

УДК 377.8

О.І.Гороховський, Т.І.Трояновська

Вінницький національний технічний університет

## РОЗРОБКА АЛГОРИТМІВ НЕПРЯМОГО ОЦІНЮВАННЯ ДЛЯ АДАПТИВНОЇ СДН

У статті наведені алгоритми непрямого оцінювання діяльності студента ДФН при роботі з навчальними матеріалами адаптивної системи дистанційного навчання (СДН).

**Постановка проблеми.** Головною проблемою традиційного навчання є його масовість. Кожен викладач веде групу студентів, з яких принаймні 15% оволодіють темою у повному обсязі. В свою чергу з цих 15% тільки 1-2% оволодіють нею настільки, що зможуть рухатись в цьому напрямку далі. Решта отримує або "шаблонні" знання на рівні штампів та типових розв'язків, або взагалі матиме загальні знання з предмету. У традиційному навчанні викладач, читаючи курс, завжди занурюється в навчальне середовище. Інтуїтивні (непрямі) оцінки, які він робить упродовж роботи, допомагають йому корегувати процес навчання в реальному часі. Дистанційне навчання такого не дозволяють, тому навчальне середовище для нього необхідно створити штучно, за допомогою технологічних засобів.

**Аналіз останніх публікацій та досліджень.** Останнім часом розробники адаптивних СДН (АСДН) значну увагу приділяють моделям та методам адаптивного контролю знань [1] та підвищенню рівня педагогічної взаємодії [2].

У зв'язку із цим до АСДН, що розробляється, пропонується увести додаткові контрольні елементи. Перший контрольний елемент носить назву непрямого оцінювання і складається із двох типів оцінок: прямих та непрямих. Прямими оцінками (ПО) називають явні види оцінювання успішності або рівня підготовки студента в процесі навчання. Непрямими оцінками (НО) вважатимемо неявні види оцінювання успішності або рівня підготовки студента в процесі навчання. Додатковим контрольним елементом є вдосконалення звичайного тестування шляхом використання двох каскадів тестових завдань. Також у рамках даної роботи вперше пропонується ввести та використовувати на практиці поняття предметно-орієнтованої домінанти студента дистанційної форми навчання. Поняття домінанти взято із робіт відомого вченого Ухтомського А. А. [3] та розширено до поняття предметно-орієнтованої домінанти (ПОД) для врахування особливостей роботи студента з навчальним матеріалом (НМ) адаптивного дистанційного курсу (АДК).

**Мета.** Розробити загальну схему механізму та алгоритми вимірювання НО.

**Виклад основного матеріалу.**

НО являють собою заміри часу, які відбуваються у певні ключові моменти діяльності студента та відслідковування траєкторії навчання студента в АДК.

Дані НО використовуються в АСДН для подальшої адаптації під особливості роботи кожного студента. Оскільки оцінка непряма, то сам процес вимірювання має бути прихованим від студента, а відтак може бути реалізований за допомогою фоновий нитки процесу (shadow thread), яка виконує необхідні функції [4, 5].

Агентом вимірювання НО є активний компонент веб-сторінки. На рисунку 1 показано функціональну модель такого компоненту. Даний рисунок розглядає випадок виникнення сигналу непрямої оцінки. В цьому випадку за допомогою компонента Selector їй відповідає обробник  $R_x$ , результатом роботи якого є множина змінюваних характеристик студента  $V_b$ , яка передається на АМОД, де формуються відповідні зміни в динамічній частині ПОД за допомогою функції  $f_{Act}(O)$  [6].

Bound Start Time – сервлет, який здійснює запис в базу даних стартової позначки таймера при переході на сторінку, або при початку читання лекції. Bound End Time – сервлет, який здійснює запис в БД кінцевої позначки таймера. Perform Test – сервлет, який обробляє дані, отримані під час проходження студентом тестового завдання, здійснює їх аналіз та видачу результату та рекомендацій по підвищенню якості освіти. Selector – аналізує отриманий від активного компонента сигнал і в залежності від нього підключає потрібний сервлет для обробки отриманих даних.

