

Ім'я користувача:
Титенко Сергій Володимирович

ID перевірки:
1013196344

Дата перевірки:
05.12.2022 17:26:35 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
05.12.2022 17:32:42 EET

ID користувача:
91350

Назва документа: PolienovaVA_TV-11mp_magistr_2022

Кількість сторінок: 69 Кількість слів: 10795 Кількість символів: 80009 Розмір файлу: 5.27 MB ID файлу: 1012959499

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

0%
Схожість

Збіги відсутні

0.29% Цитат

Цитати

3

Сторінка 72

Вилучення списку бібліографічних посилань вимкнене

17.7%
Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 10 слів та 10...

5.1% Вилучення з Інтернету

117

Сторінка 73

17.6% Вилученого тексту з Бібліотеки

701

Сторінка 75

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

17

Підозріле форматування

15
сторінок

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»

Навчально-науковий інститут атомної та теплової енергетики

Кафедра інженерії програмного забезпечення в енергетиці

"На правах рукопису"
УДК _____

«До захисту допущено»

В.о. зав.кафедри

Олександр КОВАЛЬ

“ ” _____ 202_ р.

Магістерська дисертація

За освітньою програмою «Інженерія програмного інтелектуальних кібер-фізичних систем і веб технологій»

Спеціальності 121 Інженерія програмного забезпечення

на тему: Мобільний застосунок онтологічно-орієнтованої навчальної системи з широким застосуванням карт понять.

Виконала: студентка 2 курсу, групи ТВ-11мп

Поленова Валентина Андріївна

(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник к.т.н., доцент Титенко Сергій Володимирович

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2022

**Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»**

Навчально-науковий інститут атомної та теплової енергетики

Кафедра інженерії програмного забезпечення в енергетиці

Рівень вищої освіти другий, магістерський

За освітньою програмою «Інженерія програмного інтелектуальних кібер фізичних систем і веб технологій»

Спеціальності 121 Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

в.о. зав. кафедри

Олександр КОВАЛЬ

(підпис)

« ____ » _____ 202_р.

**З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ**

Поленовій Валентині Андріївні

(прізвище, ім'я, по батькові)

1. Тема дисертації:

Мобільний застосунок онтологічно-орієнтованої навчальної системи з широким застосуванням карт понять.

Науковий керівник к.т.н, доцент Титенко Сергій Володимирович

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від "07" листопада 2022 року №4067-с

2. Строк подання студентом дисертації 05 грудня 2022 року.

3. Вихідні дані до роботи: Мова програмування Dart, фреймворк Flutter, API навчального порталу.

4. Перелік питань, які потрібно розробити: Провести дослідження сучасних навчально-інформаційних веб-порталів та мобільних систем на базі карт понять. Провести порівняльний аналіз існуючих програмних рішень. Визначити головні функціональні можливості навчального застосунку на базі карт понять, що були знайдені в аналогах. Розробити унікальний дизайн користувацького інтерфейсу. Спроекувати діаграму прецедентів та діаграму класів. Обрати технології для реалізації програмного продукту. Розробити необхідні модулі програмної системи та об'єднати їх. Розробити функціональні можливості програмної мобільної системи. Імплементувати розроблений дизайн та функціонал. Об'єднати усі складові застосунку. Протестувати готовий програмний продукт, виправити помилки та покращити функціональні можливості мобільного додатку.

5. Орієнтований перелік ілюстративного матеріалу: Титульна сторінка, Актуальність, Огляд інших систем, Mind Map Maker, CoMPASS, Інтерактивні карти понять для вивчення математики, Навчальний веб-портал, Діаграма прецедентів мобільного застосунку, Діаграма класів, Засоби розробки, API навчального порталу, Схема аналізу відношень, Схема статистичних даних, Файлова структура проекту, Приклад аналізу відношень всередині системи, Приклад статистичних даних всередині системи, Інші сторінки застосунку, Навігація, Висновки.
6. Орієнтований перелік публікацій: Методи підвищення наочності карт понять у навчальних мобільних застосунках, Лінійний навчальний шлях як метод підвищення наочності карт понять у мобільних застосунках, Educational mobile application based on concept maps.
7. Дата видачі завдання «01» жовтня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	строки виконання етапів магістерської дисертації	Примітка
1	Отримання завдання	01.10.21	виконано
2	Дослідження предметної області	01.10.21-01.11.21	виконано
3	Постановка вимог до проектування системи	01.11.21-15.11.21	виконано
4	Дослідження існуючих рішень	15.11.21-01.12.21	виконано
5	Підготовка публікацій	01.12.21-09.12.21	виконано
6	Розробка програмного продукту	05.03.2022-07.10.2022	виконано
7	Тестування	07.10.2022-11.11.2022	виконано
8	Захист програмного продукту	17.10.2022-21.10.2022	виконано
9	Підготовка магістерської дисертації	22.10.2022-20.11.2022	виконано
10	Передзахист	21.11.2022-25.11.2022	виконано
11	Захист	19.12.2022-23.12.2022	виконано

Студент

(підпис)Полєнова В.А.
(прізвище та ініціали)

Науковий керівник

(підпис)Титенко С.В.
(прізвище та ініціали)

РЕФЕРАТ

Обсяг дипломної роботи. Робота складається із вступу, семи розділів, висновку, списку використаних джерел із 17 найменувань, 3 додатків. У дисертації наявні 35 рисунків та 19 таблиць. Обсяг роботи — 82 сторінки, з них обсяг списку використаних джерел — 3 сторінки.

Актуальність теми. Карти понять є потужним інструментом візуалізації та онтологічного моделювання в навчальних системах. Зростання популярності електронного навчання та навчальних веб-ресурсів, а також використання мобільних застосунків, зумовлює необхідність створення мобільних навчальних порталів для дистанційної освіти.

Мета і задачі дослідження. Метою даної роботи є розробка навчальної мобільної системи з широким застосуванням карт понять з функцією аналізу відношень між вершинами та навчальною статистикою.

Завдання наукового дослідження:

- проаналізувати існуючі програмні системи для дистанційного навчання на базі карт понять;
- проаналізувати сучасні методи побудови карт понять на мобільних пристроях;
- обрати оптимальні функціональні можливості навчального мобільного застосунку;
- обрати засоби реалізації програмної системи;
- розробити архітектуру та структуру програмного забезпечення;
- провести моделювання розроблених алгоритмів побудови карт понять на мобільних пристроях;
- розробити застосунок на базі операційних систем Android та IOS для електронного навчання з використанням карт понять.

Об'єкт дослідження. Об'єктом дослідження є процес навчання за допомогою інтерактивних карт понять у мобільних системах.

Предмет дослідження. Предметом дослідження є мобільна система для електронного навчання із застосуванням інтерактивних карт понять.

Методи дослідження. Алгоритми візуалізації графів, об'єктно-орієнтований аналіз, онтологічне моделювання предметної області навчання.

Практичне значення одержаних результатів. Розроблену систему можна застосовувати в науковій та освітній галузі для більш ефективного дистанційного навчання.

Публікації.

1. Поленова В. А., Феденко В. А., Коваленко Д. Р., Титенко С. В. Методи підвищення наочності карт понять у навчальних мобільних застосунках // Сталий розвиток — XXI століття. Дискусії 2020: колективна монографія / Національний університет “Києво- Могиллянська академія” / за ред. проф. Хлобистова Є.В. — Київ, 2020. — С. 435-441 с. — Електронне видання. ISBN: 978-617-7668- 22-9.
2. Поленова В.А. Лінійний навчальний шлях як метод підвищення наочності карт понять у мобільних застосунках. // Сучасні проблеми наукового забезпечення енергетики. У 2-х т. / Матеріали XIX Міжнар. наук.-практ. конф. молод. вчених і студ., м. Київ, 20–23 / квіт. 2021 р. – Київ : КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2021. – Т. 2. – 196 - 197 с. – Електронне видання. ISBN 978-966-990-027-2.
3. Polienova V.A., Fedenko V.A., Tytenko, S.V. Educational mobile application based on concept maps. Modern engineering and innovative technologies, issue 23, part 1. 12–18, 2022. doi:10.30890/2567-5273.2022-23-01-014.

Ключові слова: КАРТА ПОНЯТЬ, НАВЧАЛЬНА СИСТЕМА, ПОНЯТТЯ, ТЕЗА, НАВЧАЛЬНА СТАТИСТИКА, ІНФОРМАЦІЙНО-НАВЧАЛЬНИЙ ПОРТАЛ, ОНТОЛОГІЧНЕ МОДЕЛЮВАННЯ, ЗАСТОСУНОК ДЛЯ МОБІЛЬНИХ ПРИСТРОЇВ.

ABSTRACT

The scope of the thesis. The work consists of an introduction, six chapters, a conclusion, a list of used sources from 17 titles, 3 appendices. There are 35 figures and 19 tables in the thesis. The volume of the work is 82 pages, of which the volume of the list of used sources is 3 pages.

Theme topicality. Concept maps are a powerful tool for visualization and ontological modeling in educational systems. The growing popularity of e-learning and educational web resources, as well as the use of mobile applications, necessitates the creation of mobile educational portals for distance education.

The purpose and objectives of the study. The purpose of this study is the development of an educational mobile system with a wide application of concept maps with the function of analyzing relationships between vertices and educational statistics.

Tasks of scientific research:

- to analyze existing software systems for distance learning based on concept maps;
- to analyze modern methods of building concept maps on mobile devices;
- choose the optimal functionality of the educational mobile application;
- to choose means of implementation of the software system;
- to develop the architecture and structure of the software;
- conduct simulations of the developed algorithms for building concept maps on mobile devices;
- to develop an application based on the Android and IOS operating systems for e-learning based on concept maps.

Object of study. The object of research is the learning process using interactive concept maps in mobile systems

Subject of study. The subject of the study is a mobile system for electronic learning using interactive concept maps.

Research methods. Power algorithm for construction of concept maps.

The practical significance of the obtained results. The developed system can be used in the scientific and educational fields for more effective distance learning.

Publications

1. Polienova V. A., Fedenko V. A., Kovalenko D. R., Tytenko S. V. Methods of increasing the visibility of concept maps in educational mobile applications // Sustainable development — 21st century. Discussions 2020: collective monograph / National University "Kyiv-Mohyla Academy" / edited by Prof. Khlobistova E.V. — Kyiv, 2020. — pp. 435-441. — Electronic edition. ISBN: 978-617-7668-22-9.
2. Polienova V.A. A linear learning path as a method of increasing the visibility of concept maps in mobile applications. // Modern problems of scientific energy supply. In 2 volumes / Materials of the XIX International. science and practice conf. young of scientists and students, Kyiv, 20–23 / Apr. 2021 - Kyiv: KPI named after Igor Sikorskyi, Polytechnic Publishing House, 2021. - Vol. 2. - 196 - 197 p. - Electronic edition. ISBN 978-966-990-027-2.
3. Polienova V.A., Fedenko V.A., Tytenko, S.V. Educational mobile application based on concept maps. Modern engineering and innovative technologies, issue 23, part 1. 12–18, 2022. doi:10.30890/2567-5273.2022-23-01-014.

