РОЗДІЛ 4

# Тестування системи навчального призначення

Тестування програмного забезпечення – це процес, що використовується для виміру якості розроблюваного програмного забезпечення. Це один з найбільш дорогих етапів життєвого циклу програмного забезпечення, на який відводиться не менше 40% витрат.

Існує велика кількість способів тестування, які поділено на класи за об’єктом тестування, знанням системи, ступенем автоматизації, ступенем ізольованості компонентів, часом проведення тестування.

В процесі розробки системи навчального призначення «Лектор» було проведено тестування різних рівнів:

* Тестування компонентів;
* Тестування модулів;
* Тестування системи.

Тестування проводилося за допомогою засобів середовища розробки Visual Studio 2015. Докладно процес тестування та інструменти описано у документі «Програма та методика тестування».

Програмні коди модульних тестів розташовано у Додатку З.

# Висновки до розділу

В даному розділі було вказано, що тестування є важливим етапом життєвого циклу програмного забезпечення. Також вказано документ, у якому описано процес тестування та наведено результат.

# ВИСНОВКИ

В ході проведеної роботи було спроектовано та реалізовано програмний засіб, призначений для вивчення та закріплення теоретичного матеріалу навчальних дисциплін. Розроблений ПЗ відповідає усім вимогам, описаним в технічному завданні.

В рамках даної роботи було вирішено наступні завдання:

* обробка даних користувача;
* завантаження та обробка навчальних матеріалів з файлового сервера на локальний диск;
* завантаження, обробка та представлення даних про курс дисциплін;
* пошук навчальних матеріалів на локальному диску;
* обробка та представлення лекційного матеріалу;
* оцінювання знань користувача з вивченої дисципліни;
* надання користувачу можливості ознайомитися та змінити введені дані.

Для подальшого розвитку до розробленої системи можна внести наступні поліпшення:

* розробити мобільні версії додатку, щоб надати учням можливість вивчати теоретичний матеріал у більш зручних умовах;
* додати онлайн-тестування з вивченої дисципліни;
* створити додаткові види контролю вивчення теми або дисципліни, наприклад, запис відео-відповідей, написання есе та ін.;
* надати студентам можливість отримати додаткові матеріали до вивченої теми (наприклад, таблиці, книжки та ін.).

Розроблена система навчального призначення може бути впроваджена для будь-яких організацій та закладів, які займаються дистанційним або змішаним навчанням. Робота з програмним засобом вимагає від організації чи закладу, що її впроваджує, наявності файлового сервера та програми для шифрування тестів.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дистанційне навчання [Електронний ресурс]: стаття з електронної енциклопедії // Вільна енциклопедія Вікіпедія. – Режим доступу: https://uk.wikipedia.org/wiki/%D0%94%D0%B8%D1%81%D1%82%D0%B0%D0%BD%D1%86%D1%96%D0%B9%D0%BD%D0%B5\_%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F – Назва з титул. екрану.
2. Дистанционное образование - что это? [Електронний ресурс]: стаття з інформаційного порталу // Дистанционное обучение. – Режим доступу: http://www.distance-learning.ru/db/el/284C55060CD3C3B9C3256F2C0052CF9F/doc.html – Назва з титул. екрану.
3. Що таке дистанційна освіта? [Електронний ресурс]: стаття // Київський університет імені Бориса Грінченка. – Режим доступу: http://kubg.edu.ua/48-struktura/pidrozdili/ndl-informatizatsiyi-osviti/262-scho-take-distantsijna-osvita.html – Назва з титул. екрану.
4. Масові відкриті онлайн-курси [Електронний ресурс]: стаття з електронної енциклопедії // Вільна енциклопедія Вікіпедія. – Режим доступу: https://uk.wikipedia.org/wiki/%D0%9C%D0%B0%D1%81%D0%BE%D0%B2%D1%96\_%D0%B2%D1%96%D0%B4%D0%BA%D1%80%D0%B8%D1%82%D1%96\_%D0%BE%D0%BD%D0%BB%D0%B0%D0%B9%D0%BD-%D0%BA%D1%83%D1%80%D1%81%D0%B8 – Назва з титул. екрану.
5. Наша миссия [Електронний ресурс]: інформація про проект // Coursera. – Режим доступу: https://www.coursera.org/about/ – Назва з титул. екрану.
6. Наши партнеры [Електронний ресурс]: інформація про партнерів поекту // Coursera. – Режим доступу https://www.coursera.org/about/partners – Назва з титул. екрану.
7. 86 курсов ЕШКО для дистанционного обучения [Електронний ресурс] головна сторінка сайту// ЄШКО-Україна. – Режим доступу: https://eshko.ua/ – Назва з титул. екрану.
8. Дистанційна освіта [Електронний ресурс]: стаття // Міністерство Освіти та Науки України. – Режим доступу: http://mon.gov.ua/activity/education/distanczijna/distantciyna.html – Назва з титул. екрану.
9. Дистанційна освіта в Україні[Електронний ресурс]: стаття // Міністерство Освіти та Науки України. – Режим доступу: http://mon.gov.ua/activity/education/distanczijna/– Назва з титул. екрану.
10. Блог [Електронний ресурс]: головна сторінка // Prometheus. – Режим доступу: http://prometheus.org.ua/ – Назва з титул. екрану.
11. Стартують безкоштовні масові онлайн-курси підготовки до ЗНО [Електронний ресурс]: стаття // Prometheus. – Режим доступу: http://prometheus.org.ua/zno\_online\_courses/ – Назва з титул. екрану.
12. Rapid Growth in Online and Distance Education Worldwide [Електронний ресурс]: стаття // mastersportal. – Режим доступу: http://www.mastersportal.eu/articles/393/rapid-growth-in-online-and-distance-education-worldwide.html – Назва з титул. екрану.
13. C Sharp [Електронний ресурс]: стаття з електронної енциклопедії // Вільна енциклопедія Вікіпедія. – Режим доступу: https://uk.wikipedia.org/wiki/C\_Sharp – Назва з титул. екрану.
14. Tools for every developer and every app [Електронний ресурс]: стаття // Visual Studio. – Режим доступу: https://www.visualstudio.com/en-us/dn469161 – Назва з титул. екрану.
15. Page Class [Електронний ресурс]: стаття // MSDN. – Режим доступу: https://msdn.microsoft.com/en-us/library/system.windows.controls.page(v=vs.110).aspx – Назва з титул. екрану.
16. ZipFile – класс [Електронний ресурс]: стаття // MSDN. – Режим доступу: https://msdn.microsoft.com/ru-ru/library/system.io.compression.zipfile(v=vs.110).aspx – Назва з титул.
17. Using Adobe Reader in a WPF app [Електронний ресурс]: стаття // Code Project. – Режим доступу: http://www.codeproject.com/Articles/380019/Using-Adobe-Reader-in-a-WPF-app – Назва з титул.
18. Паттерн Singleton (одиночка,синглет) [Електронний ресурс]: стаття // CPP-REFERENCE. – Режим доступу: http://cpp-reference.ru/patterns/creational-patterns/singleton/ – Назва з титул.
19. Read and Write Excel Documents Using OLEDB [Електронний ресурс]: стаття // Code Project. – Режим доступу: http://www.codeproject.com/Tips/705470/Read-and-Write-Excel-Documents-Using-OLEDB – Назва з титул.
20. Модульное тестирование кода [Електронний ресурс]: стаття // MSDN. – Режим доступу: https://msdn.microsoft.com/ru-ru/library/dd264975.aspx – Назва з титул.
21. Лекція 11. Тестування програмного продукту [Електронний ресурс]: стаття // Библиотека МГПУ. – Режим доступу: http://lib.mdpu.org.ua/e-book/vstup/L11.htm – Назва з титул.