Сигнали визначають, яка саме подія відбулась на користувальницькій стороні (передбачено три можливих сигнали – старт часового виміру, пауза часового виміру і зупинка часового виміру).

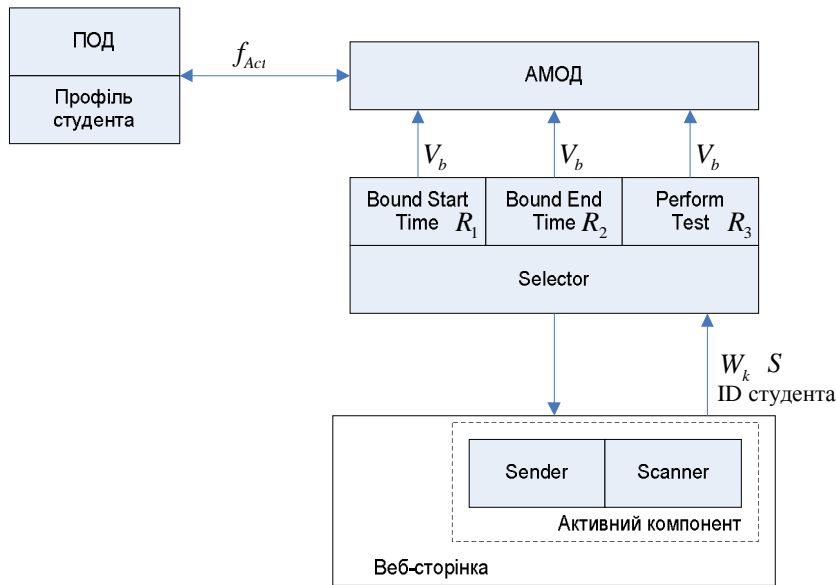


Рисунок 1. Схема агента вимірювання НО

Bound Start Time – сервлет, який здійснює запис в базу даних стартової позначки таймера при переході на сторінку, або при початку читання лекції. Bound End Time – сервлет, який здійснює запис в БД кінцевої позначки таймера. Perform Test – сервлет, який обробляє дані, отримані під час проходження студентом тестового завдання, здійснює їх аналіз та видачу результату та рекомендацій по підвищенню якості освіти. Selector – аналізує отриманий від активного компонента сигнал і в залежності від нього підключає потрібний сервлет для обробки отриманих даних.

Сигнали визначають, яка саме подія відбулась на користувальницькій стороні (передбачено три можливих сигнали – старт часового виміру, пауза часового виміру і зупинка часового виміру).

Sender – модуль активного компонента, який відповідає за пересилку сигнала сервлет. Scanner – слідкує за подіями на користувальницькій стороні.

З міркувань простоти та ефективності було вирішено реалізувати активний компонент у формі Java-аплету. Цей підхід має такі переваги перед використанням ActiveX-елементу, або складного JavaScript-рішення:

- кросплатформеність. В силу стандартизованості мови програмування Java виконання цього аплету не залежить від марки браузера та марки операційної системи;
- наявність спеціалізованих бібліотек для прямої роботи із форматами HTML, доступу до внутрішньої моделі подій браузера та ін.

За своєю структурою аplet являє собою фонову нитку процесу всередині браузера, якій виділяється окремий фрагмент сторінки, на якому може бути розміщений користувальницький інтерфейс аплету. В мові HTML для цього передбачений спеціалізований тег applet, який описує розташування такого компонента [7, 8]. Крім того, за допомогою цього тега перед завантаженням програми можна передати певні параметри, які задаватимуть, наприклад, IP-адресу АМОД, за якою слід надсилати вимірні дані.

Запуск аплету відбувається одночасно із завантаженням веб-сторінки на робочу станцію користувача, а зупинка процесу настає після того, як ця сторінка ним закривається, або відбувається перехід на наступну. Під час перебування користувача на сторінці, де розташовано аplet, його компонент має необмежений доступ до контенту сторінки, може реагувати на дії користувача через обробники подій та логічне ім'я. Доступ до контенту сторінки забезпечується спеціалізованим об'єктом JScript [9].