Keywords: *CONCEPT MAP, EDUCATIONAL SYSTEM, CONCEPT, THESIS, EDUCATIONAL STATISTICS, INFORMATION AND EDUCATIONAL PORTAL, ONTOLOGICAL MODELING, APPLICATION FOR MOBILE DEVICES.*

8

ЗМІСТ

ВСТУП	11
1 ЗАДАЧА РОЗРОБКИ МОБІЛЬНОЇ ОНТОЛОГІЧНО-ОРІЄНТОВАНОЇ НАВЧАЛЬНОЇ СИСТЕМИ З ШИРОКИМ ВИКОРИСТАННЯМ КАРТ ПОНЯТЬ	13
1.1. Призначення та основний функціонал мобільного застосунку	13
2 АНАЛІЗ ПРОБЛЕМИ ПОБУДОВИ КАРТ ПОНЯТЬ НА МОБІЛЬНИХ ПРИСТРОЯХ	17
2.1 Огляд існуючих програмних рішень	17
2.1.1 Проєкт CoMPASS	17
2.1.2 Карти понять для вивчення математики	19
2.1.3 Мобільний застосунок Mind Map Maker	20
2.1.3 Навчальний веб-сервіс semantic-portal.net	21
2.2 Семантична модель знань як основа для побудови карт понять	26
3 ЗАСОБИ РОЗРОБКИ МОБІЛЬНОГО ЗАСТОСУНКУ	28
3.1 Середовище розробки системи	28
3.2 Мова програмування та фреймворк	29
3.3 API навчального порталу на базі карт понять	31
4 АЛГОРИТМІЧНА БАЗА МОБІЛЬНОЇ НАВЧАЛЬНОЇ СИСТЕМИ	37
4.1 Силовий алгоритм побудови карт понять на мобільних пристроях	37
4.2 Алгоритм аналізу відношень на карті понять	42
5 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ	46
5.1 Архітектура мобільного застосунку з широким застосуванням карт понять	46
5.2 Компоненти освітнього застосунку	49
5.2.1 Компонент побудови карт понять	50
5.2.2 Компоненти навігації між вершинами на карті понять	51
5.2.3 Компонент відображення тез	54
5.2.4 Компонент статистичних даних	56
6 ОПИС РОБОТИ КОРИСТУВАЧА З СИСТЕМОЮ	58
7 СТАРТАП ПРОЄКТ МОБІЛЬНОГО ЗАСТОСУНКУ З ШИРОКИМ ЗАСТОСУВАННЯМ КАРТ ПОНЯТЬ	67
7.1 Опис ідеї проєкту	67
7.2 Технологічний аудит ідеї проєкту	69
7.3 Аналіз ринкових можливостей запуску стартап-проєкту	70

	9
1.4 Розробка ринкової стратегії проекту	76
1.5 Розроблення маркетингової програми	78
ВИСНОВКИ	82
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	83
ДОДАТОК А	86
ДОДАТОК Б	97
ДОДАТОК В	107

ВСТУП

Сьогодні освітні онлайн-ресурси стали невід'ємною частиною професійної та наукової діяльності. З 2016 року кількість користувачів навчальних порталів збільшилася майже у 5 разів. У сучасному світі більшість людей віддають перевагу онлайн-навчанню, а підприємства переходять на віддалену роботу, адже це допомагає зекономити час та кошти, а також є більш безпечним. У таких умовах швидко зростає цінність освітніх веб-ресурсів із структурованою та якісною інформацією. У цей же час збільшується використання мобільних програм, що спрощує доступ до необхідної інформації в будь-який час. Завдяки потужності сучасних пристроїв люди все частіше віддають перевагу телефону навіть у порівнянні з комп'ютером або планшетами. Через це дисертація передбачає розробку програмного комплексу саме для мобільних пристроїв.

Візуалізація грає достатньо велику роль у засвоєнні знань. Таким чином, освітні веб-сервіси повинні бути адаптовані до онтологічно-орієнтованої моделі освітнього контенту, що передбачатиме формалізацію інформаційного змісту онтологічне моделювання предметної галузі та дидактичну функцію, яка забезпечує подання та візуалізацію необхідної інформації [1]. Відмінно вирішують завдання візуалізації професійної та навчальної інформації карти понять [2]. З'явившись в результаті досліджень у галузі когнітивної психології, карти понять є важливим інструментом для організації, а також представлення та обміну знаннями. Це робить карти понять дуже підходящими для використання у освітніх сервісах.

Створення карт понять включає побудову текстових вузлів, з'єднаних дугами. Вузол позначає поняття, а дуга - зв'язок між парою вузлів. Однією з переваг такого подання інформації є те, що карти зазвичай лаконічні та зрозумілі в порівнянні з текстовими повідомленнями такого самого змісту. Більш того, побудова карти понять часто здійснюється з використанням різних кольорів т

фігур, що позитивно позначається на запам'ятовуванні інформації.

Хоча, незважаючи на велику кількість переваг, використання карт понять у мобільних освітніх програмах має ряд недоліків, пов'язаних, зокрема, з малими розмірами екрану, що погіршує сприйняття карти в цілому. Особливо це відчувається на перевантажених дидактичними зв'язками картах. Таким чином будуючи карти понять, важливим є не допускати перетину ребер і вершин. Таки підвищує їх наочність та сприйняття користувачем.

Інша проблема полягає в тому, що користувач не завжди розуміє, чому утворюється зв'язок між двома поняттями. Для вирішення проблеми пропонується розробити модуль аналізу відношень між вершинами в навчальній системі. Користувач зможе натиснути на ребро та дослідити відношення між вузлами.

Також проблемою освітніх порталів на базі карт понять є відсутність інструменту для відслідковування прогресу навчання користувача. В робот запропоновано розробити модуль статистики, завдяки якому користувач мати змогу дослідити які саме поняття та як довго він вивчав, а які навпаки ще не розглянув.

Метою роботи є розробка мобільного застосунку для дистанційного навчання з широким використанням карт понять із функцією аналізу статистичних даних та аналізом відношень між поняттями, що передбачає вирішення усіх приведених вище проблем. Цільовою аудиторією додатку є люди, які зацікавленні в дистанційному навчанні, в першу чергу це школярі та студенти

1 ЗАДАЧА РОЗРОБКИ МОБІЛЬНОЇ ОНТОЛОГІЧНО-ОРІЄНТОВАНОЇ НАВЧАЛЬНОЇ СИСТЕМИ З ШИРОКИМ ВИКОРИСТАННЯМ КАРТ ПОНЯТЬ

Метою даної дипломної роботи є створення інформаційної мобільної онтологічно-орієнтованої системи з широким застосуванням карт понять. Поставленою задачею була розробка програмного забезпечення з автоматизованою побудовою карт понять, функцією аналізу статистичних даних та функцією аналізу відношень між поняттями. Створення мобільного застосунку передбачає формалізацію контенту за допомогою понятійно-тезисної моделі (ПТМ) [3].

1.1. Призначення та основний функціональні можливості мобільного застосунку

Розроблене програмне забезпечення є інформаційно-навчальним мобільним порталом та призначене для дистанційного навчання. Додаток передбачає формування онтологічно-орієнтованої моделі для формалізації інформаційного наповнення контенту. Подання та візуалізацію затребуваної навчальної інформації забезпечують онтологічне моделювання предметної області та дидактична функція. Таким мобільний застосунок дозволить користувачам вивчати потрібну предметну область в зручній формі, поліпшить ефективність навчання та засвоєння інформації.

Застосунок розроблено на базі карт понять. Карти будуються у вигляді графів, а графова візуалізація інформації підвищує її засвоєння. Вершинами

графів на карті понять є власне поняття, а ребрами – зв'язки між ними. Під час вивчення інформації про поняття, користувачу пропонуються тези. Поняттям є певний об'єкт в області знань, що представлений для вивчення, а тезою – відомість про поняття, що є його ознакою або характеристикою.

Головними задачами інформаційної системи є забезпечення доступу користувача до затребуваної інформації у вигляді карти понять, даних про поняття та відношення між ними, а також до навчальної персональної статистики. Користувач мобільного застосунку має мати доступ до таких модулів:

- Авторизація.
- Список доступних курсів.
- Інтерактивна карта понять.
- Пошук та навігація.
- Відображення інформації про поняття та тези.
- Модуль аналізу відношень між вершинами.
- Статистика за обраним курсом.

Однією з найголовніших задач застосунку була розробка навчальної статистики певної предметної області. Ця функція є дуже важливою для кращого розуміння користувачем свого прогресу та також допомагає здійснювати навігацію по карті понять. Статистика передбачає собою вивід на екран двох графіків, на яких користувач може дослідити скільки часу та на яке саме поняття він витратив, а які поняття пропустив.

Розробка програмного забезпечення включає в себе такі етапи:

- Дослідження літератури та статей за темою дипломної роботи.
- Розгляд аналогів на ринку та їх порівняння.
- Розробка макетів мобільного застосунку (рисунок 1.1) в середовищі Adobe

Photoshop 2020.

- Створення алгоритму побудови карт понять.
- Навігація по картам понять.
- Пошук по курсу, що вивчається.

- Навігація по поняттям, що вивчаються.
- Аналіз відношень між поняттями.
- Розробка панелі аналізу всіх пов'язаних понять.
- Розробка навчальної статистики за курсом, що вивчається.
- Тестування програмного продукту.



Рисунок 1.1 – Приклад макету застосунку

Основною метою дослідження є швидкий і компактний доступ до всіх навчальних курсів з пристроїв на базі операційних систем Android та iOS, що дозволяє завжди мати доступ до дистанційного навчання, навіть не маючи доступу

до комп'ютера. Такий підхід зменшує витрати часу на навчання, спрощує доступ до інформації, що дозволяє виділити більше часу на навчання та робить користувача більш зацікавленим у ньому. Доступ до навчальної статистики дозволяє краще орієнтуватися в предметній області, аналізувати свої успіхи планувати подальше навчання та слідкувати за тим, щоб жодне поняття предметної області не було пропущене.

2 АНАЛІЗ ПРОБЛЕМИ ПОБУДОВИ КАРТ ПОНЯТЬ НА МОБІЛЬНИХ ПРИСТРОЯХ

Інструмент візуалізації грає досить важливу роль у засвоєнні знань, адже він підвищує зацікавленість користувача у навчанні та підвищує асоціативність Дослідження [1] встановили, що ефективність навчання можна підвищити за допомогою використання інтерактивного графового подання структури понять відношень між ними. Через це впливає важливість створення дидактично онтології на базі ПТМ, а семантико-дидактичних карти можуть представляти онтологію.

2.1 Огляд існуючих програмних рішень

Перед початком розробки навчальної мобільної онтологічно-орієнтованої системи з широким застосуванням карт понять було проаналізовано аналогічні системи. Системи було розглянуто не лише мобільні, а й веб-портали та десктоп-версії. Аналізуючи існуючі рішення, було знайдено переваги та недоліки кожної з розглянутих систем для їх подальшого усунення в розробленому мобільному застосунку.

2.1.1 Проєкт CoMPASS

В проєкті CoMPASS авторка запропонувала використовувати карти понять у навчальних системах для зручності навігації та покращеного засвоєння знань. Було підтверджено важливість використання візуалізації контенту та інтерактивного графового зображення структури понять і відношень між ними, адже це вело до позитивного впливу на успіхи у навчанні [4].

CoMPASS представляє собою гіпертекстову систему, що допомагає школярам середньої школи вивчати природничі науки. Проект представляє учня зовнішні графічні подання у вигляді карт понять, або текстових подань, що змінюються динамічно у той час, коли учні проходять через область і приймають навігаційні рішення. Карти динамічні та інтерактивні, вони збільшуються або зменшуються у вигляді «риб'ячого ока», будуються динамічно за допомогою функції сили взаємозв'язків між поняттями через отримання поняття з бази даних. Таким чином, учні бачать будь-яке конкретне поняття стосовно багатьох інших понять, а це у свою чергу допомагає їм досягти більш повного та інтегрованого представлення наукової інформації. Рисунок 2.1 зображає карту понять на одній темі з фізики: сила та рух.



Рисунок 2.1 – Карта понять у проекті CoMPASS [4]

Серед переваг даного підходу є підписання середин ребер графа, що дозволяє учням з'ясувати які саме дидактичні зв'язки утворені між поняттями. Серед недоліків даного підходу є відсутність більш глибокого аналізу відношень, адже зазасту воно представлене лише одним словом, а також відсутності

інтерактивності, що може негативно вплинути на зацікавленість студента навчанні. Також відсутність кольорових позначень вершин та різноманітних геометричних форм зменшує наочність такого подання наукової інформації.

2.1.2 Карти понять для вивчення математики

В роботі [5] запропоновано використання карт понять у навчальних веб-системах для вивчення математики, що полегшує навігацію матеріалами та покращує їх засвоєння. Навігатор карти понять в даній системі дозволяє студентам створити зміст курсу для довільної групи суміжних понять в онтології. Наприклад, так як карта понять курсу «Теорія обчислень» має стати досить заплутаною та складною, учні могли збільшувати або зменшувати масштаб для вивчення різних рівнів абстракції матеріалу. Автор праці стверджує, що навігатор карти має стати цінним та доречним додатковим інструментом адаптивних навчальних платформ. Рисунок 2.2 зображує навігатор карти понять.