# ДОДАТОК А. Діаграми

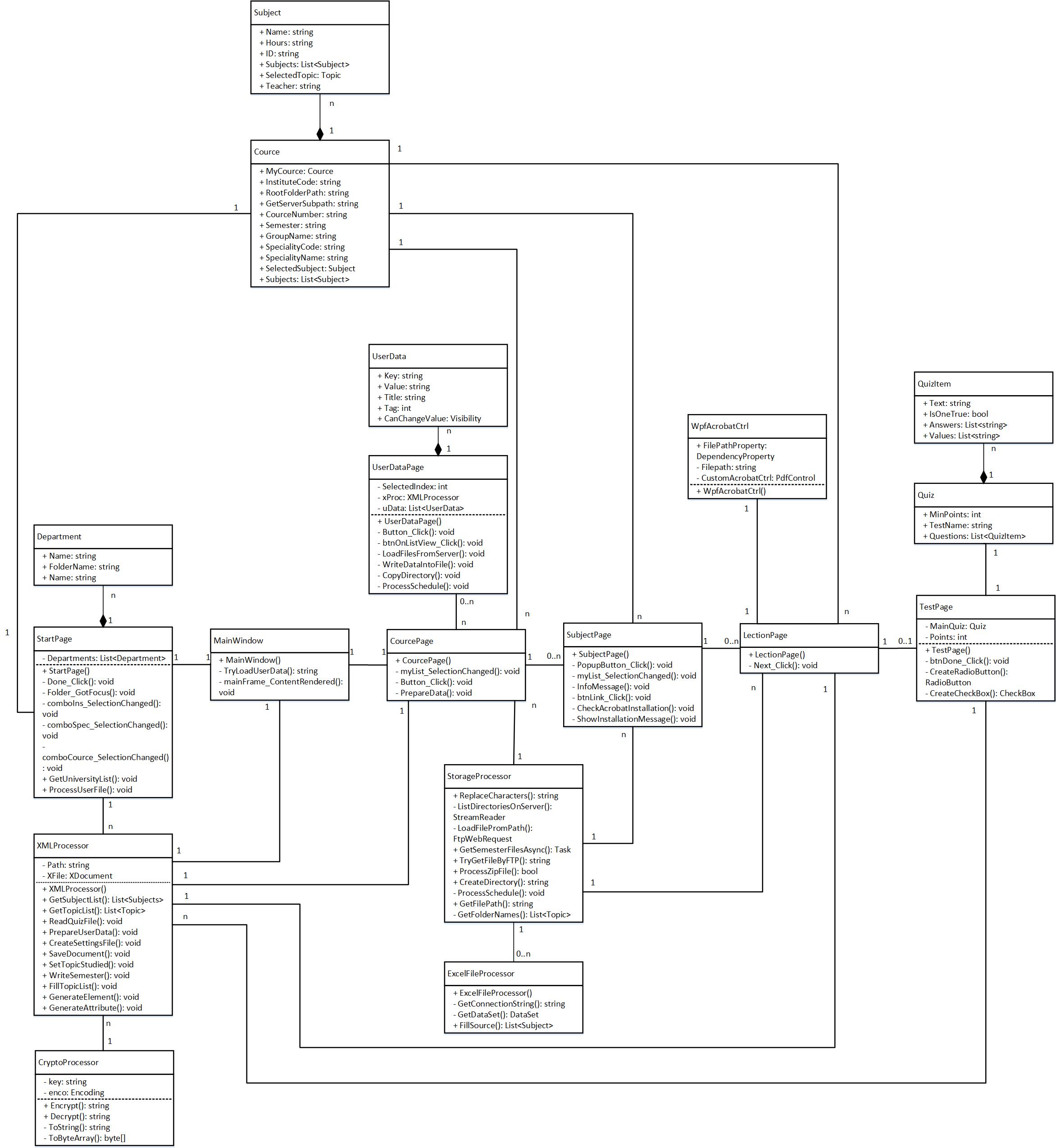


Рис. А.1. UML діаграма класів (Class diagram)

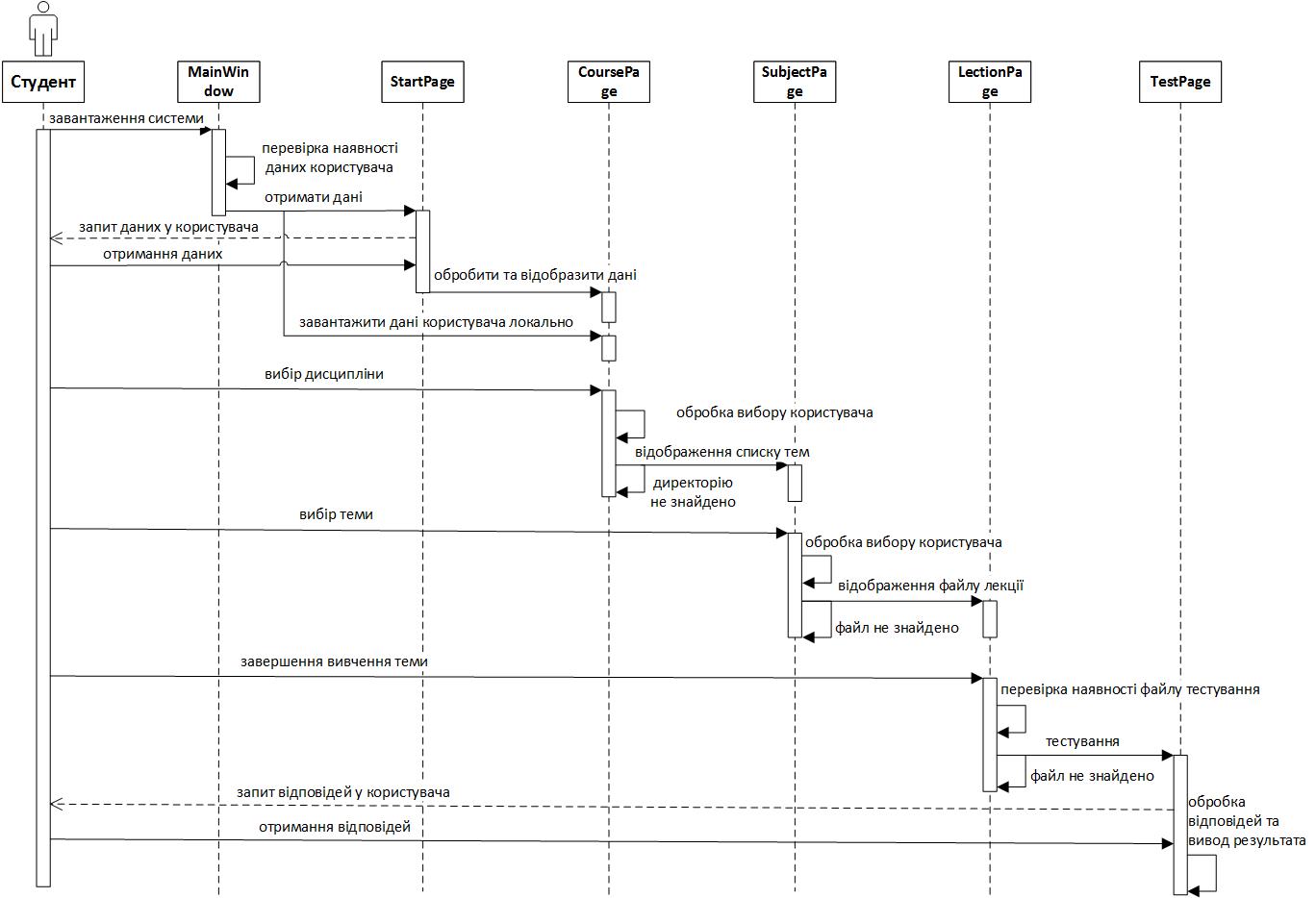


Рис. А.2. UML діаграма послідовності (Sequence diagrams)