Доступ до контенту відбувається відповідно до об'єктної моделі документа (DOM – Document Object Model), тобто за деревовидною структурою, кожен вузол якої має перелік властивостей, а також може містити підвузли [10]. Кожен документ за структурою DOM, може містити один елемент типу документа та один кореневий елемент, який містить решту вузлів [10].

Аплет може читати та оперативно змінювати дані. Ця особливість зв'язку між аплетом та сторінкою лежить в основі технології AJAX – Asynchronous Java Active Extension [11]. Оскільки аплет має доступ до історії переходів користувача зі сторінки на сторінку, це дозволяє у подальшому відстежувати шлях студента по курсу у процесі навчання [12, 13].

Отже, для роботи із контентом сторінки у вигляді DOM-дерева в об'єкті JSoject можна виділити дві схеми:

1) Схема 1. Пошук вглиб по дереву DOM має на меті знайти необхідний для оновлення/збору даних об'єкт, і, маніпулюючи його властивостями, вносяться зміни в структуру веб-сторінки. Алгоритм пошуку вглиб об'єкта DOM наведено на рисунку 2.

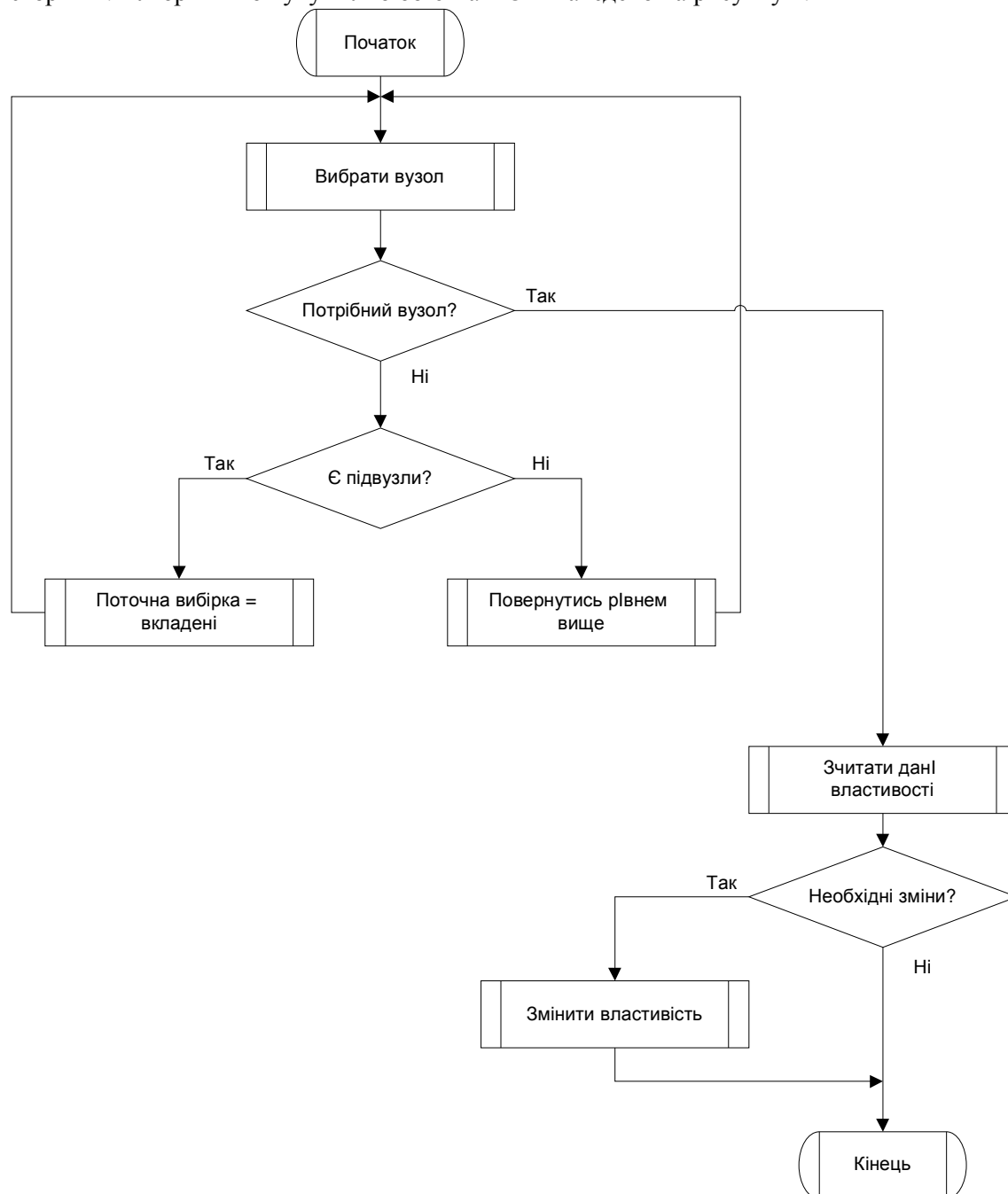


Рисунок 2. Пошук вглиб об'єкта DOM

Дана схема дозволяє аплету діяти так само, як і сценарій на мові JavaScript, встановлюючи або зчитуючи дані з властивостей окремих компонентів веб-сторінки. За допомогою такої схеми також можна реагувати на дії користувача, наприклад, введення даних в елементи форми, або прокрутку сторінки. Однак ця схема обмежена об'єктною моделлю DOM, і не дозволяє тонких маніпуляцій з контентом.

2) Схема 2. Засобами пошуку вглиб знаходиться необхідний фрагмент документа, за допомогою властивості innerHTML, яку має кожен вузол DOM-дерева, отримується фрагмент вихідного коду програми, пересилається в бібліотеку синтаксичного розбору HTML-коду, і маніпуляції відбуваються на його рівні. Алгоритм пошуку вглиб фрагмента документа наведено на рисунку 3.