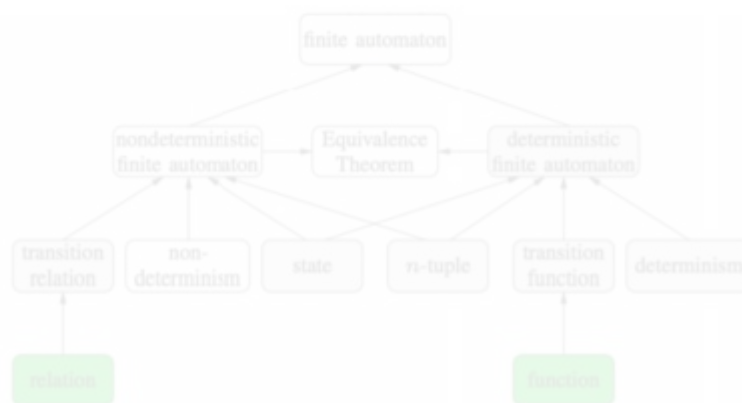


Рисунок 2.2 – Карта понять, запропонована в праці [5]

Зеленим позначено вузли, які добре вивчені студентами, сірі вузли частково зрозумілі, а білі вузли вивчені погано, або взагалі не розглядались. Перевагами

даного подання наукової інформації є присутність індивідуального підходу до кожного користувача, а також кольорові позначення вершин, адже вони забезпечують більш швидке та легке розуміння нових концепцій, покращують навігацію картою, сприяють аналізу та синтезу ідей, а також допомагають відслідковувати прогрес. До мінусів системи відноситься відсутність аналізу відношень між вершинами системи та відсутність автоматичної побудови карт під потреби студента.

2.1.3 Мобільний застосунок Mind Map Maker

Мобільний застосунок Mind Map Maker (MMM) представляє собою інструментарій, за допомогою якого можна будувати власні майнд-карти або карти понять на мобільних пристроях з операційною системою **Android**. Автором забезпечення є компанія Midomo, що стверджує, що основна задача застосунку - допомагати користувачам розкривати силу своїх ідей та зберігати власні думки на майнд-картах. Також карти можна перетворювати у презентації, поділитися їми іншими людьми, зберегти на свій девайс або синхронізувати з хмарним сховищем на будь-якому пристрої.

Користувачі застосунку стверджують, що даний застосунок є чудовим рішенням для роботи з діаграмами, має багато доступних налаштувань та приємний і легкий інтерфейс. Також користувачі відмічають позитивний вплив на робочі процеси та продуктивність. Незважаючи на маленький екран, складні карти понять легко створювати та редагувати. Застосунок регулярно оновлює функціонал, аби покращити користувацький досвід. Однак, майже все, що створюють користувачі - це документи, презентації, сценарії, відео тощо, а от автоматична побудова карт понять у застосунку недоступна. На рисунку 2.3 зображено приклад майнд-карти, розробленої в мобільному застосунку Mind Map Maker.



Рисунок 2.3 – Приклад майнд-карти застосунку MMM

Перевагами даного програмного забезпечення є зручність використання на мобільних пристроях, легкий та зрозумілий інтерфейс, можливість застосовувати різні форми та кольори для подання інформації. Серед недоліків - відсутність функції автоматичної побудови карти понять за моделлю контенту, відсутність інтерактивності. Застосунок не є навчальним, тому аналіз відношень між поняттями та статистичні дані тут відсутні, хоча за бажанням дуги можна підписати вручну.

2.1.3 Навчальний веб-сервіс semantic-portal.net

Іншим прикладом є веб-портал для дистанційної освіти <http://semantic-portal.net/> [6]. Ця веб-система використовує карти понять для онтологічного моделювання науково-освітнього контенту, що забезпечує

візуалізацію знань у певних предметних областях. На рисунку 2.4. зображено приклад карти понять з даного освітнього сервісу.

Представлена на рисунку карта побудована завдяки відношенням «part of», на ній використано різноманітні кольори, що позитивно впливає на запам'ятовування. Також, такі карти є інтерактивними - вузли та дуги можна рухати й потім за секунду карта повертається в свій початковий стан, що збільшує інтерес користувача до навчання. Як і в роботі [5], карти понять, побудовані на даному освітньому порталі мають функцію збільшення або зменшення масштабу, що поліпшує сприйняття та спрощує навігацію.



Рисунок 2.4 – Карта понять в навчальній веб-системі semantic-portal.net

При натисканні на поняття воно змінює свій колір на зелений, а це означає, що поняття вже вивчалось, а на допоміжній панелі справа (рисунок 2.5)

відображається список тез, які користувач використовує для вивчення відкритої предметної області.

Веб-система автоматично аналізує чи наявні інші поняття у тезах, що вивчаються, по ним також можна перейти з метою вивчення, клацнувши по слову. Більш того, під інформацією про певне поняття наявні допоміжні теги, що роблять навігацію по курсу більш зручною.

Важливим елементом навігації по карті понять є пошук по поняттям. Користувач може ввести назву, а система автоматично виділить його зеленим збільшить масштаб карти, наблизившись до шуканого вузла. Наявність зручного пошуку по вузлам дає змогу користувачу швидко знайти потрібне поняття та його нащадків, що є дуже важливим при вивченні більш складних та навантажених систем (рисунок 2.6).

Dart

Dart — Is a client-optimized programming language for apps on multiple platforms. It is developed by Google and is used to build mobile, desktop, server, and web applications. %

- Is an object-oriented, class-based, garbage-collected language with C-style syntax. It supports interfaces, mixins, abstract classes, reified generics, and type inference. %*
- Can compile to either native code or JavaScript. %*
- Although Dart is strongly typed, type annotations are optional because Dart can infer types. %*

Variables

Types

Functions

Classes

Рисунок 2.5 – Опис поняття в навчальній веб-системі semantic-portal.net

Не дивлячись на численні переваги представленої освітньої системи, веб-портал має певні недоліки. Такий підхід візуалізації інформації також

передбачає можливість присутності занадто масивних та перевантажених карт понять із дидактичними зв'язками, що у свою чергу може ускладнити користувацький досвід та зменшити наочність представлених карт.



Рисунок 2.6 – Приклад складнішої карти понять з освітнього порталу

<http://semantic-portal.net/>

Карты понять портала, що охоплюють велику предметну область та побудовані за допомогою дидактичних зв'язків порядку вивчення (рисунок 2.7) доволі часто мають велику кількість вузлів та ребер, що плутаються між собою. Це сильно ускладнює візуалізацію та розуміння користувача, адже в такій карті майже неможливо розібратися. До того ж, такі масивні карти є не оптимізованими.

та потребують додаткових обчислювальних витрат.

Наступним недоліком розглянутої інформаційної системи є відсутність аналізу зв'язків між поняттями. Така функція на навчальному порталі є дуже бажаною, бо це допомагає кінцевому користувачу краще поглибитися в предметну область та зрозуміти чому двома поняттями встановлено дидактичний зв'язок, що саме він означає. Для розробки такого функціоналу можна провести аналіз усіх тез, що містять обидва поняття, порівняти отримані дані та визначити властивості.

Ще одним недоліком є відсутність навчальної статистики, адже користувач не може дослідити скільки і яке поняття вивчалось, хоча й дана інформація присутня в логах порталу.



Рисунок 2.7 – Приклад перевантаженої карти понять з дидактичними зв'язків на порталі

В результаті аналізу існуючих рішень було досліджено переваги та недоліки навчальних електронних систем для дистанційної освіти на базі карт понять

Завданням дисертації є розробка мобільної навчальної онтологічно-орієнтованої системи, у якій буде вирішено розглянуті недоліки, буде імплементовано навчальну статистику та аналіз відношень між вузлами, що дозволить користувачу краще вивчити предметну область та слідкувати за своїм прогресом, спростити навігацію та додасть зацікавленості у навчанні.

2.2 Семантична модель знань як основа для побудови карт понять

Оскільки метою магістерської дисертації є розробка мобільного застосунку з широким застосуванням карт понять, було використано семантичну понятійно-тезисну модель знань (ПТМ) [3]. Таким чином, карти понять будуть представляти орієнтований ациклічний граф, поняття будуть означати вершини графа, а відношення між поняттями позначатимуть ребра. Поняттям є предмет обговорення, об'єкт предметної області, про який навчальна інформація містить знання. Знання про поняття представлятимуть тези, що є множинами відомостей про поняття в моделі, що розглядається.

Поняття вказують на певний конкретний об'єкт у предметній області. Наприклад, в курсі по вивченню мови програмування «Dart» поняттями є «клас», «змінна», «тип даних».

Тезами у курсі є відомості про поняття, або інші твердження, які є істиною для поняття, що розглядається. Наприклад, тезою для поняття «Dart» є «Dart - мова для структурованого програмування, що використовується для написання програмного забезпечення для декількох платформ». Тез для одного поняття може бути кілька.

Наріжним каменем семантичних систем є тріади [2] з такою структурою: перше поняття – відношення – друге поняття.

У наведеному на рисунку 2.8 прикладі «Dart» та «Клас» встановлено

відношення «part of», що означає «є частиною». Це означає, що поняття «клас» є частиною поняття «Dart».



Рисунок 2.8 – Схема тріади у ПТМ

Серед переваг тріадного підходу до формалізації знань є здатність системи до оперування поняттями предметної області та встановлення між ними певних відношень. Недоліком тріадного підходу є великі обчислювальні витрати, що відбуваються при формуванні цілісної інформаційної системи. Під час виконання дисертації цей недолік буде вирішено.

3 ЗАСОБИ РОЗРОБКИ МОБІЛЬНОГО ЗАСТОСУНКУ

3.1 Середовище розробки системи

Android Studio (AS) була обрана для розробки освітньої мобільної системи. AS представляє собою інтегроване середовище розробки в першу першу для операційної системи Android, але його також можна використовувати для розробки кросплатформених проєктів. Це програмне забезпечення було випущено Google у 2014 році. AS було побудовано на коді IntelliJ IDEA Community Edition.

Це середовище ідеально підійшло для поставленого завдання, через свою адаптованість до виконання типових завдань, що виникають під час розробки програм для операційної системи Android. Дане середовище також дозволяє легко переглядати різні стани мобільного додатка під час розробки та швидко його тестувати.

Середовище Android Studio призначене як для невеликих груп розробників мобільних додатків, так і для великих міжнародних організацій, які використовують GIT або інші системи контролю версій. Досвідчені розробники можуть обрати інструменти, які найкраще підходять для великих проєктів. Android Studio — універсальне середовище розробки, оскільки дозволяє оптимізувати майбутні програми для роботи не тільки на мобільних пристроях, а й на планшетах та портативних ПК. Основними перевагами розглянутого середовища розробки є:

- Наявність зручних редакторів коду.
- Можливість підтримки роботи кількома мовами програмування.
- Можливість перевірити коректність програми.
- Можливість рефакторингу готового коду.
- Наявність великої бібліотеки із готовими шаблонами.

- Можливість перевірити готову програму на наявність помилок.
- Надання детальних спеціальних посібників з використання AS.

Для розробки інформаційної програми використовувалася розподілена система контролю версій Git. Систему було розроблено в 2005 році і вона є проєктом розробки з активною підтримкою і відкритим вихідним кодом. Git часто використовують досвідчені розробники для контролю версії під час розробки програмного забезпечення. Ця система використовується багатьма професійними розробниками програмного забезпечення. Було підтверджено, що вона працює під управлінням різних операційних систем і доступна в багатьох інтегрованих середовищах розробки. Розробка за допомогою Git спрямована на забезпечення високої ефективності, безпеки та гнучкості для розподілених систем.

Також використовувався емулятор мобільного пристрою Pixel 4a із встановленим Android Red Velvet Cake (11.0). Версія Android 8.0 є мінімальною вимогою для коректної роботи розробленої програми.

3.2 Мова програмування та фреймворк

Мобільне програмне забезпечення написано за мовою Dart. Dart є мовою програмування, яку створила компанія Google у 2011 році. Спочатку за мету вони ставили вирішення фундаментальних проблем Javascript. Розробники мови поставили перед собою такі цілі:

- Створення мови, подібної до вже існуючих мов, задля зниження вхідного бар'єру.
- Розробка гнучкої та структурованої мови програмування.
- Досягнення високої продуктивності розроблених систем.

Мова Dart є потужним інструментом для створення як клієнтської, так і серверної частини забезпечення. За допомогою цієї мови програмісти мають можливість розробляти безліч рішень: не тільки мобільні додатки, а й

веб-системи, такі як html-сторінки та ігри, користуючись DOM або WebGL/Canvas.

Завдяки Dart є можливість створювати консольні ігри текстового формату, програми для операційної системи Windows та MacOS. Серйозним плюсом цієї мови програмування є можливість розробки кросплатформених мобільних додатків. Однак, незважаючи на численні переваги, мова не є чимось універсальним. Наприклад, на ній не має можливості написати операційну систему, або мати високу продуктивність для написання ігор, в чому краща C++. Dart не підходить для обробки великих обсягів наукових даних.