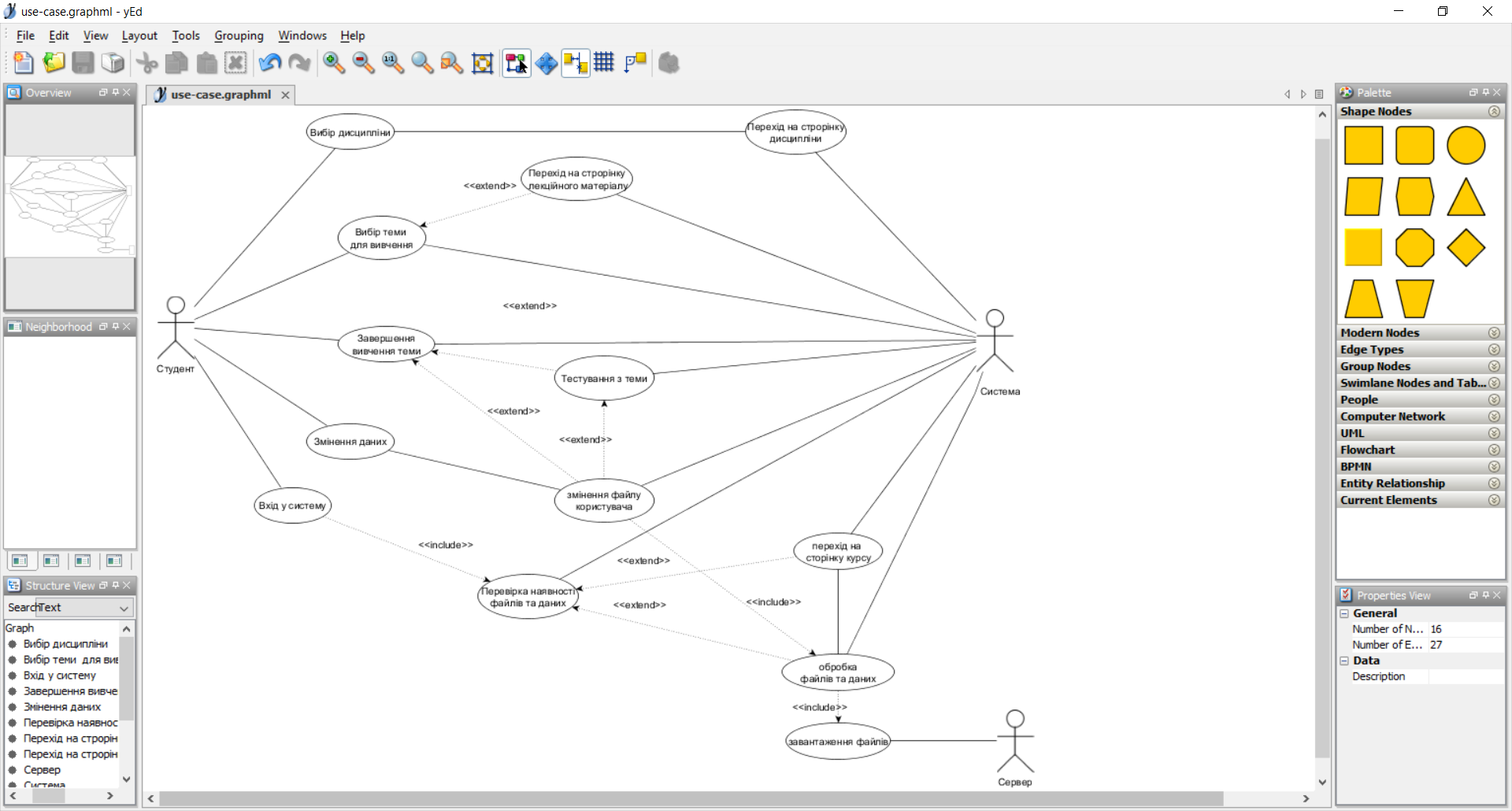


Рис. А.3. UML діаграма варіантів використання (прецедентів) (Use Case Diagram)