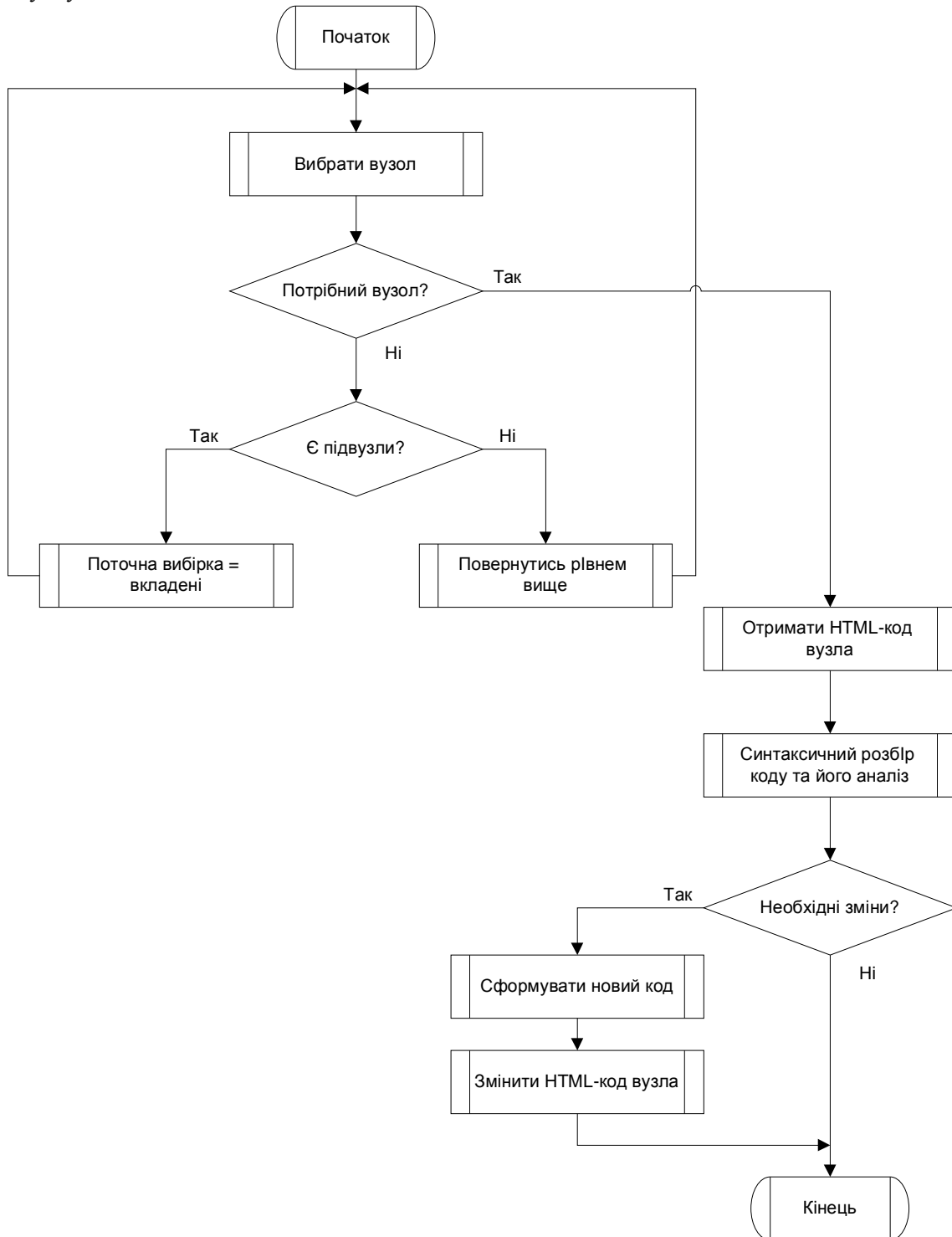


Рисунок 3. Пошук вглиб фрагмента документа

Вміст даної властивості за допомогою методів JavaScript замінюється на новий, якщо це необхідно. Така схема доцільна в тому випадку, якщо необхідні точні маніпуляції над вмістом сторінки.

Оскільки замірювання часу читання тексту або здачі тесту очевидно є проміжком часом, який знаходиться між завершенням завантаження сторінки та переходу до наступної, для збору таких даних можна скористатись схемою 1 для того, щоб заміряти цей час.

При завантаженні сторінки в момент ініціалізації аплету можна запустити окремим фоновим процесом лічильник, який вимикатиметься при переході студента на наступну сторінку [14]. Таким чином, роботу такого аплету по збору даних для НО можна описати алгоритмом, який показано на рисунку 4.