Dart — це мова програмування, яка спочатку використовувалася всередині Google для створення веб-додатків, серверів і мобільних додатків. Dart існує з 2011 року, але вона не привертала уваги за межами Google. Dart компілює вихідний код подібно до інших мов програмування, як-от JavaScript, але окремий пакет SDK Dart постачається разом із віртуальною машиною Dart. На відміну від інших мов програмування, Dart має власний менеджер пакетів під назвою Pub. Dart почав привертати більше уваги в 2017 році, коли Google офіційно анонсувала бета-версію Flutter для розробки кросплатформних мобільних додатків. Відтоді популярність Dart різко зросла. На даний момент розробники мобільних додатків прагнуть перейти на Flutter, але вони повинні навчитися дарту, щоб почати. Однак розробники, які мають досвід роботи з об'єктно-орієнтованими мовами, можуть легко вивчити Dart.

Мобільний застосунок розроблявся саме за допомогою фреймворку Flutter. Flutter — це безкоштовна платформа мобільного інтерфейсу користувача з відкритим вихідним кодом, створена Google і випущена в травні 2017 року [17]. Вона дозволяє створювати нативну мобільну програму лише з однією кодовою базою. Це означає, що програмісти можуть використовувати одну мову програмування та одну кодову базу для створення двох різних програм (для iOS та Android). Флаттер складається з двох важливих частин:

1. SDK (набір для розробки програмного забезпечення): набір інструментів, які допомагають розробляти програми. Це включає в себе інструменти для

компіляції написаного коду в нативний машинний код (код для iOS і Android).

2. Фреймворк (бібліотека інтерфейсу користувача на основі віджетів): набір багаторазово використовуваних елементів інтерфейсу користувача (кнопки текстових введів, повзунків тощо), які можна персоналізувати для власних потреб.

Таким чином, Flutter — це сучасний, гнучкий і потужний фреймворк. З ним набагато простіше створювати мобільні застосунки та тестувати їх.

3.3 API навчального порталу на базі карт понять

Навчальне онтологічно-орієнтоване програмне забезпечення з широким застосуванням карт понять було розроблено, використовуючи дані, взяті з веб-сервісу <http://semantic-portal.net/>. Було використано відкритий API (Application Programming Interface - набір HTTP-запитів та структур HTTP-відповідей формату JSON) (рисунок 3.1), що дає змогу отримувати дані про поняття, їхні тези та зв'язки між ними, знаходити поняття в тезах інших понять тощо. Також за допомогою API було отримано доступ до логів користувачів, що дозволило зібрати статистичні дані для мобільного застосунку [16]. Для цієї функції було розроблено авторизацію, адже додаток є персоніфікованим. Веб-портал зберігає онтологічно-формалізовані дані у понятійно-тезисній моделі, до них може отримати доступ розробник за рішенням власника порталу.

API порталу складається з декількох розділів. «**Branch**» надає доступ до даних про курси та їхні складові, а саме поняття, тези, тип відношень між ними. Розділ було використано в роботі для побудови карт понять певної предметної області в розробленому застосунку.

Розділ «**Page**» зберігає дані про сторінки порталу, такі як заголовки, вхождення понять в них, картини (за їх наявності) та інші. Дану інформацію

використано для ініціалізації початкових понять під час побудови карт понять.

Розділ “Log” зберігає дані про логи користувача, а саме містить в собі об’єкти дій користувача за одну сесію. В цих об’єктах є дані про тип події, її тривалість та точний час початку й кінця події. Також за допомогою спеціального ендпоінта можна залогувати дії користувача, що й було використано у мобільному застосунку. Оскільки API єдине і для веб-версії і для мобільної системи - дії зберігаються й відображаються на обох порталах.

Для користування логам користувача його спершу потрібно створити. Для цього було використано розділ API “Auth”, що дозволяє авторизуватися у додатку й зберегти токен сесії і ідентифікаційний номер користувача в середині додатку. Для реєстрації користувач мусить перейти до веб-версії portalu.



Рисунок 3.1 – Приклад API з освітнього portalu <http://semantic-portal.net/>

Розділ «**Concept**» в API зберігає дані про всі доступні portalu поняття. На рисунку 3.2 зображено приклад API для мови програмування Dart. Розробник застосунку може за запитом отримати дані про одне чи кілька понять, вказавши їхній id.


```
"id": "2607",  
"concept": "Dart",  
"domain": "dart-tour",  
"class": "tech",  
"rating": "0",  
"forms": "",  
"searchConceptsCalled": "-1",  
"parsed": null,  
"definition": null,  
"getDefinitionCalled": "-1",  
"isAspect": "0",  
"aspectOf": null
```

Рисунок 3.2 – Приклад API для concept

Таким чином, є можливість отримати інформацію про тези, що відносяться до обраного поняття, та побачити, з якими іншими концепціями вони пов'язані дидактичними зв'язками. Таким чином, можна отримати доступ до наступної інформації про поняття: ідентифікаційний номер, клас, домен, назва, форма. Ця інформація можна використовувати його для доступу до даних про поняття та побудови карти понять й аналізу відношень на карті. За допомогою форм певного поняття можна виділяти різними кольорами всі входження поняття для зручного вивчення й навігації по тезам.

З використанням розділу API "C in T", який означає "поняття в тезах", розроблено модуль аналізу взаємозв'язків між вершинами карти понять. Даний розділ містить у собі дані про знаходження того чи іншого поняття у тезах інших понять та навпаки. На рис. 3.3 показано API розділу C in T для концепції Dart.

За допомогою отриманих даних можливо виявити, що поняття Dart зустрічається в інших 16 тезах, а також знайти ідентифікатор тези і назву концепції, на до якої відноситься теза. Також можна отримати інформацію про те,

які концепції з'являються у тезах Dart та які конкретно тези їх містять.

```
{
  "thisConcept": [ - ], // 16 items
  "otherConcepts": [
    {
      "thesis_id": "785408",
      "concept_id": "2634",
      "count": "1"
    },
    { - }, // 3 items
    { - } // 3 items
  ]
}
```

Рисунок 3.3 – Розділ з API “C in T”

У розділі «thesis» містяться дані про тези. Розробник може отримати доступ до всіх тез, що зв'язані з бажаним поняттям, або до напряму тез та інформації про них, використовуючи ідентифікатор. Рисунок 3.4 зображує приклад **JSONy** для довільної тези, заданої по id. Таким чином, отримавши тезу за її ідентифікатором можна отримати дані про її зміст, поняття, до якого вона має відношення, до класу, оформлення її вигляду всередині системи, а також до поля “has Concept”, яке вказує на згадування у тезі інших понять, якщо такі є. Клас тези потрібен для оформлення тези всередині системи, наприклад, якщо клас є кодом, то таку тезу буде оформлено у вигляді сніппету програмного коду.

У розділі “log” містяться дані про дії користувача в середині порталу. Розробник може отримати доступ до всіх логів порталу, до користувачів, що зареєстровані в системі, до логів за певним обраним курсом та до логів конкретного користувача за його ідентифікатором. Більш того, розробник має змогу отримати доступ до логів користувача у потрібний йому проміжок часу.

Приклад використання API та доступних ендпоінтів зображено на рисунку 3.5.

```
{
  "id": "785453",
  "concept_id": "2518",
  "view": "dart-four-variables",
  "thesis": "Even variables with numeric types are initially null, because numbers -- like everything else in Dart -- are objects",
  "class": "essence",
  "rating": "0",
  "count": null,
  "parsed": null,
  "hasConcept": "0",
  "to_concept_id": null,
  "to_thesis_id": null,
  "to_thesis_caption": null
}
```

Рисунок 3.4 – Приклад API для тези

Log API

```
/log/api/ - manual
/log/api/log - Logs
/log/api/log/from/{time} - Logs from {time} (timestamp)
/log/api/log/between/{time1}/{time2} - Logs from {time1} to {time2} (timestamp)

/log/api/user - users
/log/api/user/{id}/details - user details by user {id}
/log/api/user/{id} - user logs by user {id}
/log/api/user/{id}/from/{time} - user logs by user {id} from {time}
/log/api/user/{id}/between/{time1}/{time2} - user logs by user {id} between times {time1} and {time2}

/log/api/course - courses
/log/api/course/{course} - course details, its users & branches

Raw Logger
/log/api/log/concept - POST 'log' post parameter as JSON:
{
  "userId": int
  "contentId": concept_id
  "time": timestamp (first user contact in current session)
  "seconds": int (real seconds spent on this concept during current session)
  "lastTime": timestamp (the last user contact in current session)
}

Multiple concept records during one session are allowed.
```

Рисунок 3.5 - Доступні ендпоїнти логування

Важливою функцією є можливість зберегти лог користувача. На сервер розробник може відправити яку саме дію, коли та скільки секунд відтворював користувач освітнього порталу. Дана функція є вкрай важливою для збору та аналізу статистики всередині додатку. Прикладом використання такого

функціоналу є можливість виявити які поняття вивчав юзер та порахувати сумарний час навчання. Отримані дані можна відсортувати та скласти статистику, а потім оформити її у вигляді зручних діаграм для подальшого ознайомлення та аналізу успіхів.

Дані, взяті з API наукового порталу, використовуються для побудови власних карт понять, аналізу відношень між поняттями в системі, а також для модуля аналізу статистичних даних.

4 АЛГОРИТМІЧНА БАЗА МОБІЛЬНОЇ НАВЧАЛЬНОЇ СИСТЕМИ

4.1 Силовий алгоритм побудови карт понять на мобільних пристроях

При розробці онтологічно-орієнтованого мобільного програмного забезпечення на основі карт понять було виявлено низку проблем. Перша проблема полягає в тому, що використовувати карти на мобільних девайсах складно через обмежений і відносно невеликий розмір екранів пристроїв [7]. Наступною проблемою є відсутність готових рішень для побудови графів на мов програмування Dart. Наприклад, у мові програмування JavaScript є бібліотека D3.js, яка дозволяє швидко створювати графи на основі передаваних даних. В даній магістерській дисертації одним з завдань було розробити алгоритм побудови графів. Алгоритм повинен побудувати граф таким чином, щоб вершини та ребра не перетиналися та не заважали один одному, щоб підвищити наочність карт понять та полегшити сприйняття.

Існує доволі багато алгоритмів побудови графів, що могли б підійти для побудови складних та комплексних карт понять [8]. Алгоритми було поділено за способом їх реалізації. Прикладами популярних алгоритмів для побудови графів є:

- алгоритми дугових діаграм;
- алгоритми спектрального компонування;
- методи силової візуалізації графів;
- алгоритм кругового макетування;
- алгоритми ортогональної розмітки.

В якості основи розробленого алгоритму побудови графа було обрано статичний алгоритм візуалізації. Мета цих методів - розмістити вершини графа просторі так, щоб усі ребра були приблизно однакової довжини, при цьому

кількість перетинів ребер повинна бути мінімальною. Для досягнення цієї мети на ребра та вершини діють сили, засновані на їхньому взаємному розташуванні [9].

Існує кілька потужних алгоритмів для силового створення графів. Одним з них - це алгоритм Ідеса [10]. Основна ідея цього алгоритму полягає у моделюванні мережі «намагнічених» ребер та вершин, представлених певною пружиною. Графічний метод Ідеса - перший алгоритм, що використовує відштовхування та силу тяжіння. Сила тяжіння f_a прикладена до вершин графа, безпосередньо з'єднаних ребрами. До непрямо пов'язаних вершин застосовується відштовхування f_r .

Сили відштовхування та тяжіння у алгоритмі Ідеса визначають так:

$$f_a(i, j) = C_a \log \log \frac{d(i, j)}{d_0} \quad (1)$$

$$f_r(i, j) = C_r \frac{1}{d(i, j)^2} \quad (2)$$

Де d_0 – ідеальна довжина ребра, d – ідеальна відстань між вузлами i та j , а C_a та C_r - константи. Головною метою представленого алгоритму є знаходження місця розташування нульових сил для всіх вершин системи для досягнення стану рівноваги у пружинній системі.

Наступний силовий алгоритм для побудови графів – алгоритм Фрухтермана-Рейнгольда [11]. Базою для даного алгоритму став алгоритм Ідеса який був представлений вище. Таким самим чином як і в алгоритмі Ідеса, в алгоритмі Фрухтермана-Рейнгольда використовуються сили f_a тяжіння та відштовхування f_r , що визначаються за такими формулами:

$$f_a(i, j) = \frac{d^2}{k} \quad (3)$$

$$f_r(i, j) = \frac{k^2}{d} \quad (4)$$

Де d – це відстань між вершинами, k – постійна ідеальної попарної відстані.