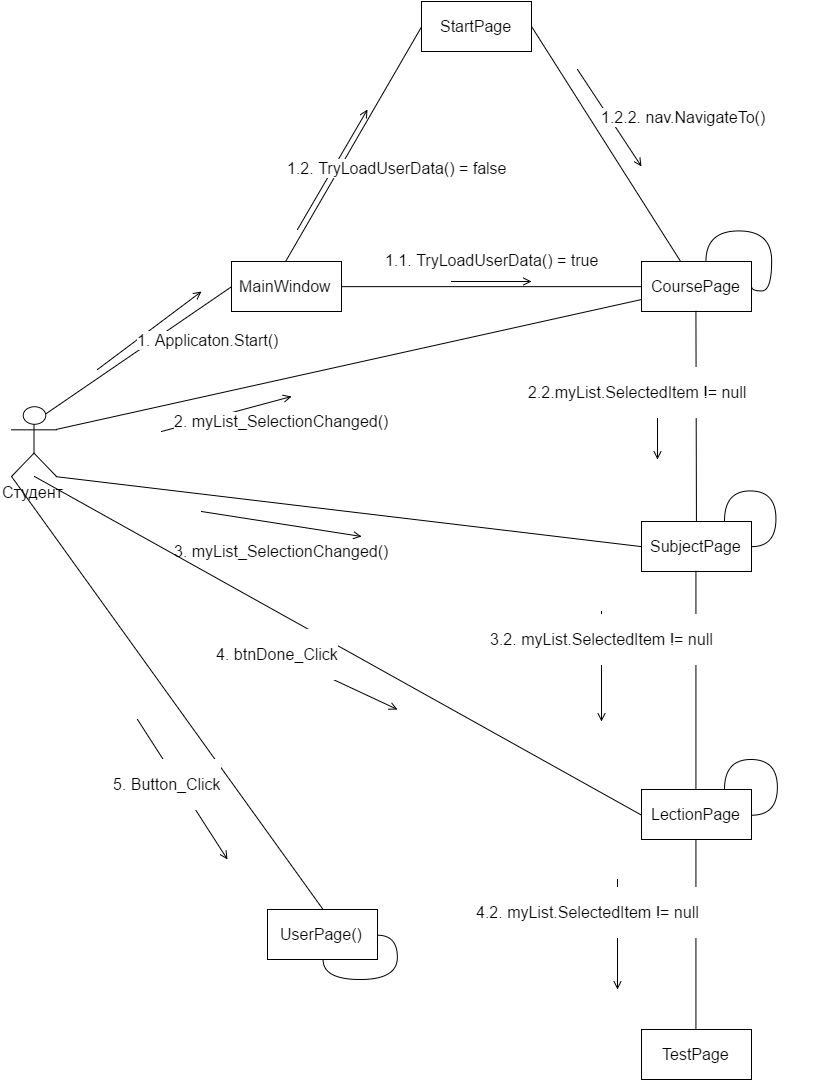


Рис. А.4. UML діаграма кооперації (Collaboration diagrams)