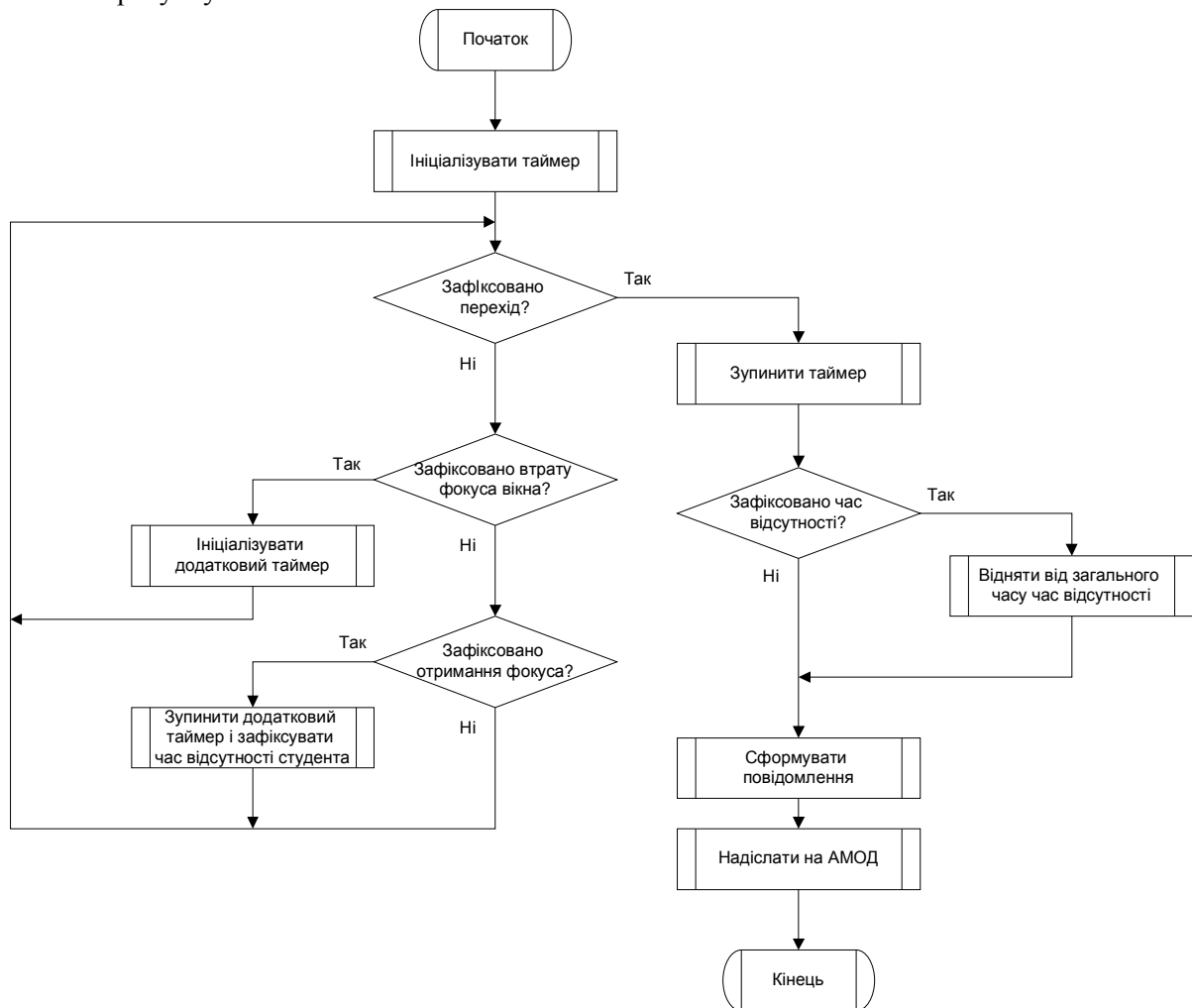


Рисунок 4. Алгоритм роботи аплету непрямої оцінки вимірювання часу

Для вимірювання кількості прочитаного тексту лекції можна виміряти вихідний код програми, з якого видалено всі HTML-теги. Алгоритм НО вимірювання обсягу тексту приведено на рисунку 5.