Для f_a , k можна записати як $a \times \sqrt{\frac{(W \times H)}{n}}$, а для f_r як $r \times \sqrt{\frac{(W \times H)}{n}}$. Де H – висота

фрейму W – ширина фрейму, n – кількість вершин графа, a – константа для

помножувача відштовхування, α — константа для помножувача притягання.

Поданий алгоритм працює ітеративно. На кожній ітерації після обчислення сил на вузлах ми переміщуємо вузли разом. Під час кожної ітерації алгоритм додає параметр «зміщення» координат вузла для зберігання зміщення положення. Ї початку кожної ітерації обчислюються значення відштовхування для кожної вершини графа і значення тяжіння для кожної вершини ребра. Наступним кроком є оновлення положення вершини з використанням значення translate. На початку роботи вершини застрягали в локальних мінімумах через занадто великі значення параметра «Переміщення», тому алгоритм був доповнений. Для оптимізації введено температурну характеристику t . Температура керує максимальним значенням параметра "зміщення" за допомогою алгоритму імітації відпалу. Тому наприкінці кожної ітерації значення температури зменшується.

Алгоритм, заснований на алгоритмі Фрухтермана-Рейнгольда, був розроблений для побудови карт понять всередині розробленого програмного забезпечення. Візуалізація графа, отримана під час виконання цього алгоритму найкраще відповідає поставленій задачі. Крім того, класичний алгоритм побудови графа, алгоритм Фрухтермана-Рейнгольда, дає розробникам можливість експериментувати та доповнювати вихідні алгоритми.

На початку побудови графа за цим алгоритмом ініціалізуються сторони каркасу та радіуси відштовхування. Також ініціалізуються ідеальні попарні константи відстані. Після цього починається робота з основного циклу. Вона триває протягом заданої розробником кількості ітерацій.

Сили відштовхування також розраховуються для кожної вершини. Алгоритм починається з перевірки відстані від однієї вершини до іншої. Якщо відстань між вершинами менша за максимальний радіус дії сили відштовхування, обчислюються взаємодії вершин. Якщо відстань між вершинами надто мала (менша за параметр "hot Distance"), параметр вершини "is Hot" встановлюється рівним істині. Це збільшує відштовхування між вершинами вектор зсуву. Щоб

знайти цей вектор, нам потрібно знайти різницевий вектор двох вершин. Потім вектор форматується одиниці і множиться на значення, задане силою відштовхування. Отриманий вектор додається до параметра вершини “displacement”. Це вектор, що вказує напрямок зміщення вершини.

Потім обчислюється сила, з якою дзига відштовхується від краю. Точка прикладення сили відштовхування знаходиться посередині ребра. Якщо вершина не міститься в ребрі, обчислюється центральне положення ребра, і вершина відштовхується, як зазначено вище.

Наступним кроком є розрахунок гравітаційної сили. Різний вектор обчислюється кожної пари вершин, з'єднаних рубом. Потім перетворіть вектор на одиниці і помножте значення, дане гравітаційної силою. Цей вектор додається до вектора, заданого для кожної вершини параметром “displacement”.

Останнім кроком у алгоритмічній операції є коригування положення кожної вершини у графі. Це залежить від вектора “displacement”. Значення векторного модуля “displacement” може бути занадто великим. І тут це значення форматується першим і викликається функція під назвою displacement(). Ця функція нормалізує значення усунення з урахуванням параметра “is Hot”, радіуса тяжіння та інших параметрів. Параметр “position” замінює значення для врахування максимального розміру, доступного в інтерфейсі. Це дозволить уникнути помилок у майбутньому. Таким чином, алгоритм не завершить виконання доти, доки не буде виконано щодну ітерацію і значення змінної не зрівняється зі значенням раніше заданого максимального числа ітерацій.

Алгоритм змінено для оптимізації та запобігання застряганню вершин у локальних мінімумах. На рисунку 4.1 показаний результат зміни кожної вершини та встановлення початкового положення.

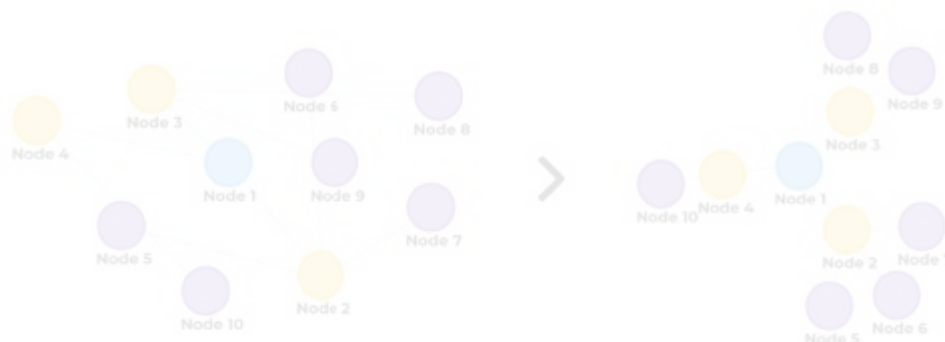


Рисунок 4.1 – Порівняння початкової позиції вершин у оригінальному та модифікованому силових алгоритмах

Першим моментом, що відрізняє розроблений алгоритм від оригіналу, є початкове положення вершин. В алгоритмі Фрухтермана-Рейнгольда вершини спочатку присвоюються випадкові позиції в певному кадрі. Через це кількість ітерацій, що виконуються для побудови одного і того ж графа, може кожного разу сильно відрізнитися, тому вершини з більшою ймовірністю застрягають на локальних мінімумах, а ребра з більшою ймовірністю перетинаються. Алгоритм виконує більше ітерацій, збільшуючи час виконання та обчислювальні витрати.

У модифікованій версії алгоритму початкові положення вершин не випадкові. Щоб скоротити час, необхідний для побудови дерева, вирішив побудувати схематичне дерево перед запуском потужного алгоритму. Для цього використовувався рекурсивний обхід деревини. Таким чином, безліч вершин і ребер $G = (V, E)$ перетворюється на безліч вузлів $G = (N)$. Кожен вузол зберігає інформацію про вузли-нащадки, тому для побудови дерев можна використовувати методи сортування. Такий підхід спрямований на оптимізацію, скорочує час побудови графа та дозволяє отримувати однакові графи при кожному запуску алгоритму. Оригінальний підхід щоразу створював зовсім новий візуальний граф. Схематична діаграма сил, що діють на вершини під час виконання алгоритму

показано на рисунку 4.2. Точка відштовхування позначена червоною точкою посередині ребра.

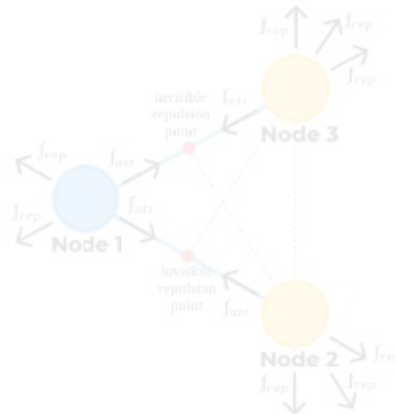


Рисунок 4.2 – Схематичне представлення сили відштовхування f_{rep} та сили притягання f_{atr} в модифікованому алгоритмі

Під час розробки алгоритму було помічено, що в якийсь момент вершина зазнала дії аналогічної сили з боку двох інших вершин з протилежних сторін, і це становище на ребрі між двома вершинами було локальним мінімумом. Для вирішення цієї проблеми було додано ще одну силу відштовхування, що діє від центру до вершини кожного ребра. Це трохи збільшує час виконання однієї ітерації, але зменшує загальну кількість ітерацій, що призводить до покращення рівноваги в системі.

4.2 Алгоритм аналізу відношень на карті понять

Можливість аналізу відношень між поняттями дозволяє користувачам краще зрозуміти зв'язки між поняттями та краще вивчити навчальний матеріал. Цей

алгоритм використовує “C in T” (концепти в тезах) - елемент інформації про концепції з API наукового порталу як вхідні дані. Цей пункт містить два підпункти: “this Concept” і “other Concepts”. У цих підрозділах зберігається інформація про активне поняття в тезі про інші поняття та інші поняття в активній тезі. На рисунку 4.3 показано схему модуля «Аналіз відношень».

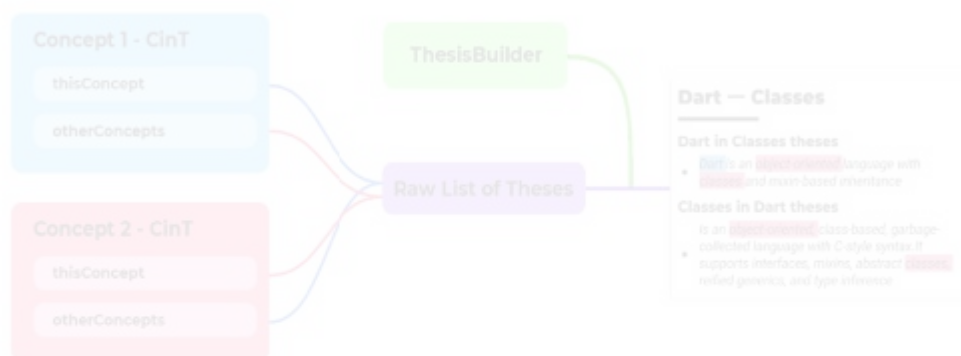


Рисунок 4.3 – Схема роботи модуля аналізу відношень на карті понять

Кожне відношення містить мінімум два поняття. Щоб запустити модуль аналізу, потрібно вибрати ребро, яке користувач хоче проаналізувати. Це може бути дидактичний зв'язок чи зв'язок part-of. Головне, що він складається із двох понять. Після вибору ребра з пристрою користувача на сервер API освітнього порталу надсилається запит інформації про поняття у тезах. У відповідь систем отримує два JSON-файли і конвертує їх в екземпляри моделі, об'єкти класу ConceptInTheses. Після цього проводиться порівняння тез у розділах цього концепту (thisConcept) і другого іншого концепту (otherConcept) і навпаки. Збігаються номери дисертацій заносяться до одного списку номерів.

Цей список знову надсилається на сервер і повертається список документів. Після отримання відповіді необроблені дані з сервера перетворюються на список екземплярів класу «Тези». Цей список переписується до асоціативного списку, д

тези це ключі, а значення — номери відповідних понять.

Отриманий список буде надіслано до класу BottomPanel для візуалізації, але спочатку кожна теза в цьому списку має пройти через клас ThesesBuilder. Залежно від типу тез буде вибрано відповідний клас для їхнього відображення та верстки, а поняття, до яких тези мають відношення, будуть знайдені та виділені у текст різними кольорами, що відповідають кольорам вершин. На рисунку 4.4 показано блок-схему алгоритму пошуку входжень понять у тези.

Алгоритм починається з розподілу тез на списки окремих слів. Кожне поняття має перелік своїх форм. Концептуальні форми – це різні варіанти слова, які відповідають даному поняттю. До кожного слова у тезах проводиться порівняння з кожною формою вихідного поняття. Форма поняття також розбивається на список окремих слів, після чого кожне слово у цій формі порівнюється з наступним словом у тезі. Це необхідно для того, щоб алгоритм міг виявляти концептуальні форми з великою кількістю слів у тезах.

Якщо форма обраного поняття відповідає слову у тезі, то конкретний елемент зі стилем, що відповідає першому концепту, додається до нового списку слів, який потім передається до класу, що відповідає візуалізації тези. Аналогічна дія відбувається і по відношенню до другого поняття. Якщо для слова у його тезах не знайдено відповідної концептуальної форми, до нового списку слів додається новий елемент без стилю. Після того, як останнє слово тези позначено білою галочкою, список, що згенерований, передається аналізатору типу тез, а потім новий список передається у відповідний клас рендерингу та верстки тез.



Рисунок 4.4 – Алгоритм пошуку входження понять у тези

Представлений модуль використовується як у окремій панелі аналізу дидактичних зв'язків, і у основній карті понять.

5 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ

5.1 Архітектура мобільного застосунку с широким застосуванням карт понять

Перед до розробкою програмної системи важливо спершу побудувати архітектуру. Це включає в себе найважливіші рішення щодо організації програмного продукту. Спершу досліджується структура та поведінка системи врівноважується комплекс її потреб. Перед розробкою онтологічно-орієнтованог мобільного застосунку було розроблено діаграму прецедентів (рисунок 5.1) та діаграму класів (рисунок 5.2).