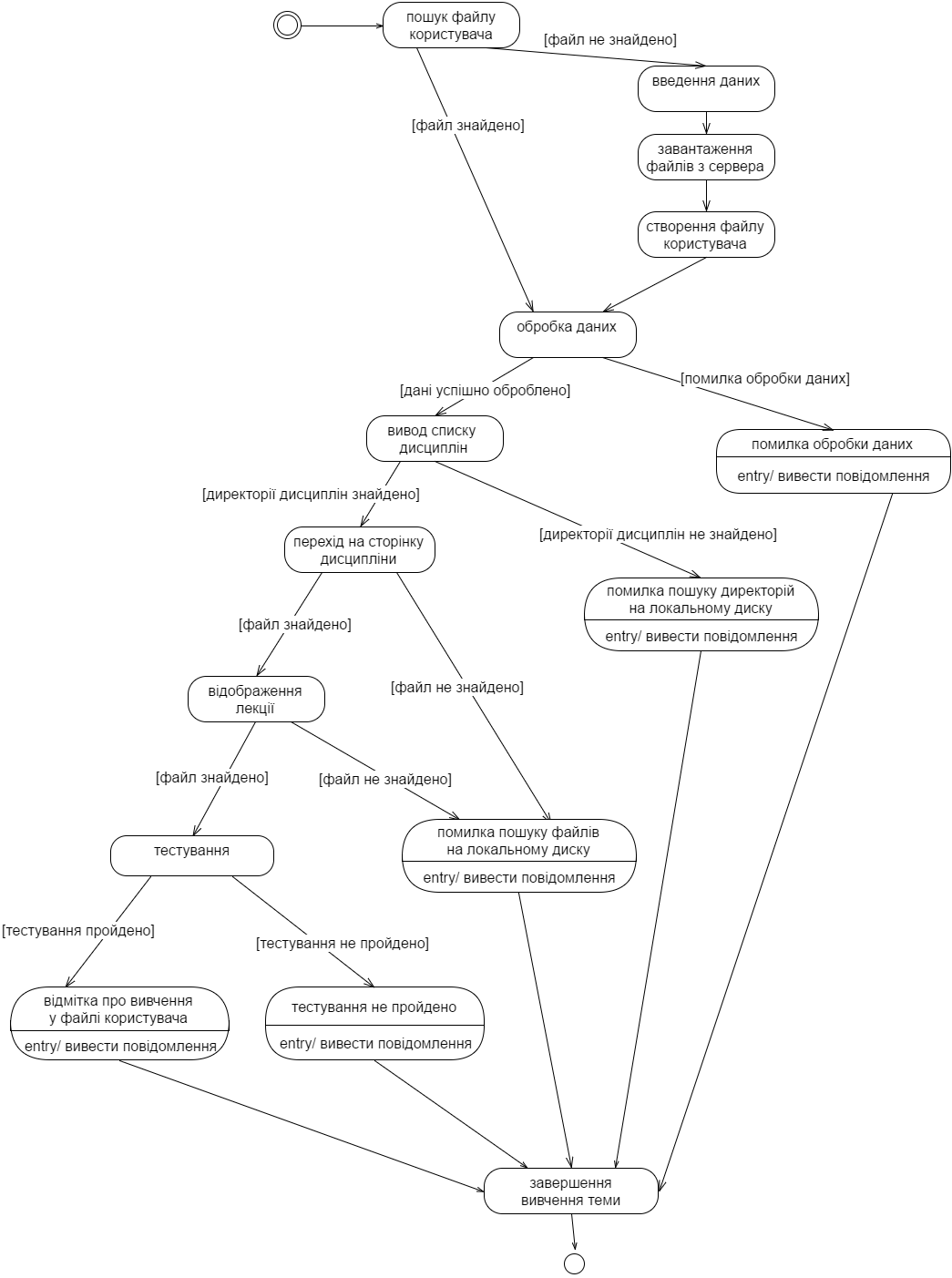


Рис. А.5. Діаграма дій

# ДОДАТОК Б. Блок-схема алгоритму тестування

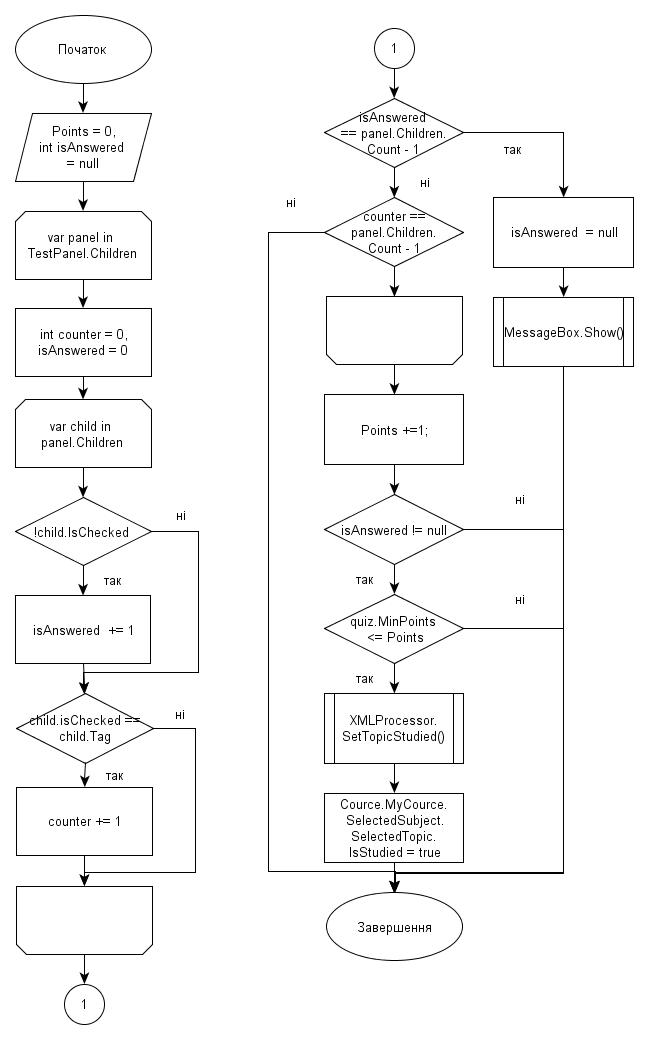