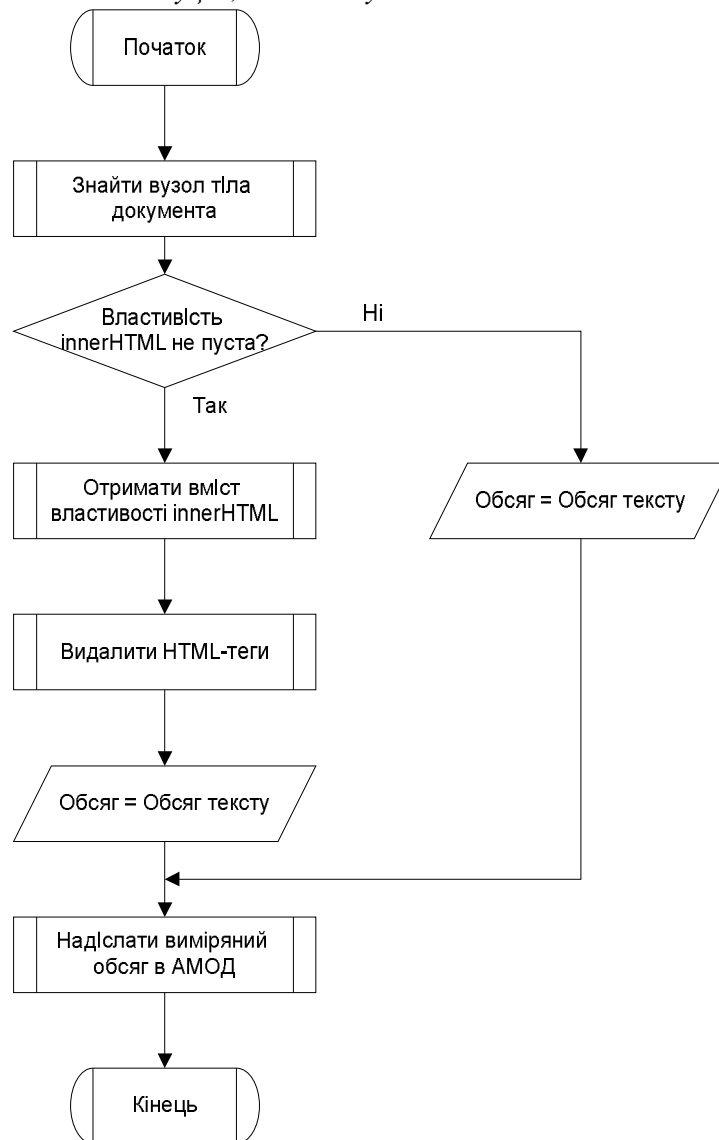


Рисунок 5. Алгоритм роботи вимірювання НО обсягу тексту

У якості цільового каталогу слід вибрати службовий каталог, який зареєстрований в АСДН як джерело АДК.

**Висновки.** Нині зростає інтерес до нового виду навчання – дистанційного. ДН та системи, що його забезпечують, набули та продовжують набувати популярності в зв'язку з стрімким розвитком науки, техніки та інформаційних технологій. Все більше користувачів персональних комп'ютерів стають студентами віртуальних навчальних закладів. Найбільш відомими є СДН, що пропонують індивідуалізований підхід до діяльності студента під час отримання знань. У даній АСДН пропонується механізм непрямого оцінювання, як засіб організації індивідуального підходу до засвоєння навчального матеріалу студентом АДК.

1. Сысоева Л. А. Динамика использования Интернет-сервисов студентами российских университетов (на примере РГГУ) // *Educational Technology & Society*. – №9(3). – 2006. С. – 307-312.
2. Лавров О. А. Дистанционной обучение: экспресс-организация процесса // *Educational Technology & Society*. – №7(1). – 2004. С. – 172-181.
3. Ухтомский А.А. Доминанта. – Санкт-Петербург: Питер, 2002. - 448с.
4. Гороховський О.І., Трояновська Т.І. Застосування UML при проектуванні засобів дистанційного навчання // *Інформаційні технології та комп'ютерна інженерія №2, 2005. Вінниця, ВНТУ, 2005. С. 72-77.*
5. Cockburn A. *Writing Effective Use Cases* – Addison-Wesley, 1999-2000.