Рисунок 5.1 – Діаграма прецедентів

Діаграма прецедентів (рисунок 5.1) показує та визначає контекст та вимоги

всієї системи чи важливих частин системи. Розробники можуть використовувати одну діаграму прецедентів використання для моделювання складної системи а створювати множину діаграм для моделювання компонентів системи. Попередня побудова діаграми зазвичай створюються на початку проекту, і на неї посилаються у процесі розробки. Діаграми випадків корисні у таких ситуаціях:

- Перед початком проекту, щоб змодельовати бізнес і переконатися, що всі учасники проекту мають загальне уявлення про співробітників, клієнтів та бізнес-діяльності.
- Збираючи вимоги, щоб представити системні вимоги та показати іншим, що система має робити.
- На етапі аналізу та проектування можна використовувати юз кейси та акторів на діаграмі прецедентів, щоб визначити класи, необхідні для системи.
- На етапі тестування діаграми для визначення тестів для системи.

Діаграма класів (рисунок 5.2) — це графічна нотація, яка використовується для побудови та візуалізації об'єктно-орієнтованих систем. Діаграма класів в уніфікованій мові моделювання (UML) — це тип статичної структурної діаграми, яка описує структуру системи, показуючи системні:

- класи,
- їхні атрибути,
- операції (або методи),
- і зв'язки між об'єктами.

Таким чином, з представлених діаграм впливає, що основними функціями та модулями системи є: авторизація, доступ до курсів, карта понять, пошук по карті та навчальна статистика.

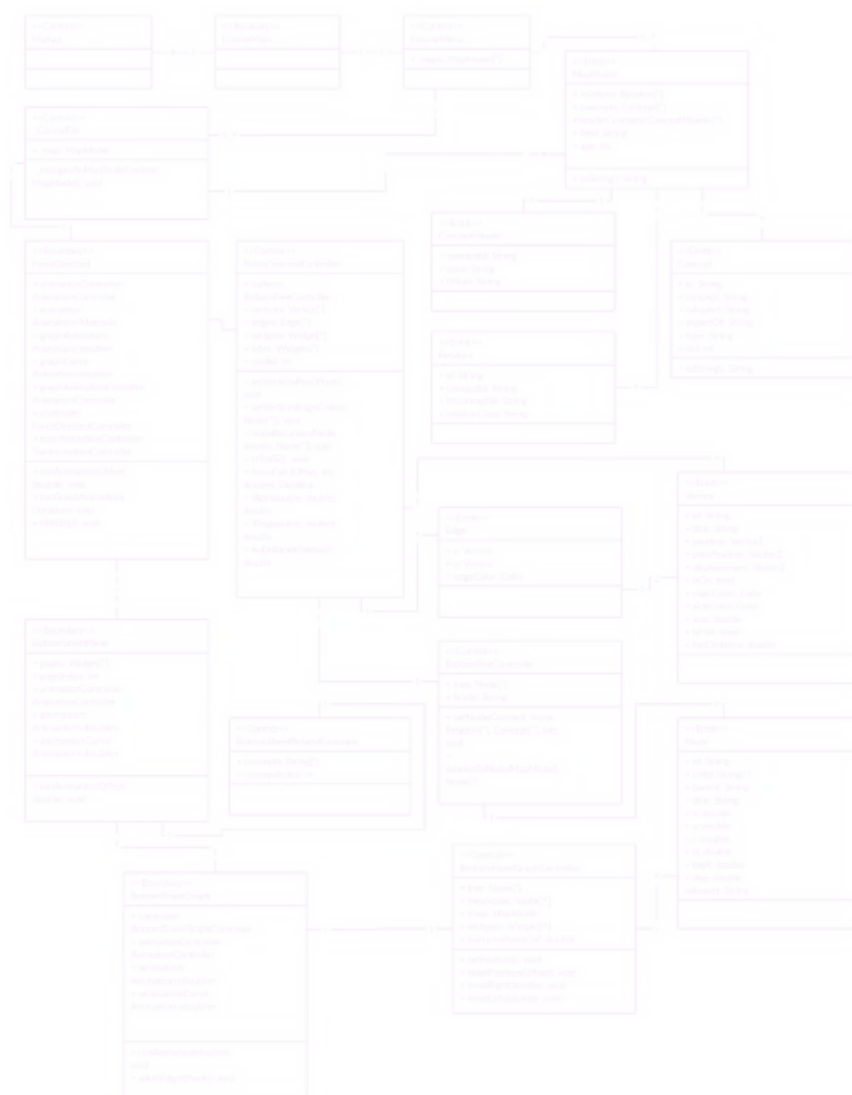


Рисунок 5.2 – Діаграма класів

Головними класами системи є сутності Vertice та Edge, що використовуються для побудови карт понять, а сама побудова відбувається у класі ForceDirected.

5.2 Компоненти освітнього застосунку

Застосунок було розроблено на базі архітектури MVC (Model View Controller). Структура мобільної системи містить два типи сутностей: моделі для будування графу та моделі, що представляють об'єкти отримані з API навчального веб-порталу semantic-portal.net. Схему структури наявних в мобільному застосунку компонентів зображено на рисунку 5.3.

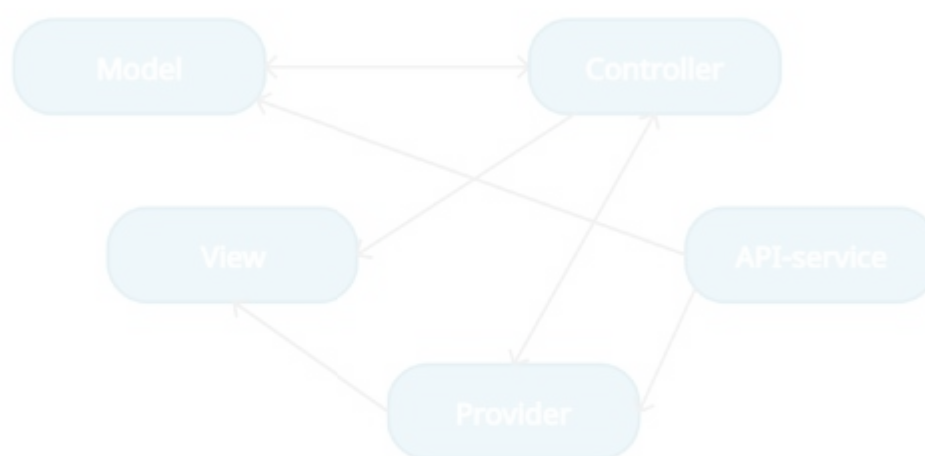


Рисунок 5.3 – Структура компонентів і їх зв'язків в мобільному застосунку

Контролер (Controller) в розробленій системі відповідає за побудову карт понять та навігацію в застосунку. Представлення (View) видимі користувачу компоненти під час користування застосунком (верстку). Важливим компонентом застосунку є Provider, задача якого зберігати тимчасові дані та змінні. Також він містить програмні функції, до яких можна доступитися з будь-якої точки проекту.

5.2.1 Компонент побудови карт понять

Моделі, використані в системі, були розроблені на базі сутностей, що отримуються з API освітнього веб-порталу [6]. Найголовніші сутності в моделі - це Thesis та Concept (рисунок 5.4). Представлені сутності є основними елементами ПТМ [11].



Рисунок 5.4 – Структура об'єктів сутностей системи

Клас Node потрібен для побудови оптимізованого початкового розташування вершин на карті понять, а також навігації. Сутності Vertex (вершина) та Edge (ребро) є аналогами моделей Concept та Relation, що представляють поняття та зв'язок. Вони містять поля, необхідні для побудови карти понять. Клас MapModel забезпечує доступ до списків об'єктів класів Concept та Relation.

5.2.2 Компоненти навігації між вершинами на карті понять

Представлення системи містить видимі користувачу віджети - UI елементи системи, та їхню верстку. Серед цих віджетів - відображення текстового наповнення контенту тез, зовнішній вигляд карт понять, елементи пошуку та навігації та інші. Такі класи не повинні містити логіки або спілкуватися з сервером, адже це задача контролера та сервісів. Рисунок 5.5 зображує файлову структуру представлення системи.

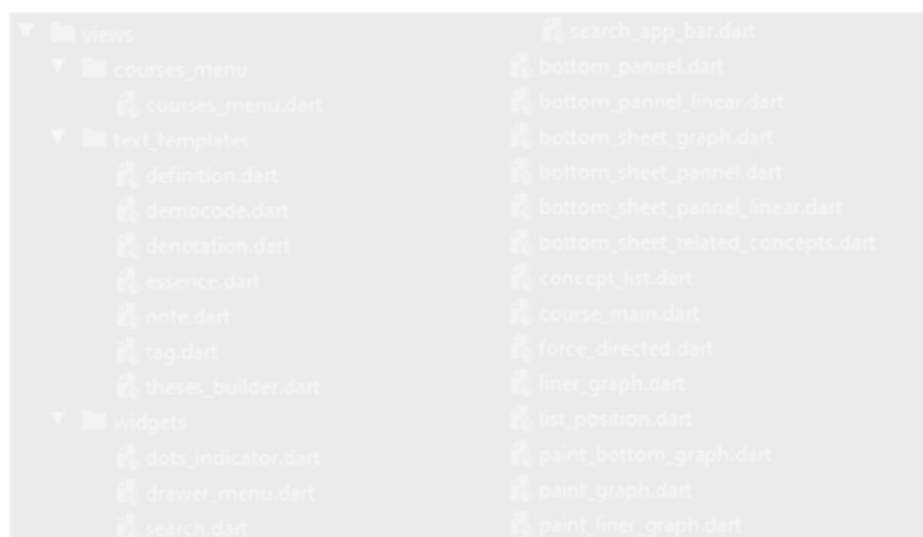


Рисунок 5.5 – Файлова структура представлення

Файл `force_directed.dart` містить клас, що відповідає за побудову карт понять та їхню верстку. Клас `ForceDirected` отримує від провайдера інформацію про вершини графу та зв'язки. Провайдер де зберігає дані, що потім передаються контролер. Контролер розраховує початкові положення вершин графа, їх кольори. Після виконання силового алгоритму побудови карти понять координат кінцевих положень також зберігаються в контролер. Отримані дані йдуть до

функції, яка відповідальна за створення лісту віджетів, якими буде наповнений граф. Кожен віджет відповідає вершині створеного графа, ребрам та текстовим назвам вершин. Побудовані віджети інтерактивні та реагують на дії користувача. Під час клацання по обраній вершині графа, викликається метод `runAnimation`, який відповідає за запуск анімації й фокусування вершини, приближуючи інтерфейс карти понять до неї й змінюючи масштаб карти. Цей метод запускає анімацію відкривання нижньої панелі, це триває 0.5 секунди. Користувач може відкрити допоміжну панель також за допомогою свайпу вгору, або закрити свайпом вниз. Кожен раз панель відкривається на 0.4 від висоти екрана мобільного девайсу. Після виконання алгоритму інформація про активну вершину зберігається у провайдер під назвою змінної `focusNode`.

Класи мобільного застосунку, назва яких містить слово «`paint`» або «`painter`» зберігають верстку ребер направленного інтерактивного графа зі стрілками й напрямку. Для побудови ребер графа використано клас «`Canvas`». Цей клас відповідає за побудову в застосунку необхідних геометричних фігур будь-якої форми. Ребра побудовані за двома точками - центрами вершин, між якими ребро проведено. Стрілки на ребрах вказують напрямок направлення та будуються посередині ребра за 3 точками, які знаходяться автоматично за розрахунками під час виконання методу побудови.

Файл `courses_menu.dart` містить клас `CoursesMenu`, в якому збережено верстку меню з вибором курсу для вивчення. Меню курсів виглядає як перелік градієнтних заокруглених прямокутників з назвою курсу, розтянутих на ширину девайсу. При натисканні на обраний користувачем курс застосунок переходить до наступного екрану з картою понять.

Файл `bottom_sheet_annel.dart` містить улас `BottomSheetPannel`, де знаходиться верстка нижньої панелі. Допоміжна панель має три вкладки, які помічені стікерами: зеленим, фіолетовим та помаранчевим. `BottomSheetPannel` є провідним класом між класами `BottomPannel` (зелений стікер у верстці) та `BottomSheetGraph` (позначений фіолетовим). Така верстка втілює дизайн конспекту. Перелічені вище

класи відповідальні за верстку допоміжної нижньої, яка містить дані про поняття, тези, та навігацію по поняттям. Клас BottomSheetPannel зберігає методи для запуску анімацій відкриття та закриття нижньої панелі. Він також містить умов переходів по вкладкам.