Рис. Б.1. Блок-схема алгоритму тестування

# ДОДАТОК В. Блок-схема алгоритму завантаження та обробки навчальних матеріалів

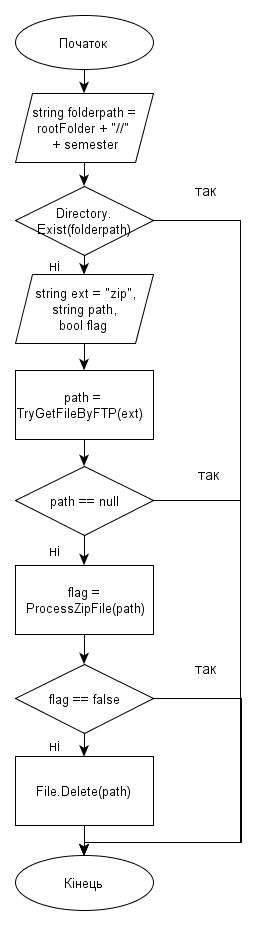


Рис. В.1. Блок-схема алгоритму

# ДОДАТОК Г. Блок-схема алгоритму завантаження та обробки розкладу

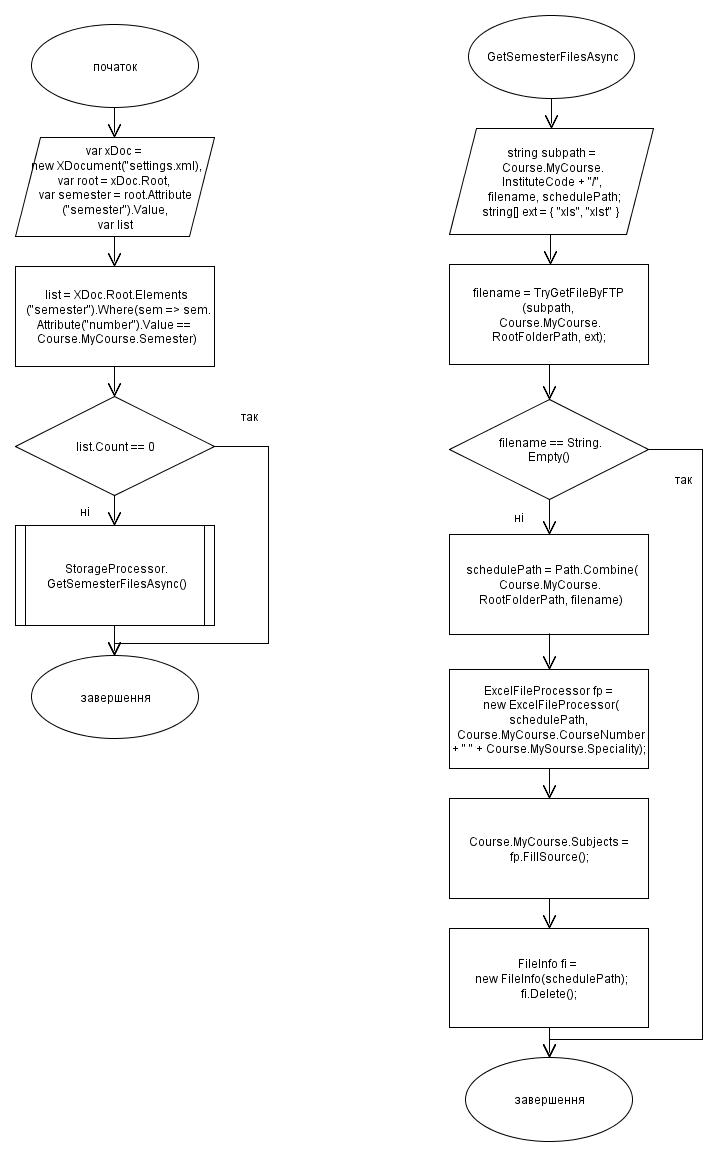
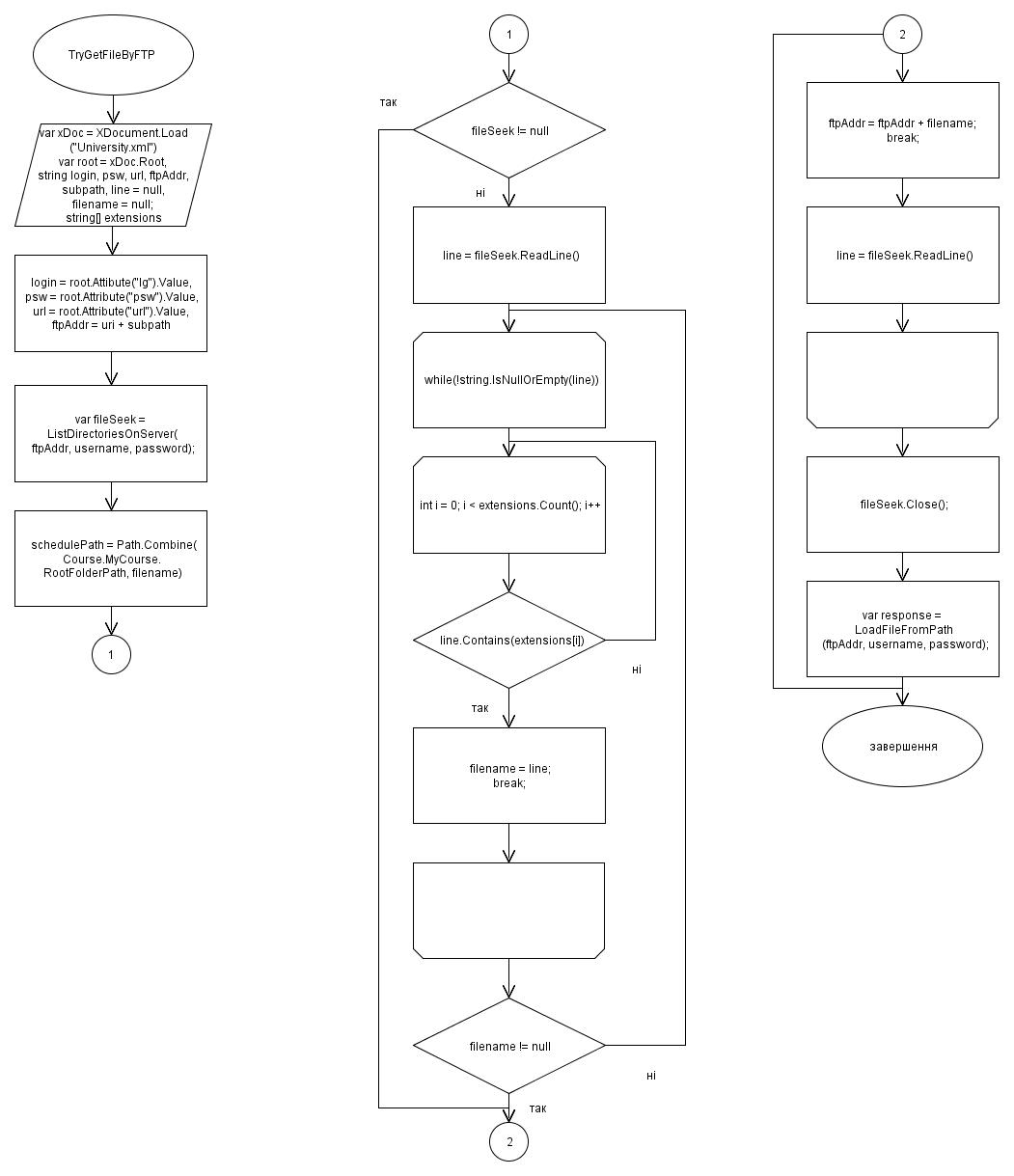