6. Гороховський О.І., Трояновська Т.І. Інформаційна технологія розробки адаптивних дистанційних курсів // *Інформаційні технології та комп'ютерна інженерія №2, 2009. Вінниця, ВНТУ, 2009. С. 75-80.*
7. Гороховський О.І., Трояновська Т.І. Аспекти створення, моделювання та практики автоматизованих систем дистанційного навчання // «Автоматика-2006» XII МК, ВІННИЦЯ, 25-28 вересня 2006р., Вінниця. В.:ВНТУ, 2006. С. – 453.
8. Oaks S., Wong H. / *Java Threads, 2nd Edition*. O'Reilly, 1999.
9. Musciano C., Kennedy B. / *HTML: The Definitive Guide, 3rd Edition* – O'Reilly, 1998.
10. Nagappan R., Steel C. / *Java Security Architecture* – InformIT, 2006.
11. LeHors A., LeNegaret P., Wood L., Nicol G., Robie J., Champion M., Byme S. / *Document Object Model (DOM) Level 2 Specification* – W3C Recommendations, 13 November 2000.
12. Трояновська Т.І. Розробка комп'ютерної підсистеми аналізу та формування предметно-орієнтованої домінанти студента системи дистанційного навчання // *Вісник Черкаського державного технологічного університету. Черкаси, ЧДТУ, 2007. Вип. 3-4. С. 41-46.*
13. Трояновська Т.І. Розробка комп'ютерної підсистеми аналізу та формування предметно-орієнтованої домінанти студента системи дистанційного навчання // *НТК «Обробка сигналів та негауссівських процесів» пам'яті проф. Кунченка Ю.П., 21-26 травня, Черкаси. Ч.: ЧДТУ, 2007. С. – 202-204.*
14. Chapell D., Jewell T. / *Java Web Services* – O'Reilly, 2002.