Клас BottomSheetGraph містить верстку вкладки нижньої панелі, яка відповідає за навігацію картою понять. Цей елемент навігації карти понять є круговим списком нащадків вибраного поняття, що відображається в центрі. Після переходу до навігаційної каруселі на екран виводяться 4 нащадки вибраного концепту. Цих нащадків можна обертати ліворуч або праворуч неначе рулетку з допомогою жестів свайпу, за що відповідає клас GestureDetector. Для більш комфортного використання навігаційний елемент має фізичне прискорення, яке збільшується із силою руху пальця. Крім того, список є циклічним, що дозволяє користувачеві нескінченно обертати нащадків. За періодичне переміщення елементів рулетки відповідає окремо розроблений алгоритм обертання, який аналізує список активних та неактивних елементів. На рисунку 5.6 показано схему переходів між елементами на карті понять. Користувач може переходити від обраного поняття до дочірніх понять у бажаному порядку та назад у міру необхідності.

Для більшої зручності користувача кількість загальних дочірніх елементів та кількість активних елементів можна відстежувати у графічній навігаційній панелі. Індикатор дочірніх понять представлений як список точок, де активні поняття виділені синім кольором, а неактивні — сірим. Користувач може перейти на нащадка елемента обраного поняття, натиснувши на нього. Потім вибрані концепції стають активними у провайдері. Цей перехід відбувається не тільки навігаційній панелі, але і в самій карті понять. Інтерфейс над панеллю переміститься до вибраної концепції, змінюючи масштаб. Також за бажанням користувач може клацнути синю стрілку під активним поняттям, щоб повернутися до раніше обраного поняття або до нащадку поточного поняття.



Рисунок 5.6 – Схема навігації по карті понять за допомогою панелі

Представлений навігаційний дизайн вирішує проблему використання карт понять на обмежених розмірах екранів мобільних девайсів. Завдяки круговій прокрутці було розроблено мінікарту, на якій поміщається нескінченна кількість елементів карти. Розроблений дизайн мінікарти зрозумілий користувачу завдяки загальним та активним понятійним індикаторам.

5.2.3 Компонент відображення тез

Клас BottomPanel зберігає перелік тез, що зів'язані з поняттям, яке знаходиться в фокусі застосунку. Із класу виконується звернення до провайдера, що в свою чергу бере дані з API. На рисунку 5.7 зображено схематичну роботу класу

ThesisBuilder, який відповідає за представлення тексту в системі, а також представлені приклади відповідних шаблонів.

Для перегляду статті використано ID конкретного концепту. Потім дані про поняття надсилаються класу ThesesBuilder, що відповідальний за подання тез у належному форматі. У класі реалізується пошук класу представлення та версти текстової тези в залежності від її типу.

Дані класи презентацій тез на інформаційній панелі перераховані за їхнім типом. Клас ThesisBuilder обробляє тип тези, отриманий від API, та перевіряє текстовий шаблон, якому вона відповідає. Потім викликається контролер, який відповідає результату порівняння тези та її типу.

В системі представлено до шести можливих типів тез: означення, сутність, елемент програмного коду, позначення, примітка і тег. Дані типи збережено в enum. Таке розмаїття оформлення текстових тез ґрунтується на кращому засвоєнні знань під час навчання [12].



Рисунок 5.7 – Схема роботи ThesisBuilder і приклади текстових шаблонів

5.2.4 Компонент статистичних даних

Клас `UserProvider` містить у собі дані про поточного користувача системи, а саме його ідентифікатор, обрані для навчання курси та логи - дії користувача в системі. За допомогою `id` користувача застосунок звертається до сервера і отримує доступ до потрібної персональної інформації. Дії користувача представляє модель `UserLog`, у якій збережено ідентифікатор події, її тип, тип контенту, час початку та кінця й її тривалість у секундах. Тип події “view” означає, що користувачем продивлявся певний контент освітнього порталу. Тип контент “concept” вказує на те, що було відкрито саме поняття. Такий підхід дозволяє розробнику збирати й аналізувати інформацію про навчання користувача, модифікувати карту понять під його особисті потреби або будувати корисну статистику.

При запуску додатку йде перевірка на поточного користувача. Якщо немає даних про його авторизацію, додаток направляє користувача на сторінку з формами вводу логіна та пароля, а потім викликається метод `authorizeUser()`, що зберігає поточного юзера в системі. Після успішної авторизації або при запуску додатку викликаються методи `fetchCourses()` та `fetchUserLogs()`. Дані методи отримують вибрані користувачем курси та логи користувача відповідно, зберігаючи їх у списки.

Отримані логи користувача обробляються спеціальним алгоритмом, який аналізує кожну подію та зберігає івенти продивлення понять у окремий список. Також для зручності подальшого аналізу ідентифікатори понять зберігаються у `Set` - список з елементів, які не можуть повторюватися. В цей же час запускається алгоритм рахування часу й у модель поняття `Concept` записується сумарний час, витрачений на вивчення поняття в секундах. На отриманих даних побудовану функцію помічання вивчених понять - якщо користувач її ввімкне, відповідні вершини та ребра на карті понять пофарбуються у зелений колір, сигналізуючи, що вони вже вивчалися. Також над тезами поняття на допоміжній нижній панелі

вказується час, витрачений на вивчення концепту. Завдяки методу `formatTime()` час с секунд переводиться у години та хвилини, ці дані надходять у верстку не допоміжну панель.

При відкриванні поняття на карті понять викликається функція `startLoggingConcept()`, яка починає рахувати в секундах скільки часу користувач провів на понятті, а потім відправляє ці дані на сервер.

Для відображення статистики та аналізу статистичних даних викликається функція `getStatistics()`, яка приймає як аргумент поточну карту понять. Функція збирає у Map дані про поняття та час їхнього вивчення, ключем слугує назва поняття, а значенням - час у секундах. Далі статистичні дані сортуються й виводяться користувачу на екран в порядку спадання часу.

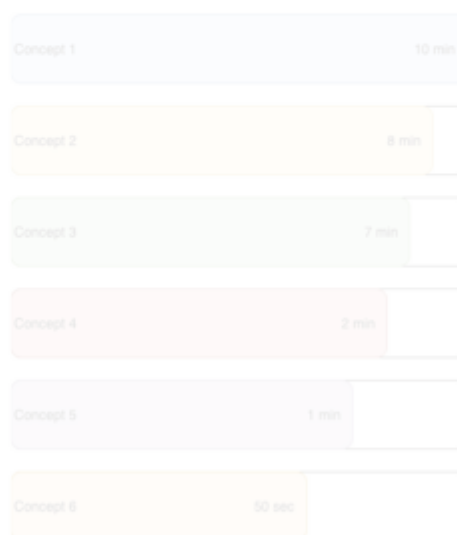


Рисунок 5.8 – Схема компоненту статистики

Представлені навігаційні елементи є важливою частиною мобільного застосунку, адже вони значно спрощують розуміння карти понять завдяки зручній навігації та позитивно впливають на навчання завдяки статистиці.

6 ОПИС РОБОТИ КОРИСТУВАЧА З СИСТЕМОЮ

Для запуску мобільного застосунку потрібен смартфон з встановленими операційними системами Android версії 8.0 та вище або IOS версії 11.0 та вище. Програмне забезпечення можна завантажити у магазинах App Store або Google Play. Також на Android можна інстальювати виконуваний файл з програмою concept_maps.exe. За умови відсутності мобільного девайсу, систему можна запустити у емуляторі на ПК.

Розроблений застосунок є адаптивним під будь-які розміри екрану та може бути запущений на планшеті або комп'ютері з операційною системою MacOS. Після інсталювання та запуску освітнього мобільного додатку, користувачу відкривається екран авторизації (рисунок 6.1).

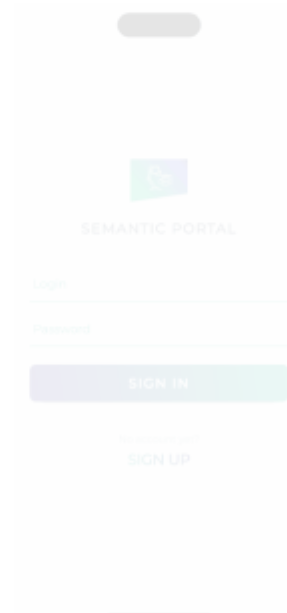


Рисунок 6.1 - Авторизація мобільного застосунку

Користувач може увійти, ввівши свій логін та пароль. У разі відсутності

аккаунту, натиснувши на кнопку “sign up”, відкривається мобільний браузер на сторінці реєстрації semantic-portal.net, де користувачу буде запропоновано створити акаунт. Після успішного створення потрібно повернутися у додаток та авторизуватися. Авторизація у мобільному застосунку є обов’язковою.

Після входу у систему користувач бачить дві вкладки - список обраних для навчання курсів та список усіх наявних доступних для вивчення курсів (рисунок 6.2).

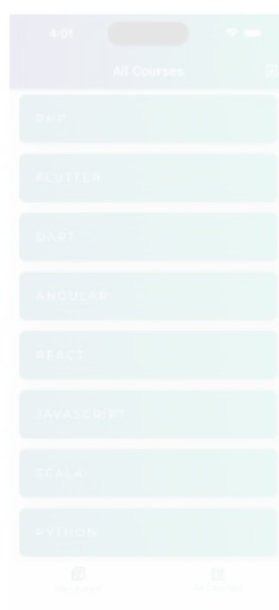


Рисунок 6.2 – Список доступних для вивчення курсів

Після того, як користувач вибере цікавий йому курс, відкривається екран із побудованою картою понять (рисунок 6.3). За допомогою жестів та свайпів карту понять можна змінювати в масштабі, приближуючи та віддаляючи потрібні поняття. Якщо використано емулятор масштаб можна змінити натиснувши клавішу ctrl й прокручуючи колесо миші.



Рисунок 6.3 - Карта понять у мобільному застосунку

Окрім карти понять на екрані користувачу пропонуються опції пошуку, навчальної статистики та відмічання понять, що вивчалися. Функція відмічання понять за замовчуванням вимкнена та кнопка позначена сірим кольором з пустим чекбоксом. Якщо користувач ввімкне дану функцію - кнопка змінить свій колір та у чекбоксі з'явиться галочка. Проглянуті поняття на карті буде пофарбовано в зелений колір (рисунок 6.4), натискаючи на нові поняття вони також будуть фарбуватися. Функція зберігається локально на телефоні, тому до її вимкнення надалі всі карти понять будуть відкриватися також з пофарбованими вузлами, і збережеться навіть після закриття мобільного застосунку повністю.



Рисунок 6.4 - Відмічені вивченні поняття на карті

Користувач мобільної системи може обрати потрібне йому для вивчення поняття. Після натискання на поняття, карта понять змінює свій масштаб, приближаючись до обраного поняття, відкривається нижня допоміжна панель. І панелі за замовчуванням буде відкрито першу вкладку із тезами, які відносяться даного поняття (рисунок 6.5). Вкладка із тезами позначена зеленим стікером. Якщо клацнути по поняттю 2 рази, карта віддаляється, а панель закривається.

На допоміжній панелі користувачу доступно всього 3 вкладки, що позначені зеленим, фіолетовим та помаранчевим стікерами. Перша вкладка дозволяє дослідити доступні дані про обране поняття, або провести аналіз відношень між поняттями при клацанні на ребро між ними, що зображено на рисунку 6.6. Система автоматично виділить назву поняття (або інші форми його назви) іншим

кольором під час проведення аналізу відношень. Колір виділення назви відповідає кольору вершини графа на карті. Також над тезами вказано час вивчення поняття, цю інформацію виділено курсивом. Якщо поняття не було розглянуто, час не вказується.