Рис. Г.1. Алгоритм основної функції та методу GetSemesterFilesAsync класу StorageProcessor

Рис. Г.2. Продовження блок-схеми алгоритму завантаження та обробки розкладу. Метод TryGetFileByFTP класу StorageProcessor



# ДОДАТОК Д. Блок-схема алгоритму обробки персональних даних користувача

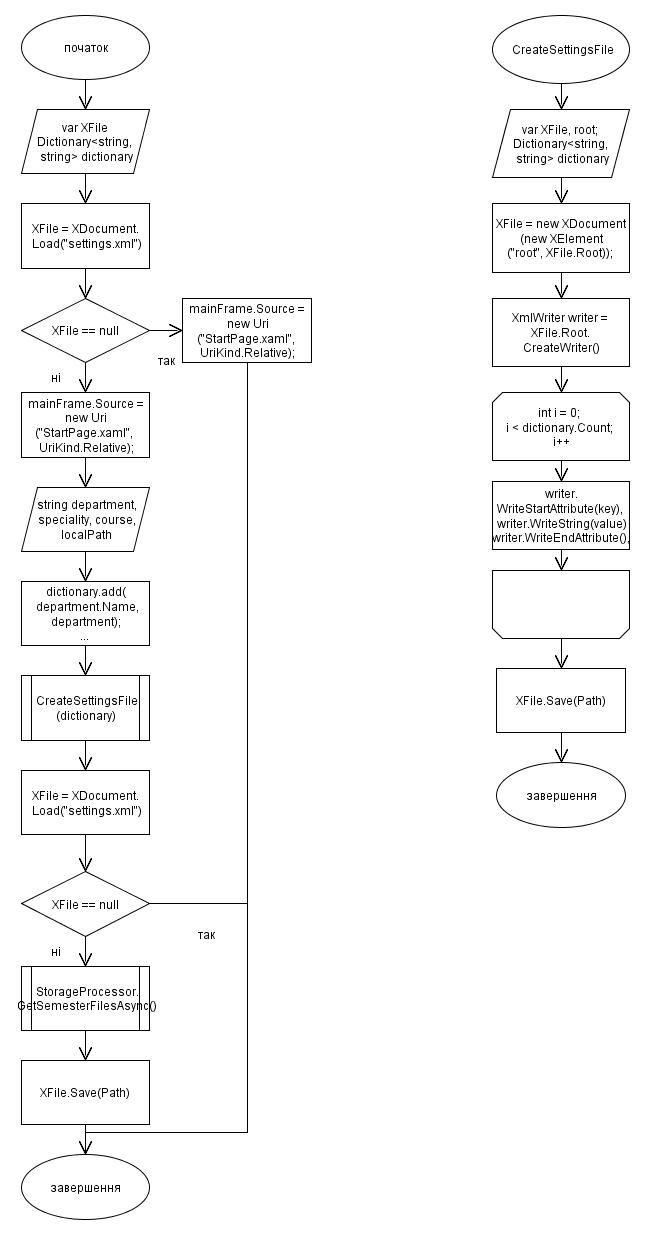


Рис. Д.1. Блок-схема алгоритму

# ДОДАТОК Ж. Блок-схема алгоритму пошуку навчальних матеріалів на локальному диску

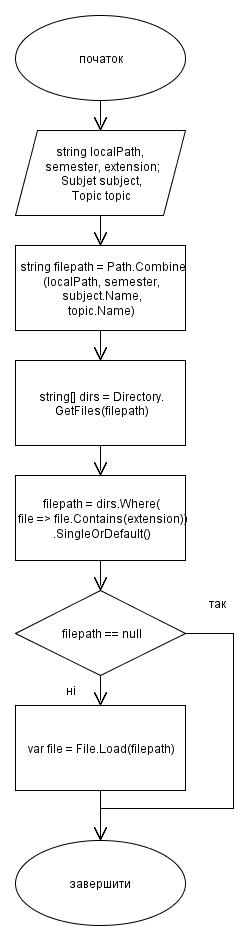


Рис. Ж.1. Блок-схема алгоритму

# ДОДАТОК З. Програмні коди модульних тестів

using Microsoft.VisualStudio.TestTools.UnitTesting;

using Lecturer.Data.Entities;

using Lecturer.Data.Processor;

namespace Lecturer.UnitTest

{

[TestClass]

public class UnitTest1

{

[TestMethod]

public void GetSchedule\_Test()

{

//arrange

Cource.MyCource.RootFolderPath = @"D:\";

Cource.MyCource.GroupName = "4 ПІ";

//act

StorageProcessor.ProcessSchedule("IFMIT");

//assert

Assert.AreNotEqual(null, Cource.MyCource.Subjects);

}

[TestMethod]

public void ProcessZip\_Test()

{

//arrange

Cource.MyCource.RootFolderPath = @"D:\";

string subfolder = "IFMIT/PI/8/";

string[] ext = { "zip" };

string path = StorageProcessor.TryGetFileByFTP(subfolder, Cource.MyCource.RootFolderPath, ext);

//act

bool flag = StorageProcessor.ProcessZipFile(path, Cource.MyCource.RootFolderPath);

//assert

Assert.AreEqual(true, flag);

}

}

}