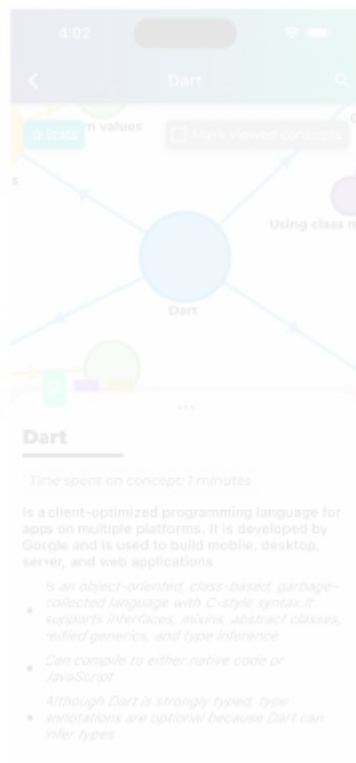


Рисунок 6.5 – Допоміжна панель застосунку

Натискання на фіолетовий стікер переключає нижню панель на другу вкладку, що представляє навігацію за поняттями (рисунок 6.7). Користувачу пропонується карусель, на якій знизу по центру знаходиться обране поняття, зверху зображено усіх його нащадків. Якщо у поняття нащадків багато – користувачу пропонується індикатор кількості нащадків. При натисканні на різ

поняття система змінює фокус на карті та карусель змінюється: головним батьківським поняттямзначається обране, а над ним знаходяться саме його нащадки. Щоб повернутися на рівень нижче або до минулого поняття, потрібно натиснути стрілку вниз.



Рисунок 6.6 - Аналіз відношень при натисканні на ребро

Натискання на помаранчевий стікер переключає допоміжну панель на третю вкладку й відкривається аналіз відношень, на якому зліва зображається обране поняття, а справа доступний список усіх понять, які пов'язані з обраним дидактичними зв'язками (рисунок 6.8). Поняття можна змінити за допомогою жестів – свайпу догори або донизу. При прокручуванні кожне поняття буде зупинено навпроти лінії, яка з'єднує два поняття. Лінію градієнтно пофарбовано в кольори, які відповідають кольорам вершин на графі.

Верхня синя панель містить кнопку переходу у меню, назву сторінки та кнопку пошуку. При переході у пошук, користувач може ввести назву потрібного йому поняття у довільному регістрі, після цього відбувається перехід до шуканого концепту, карта приближується та відкривається нижня панель з описом поняття. За замовченням допоміжна панель відкривається на вкладці, що помічена зеленим стікером. Проте якщо під час користування застосунком користувач ще використовував нижню панель, то відкриється остання проглянута користувачем вкладка.

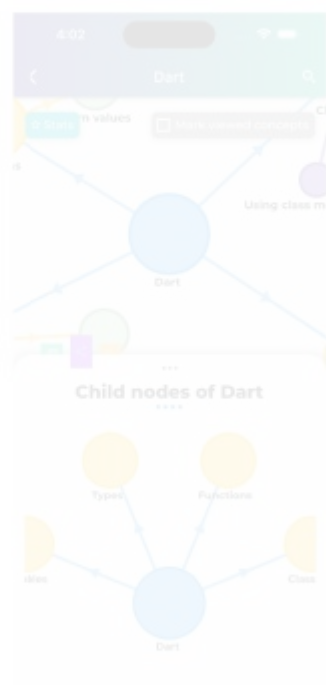


Рисунок 6.7 – Вкладка навігації по карті понять

На апбарі екрану є кнопка з лупою, при натисканні на яку відкривається пошук. Ввівши потрібний запит користувач може швидко та зручно перейти до шуканого поняття. Під полем пошуку доступна історія запитів. Історія зберігає до 10 назв понять, що шукалися користувачем. Справа від назви поняття

розташовано кнопку з іконкою хрестика, за допомогою якої користувач може видалили елемент з історії пошуку.

Нижню панель можна підняти або опустити будь-якої миті, провівши пальцем вгору або вниз. Коли панель повністю згорнута, свайп пальцем вгору дозволяє підняти її на половину від висоти екрана. Крім того, юзер може провести вгору, щоб збільшити висоту екрана ще на 40%. Так само ви можете провести вниз, щоб згорнути панель. Жести вправо та вліво недоступні для перемикачів вкладками, позначеними зеленою, фіолетовою та помаранчевою наклейками. Користувач повинен натиснути на сам стікер для перемикачів. Якщо натиснути на наклейку двічі, панель повністю буде закрито.

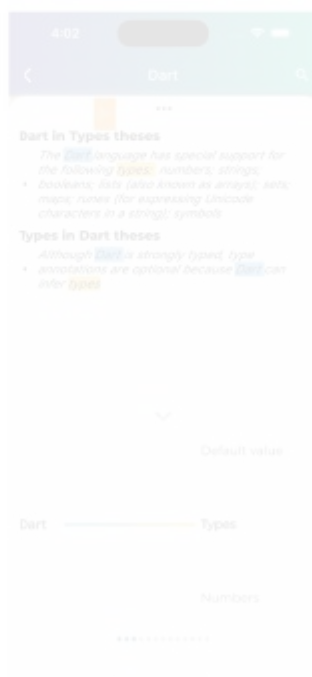


Рисунок 6.8 – Панель аналізу відношень між поняттями

Останнім навігаційним елементом системи є список вкладений понять

65

(рисунок 6.9). Цей список є змістом курсу та дозволяє відслідкувати всі поняття та їх нащадків.



Рисунок 6.9 – Зміст обраного курсу

Окремі рівні змісту для комфортної навігації курсом користувач може згорнути та розгорнути, клацнувши на стрілку зліва. Стрілка відсутня у випадк відсутності нащадків.

ВИСНОВКИ

Під час виконання магістерської дисертації було розроблено освітній онтологічно-орієнтований мобільний застосунок з широким застосуванням карт понять. Застосунок призначений для студентів та школярів, допомагаючи їм вивчати потрібну предметну область. Студенти можуть досліджувати інформацію про поняття та відношення між ними, аналізувати дидактичні зв'язки, слідкувати які саме концепти були вивчені та доступатися до навчальної персональної статистики. Під час розробки програмного продукту було побудовано архітектуру застосунку, розроблено зручний та зрозумілий дизайн, реалізовано програмний код та описано інструкції користувача. Застосунок написано мовою програмування Dart. Було використано середовище розробки Android Studio та фреймворк Flutter. Дані для побудови карт та навчальної статистики користувача було взято з API освітнього веб-сервісу semantic-portal.net.

Таким чином, під час виконання дисертації було здійснено наступні етапи роботи:

- 1) досліджено літературу за темою магістерської дисертації;
- 2) проведено аналіз існуючих програмних рішень;
- 3) досліджено можливості мови Dart та здійснено роботу з API;
- 4) спроектовано архітектуру мобільного додатку, побудовано діаграму прецедентів та діаграму класів;
- 5) розроблено макети мобільного застосунку;
- 4) розроблено алгоритм побудови карт понять та аналізу статистичних даних;
- 5) написано програмний код продукту;
- 6) протестовано розроблену систему.

Під час виконання дипломної роботи було розроблено та протестовано повнофункціональний онтологічно-орієнтований мобільний застосунок з широким застосуванням карт понять.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Romero, C., Cazorla, M., & Buzón, O. (2017). Meaningful learning using concept maps as a learning strategy. *Journal of Technology and Science Education*, 7, 313-332. Ikkink, Hubert (November 2012). *Gradle Effective Implementation Guide* (First ed.). Packt Publishing. p. 382. ISBN 978-1849518109.
2. S.V. Tytenko, "Construction of didactic ontology based on the analysis of concept-thesis model elements", *Naukovi Visti NTUU KPI*, no. 1, pp. 82—87, 2010.
3. Slavomir Stankov, Branko Žitko and Ani Grubišić. *Ontology as a Foundation for Knowledge Evaluation in Intelligent E-learning Systems. AIED'05 Workshop SW-EL'05: Applications of Semantic Web Technologies for E-Learning. Papers of 12th International Conference on Artificial Intelligence in Education (AIED 2005)*. Amsterdam, 2005.
4. E. Andres et al., "An adaptive Theory of Computation online course in ActiveMath", in *Proc. 5th Int. Conf. Computer Science & Education*, 2010, pp. 317—322.
5. Semantic Portal [Електронний ресурс] – Режим доступу до ресурсу: <http://semantic-portal.net>.
6. S. Puntambekar et al., "Improving navigation and learning in hypertext environments with navigable concept maps", *HumanComputer Interaction*, vol. 18, no. 4, pp. 395—428, 2003. doi: 10.1207/S15327051HCI1804_3
7. Поленова В. А., Феденко В. А., Коваленко Д. Р., Титенко С. В. Методи підвищення наочності карт понять у навчальних мобільних застосунках // *Сталий розвиток — 21 століття. Дискусії 2020: колективна монографія* / Націо-нальний університет "Києво-Могилянська академія" / за ред. проф.

Хлобистова Є.В. — Київ, 2020. — С. 435-441 с. — Електронне видання.
ISBN: 978-617-7668-22-9.

8. Thomas M. J. Fruchterman, Edward M. Reingold (1991). Graph drawing by force-directed placement. , 21(11), 1129–1164. doi:10.1002/spe.4380211102
9. Gansner, E., Koren, Y., & North, S. (2004). Graph Drawing by Stress Majorization. Graph Drawing.
10. Thomas M. J. Fruchterman; Edward M. Reingold (1991). Graph drawing by force-directed placement. , 21(11), 1129–1164. doi:10.1002/spe.4380211102
11. Di Battista, Giuseppe, Eades, Peter; Tamassia, Roberto, Tollis, Ioannis G. (1994), "Algorithms for Drawing Graphs: an Annotated Bibliography", Computational Geometry: Theory and Applications.
12. Tytenko, S. V. Interactive concept maps in ontology-oriented information and learning web-systems. KPI Science News, no. 2, pp. 24–36, 2019. doi:10.20535/kpi-sn.2019.2.167515
13. Nesbit, J. C., & Adesope, O. O. (2006). Learning With Concept and Knowledge Maps: A Meta-Analysis. Review of Educational Research, 76(3), 413–448. doi:10.3102/00346543076003413
14. Peter Eades. A heuristic for graph drawing. Congressus numerantium, 42:149–160, 1984.
15. Rap Payne. Beginning App Development with Flutter: Create Cross-Platform Mobile Apps 1st ed. Edition, 35–74, 2019.
16. Schwab Michail, Strobelt Hendrik, Tompkin James, Fredericks Colin, Huff Connor, Higgins Dana, Strezhnev Anton, Komisarchik Mayya, King Gary, Pfister Hanspeter (2017). booc.io: An Education System with Hierarchical Concept Maps and Dynamic Non-linear Learning Plans. IEEE Transactions on Visualization and Computer Graphics, 23(1), 571–580. doi:10.1109/tvcg.2016.2598518
17. Polienova V.A., Fedenko V.A., Tytenko, S.V. EDUCATIONAL MOBILE APPLICATION BASED ON CONCEPT MAPS. Modern engineering and

69

innovative technologies, issue 23, part 1., 14, 12–18, 2022.
doi:10.30890/2567-5273.2022-23-01-014.

Цитати

Цитати

3

1 Ikkink, Hubert (November 2012).

2 Gradle Effective Implementation Guide (First ed.).

3 Schwab Michail, Strobelt Hendrik, Tompkin James, Fredericks Colin, Huff Connor, Higgins Dana, Strezhnev Anton, Komisarchik Mayya, King Gary, Pfister Hanspeter (2017).

Вилучення

Вилучення

117

http://www.setlab.net/downloads/Polienova-Fedenko-Kovalenko-Tytenko.pdf	2.01%
http://www.setlab.net/downloads/publications/Kovalenko-Tytenko-MEIT-2021.pdf	4 джерела 1.57%
http://www.setlab.net/?view=group:articles	17 джерел 1.34%
https://core.ac.uk/download/pdf/323525218.pdf	0.91%
https://core.ac.uk/download/pdf/323530903.pdf	0.88%
http://www.setlab.net/downloads/Publications/Tytenko_KPI_2019.pdf	0.78%
http://www.setlab.net/?view=Tytenko-KPI-2019	0.76%
https://core.ac.uk/download/pdf/323529039.pdf	0.73%
https://core.ac.uk/download/pdf/323536723.pdf	0.67%
http://www.setlab.net/downloads/Kovalenko-Tytenko-kpi-2021.pdf	0.6%
https://ua-referat.com/uploaded/paraleleni-algoritmi-trenuvannya-nejronnih-merej/index1.html	0.58%
https://ames.kpi.ua/wp-content/uploads/2021/01/ToporivskyiAR_magistr.pdf	0.57%
https://core.ac.uk/download/pdf/323527635.pdf	0.54%
https://ela.kpi.ua/bitstream/123456789/46543/1/Bezbakh_magistr.pdf	0.28%
http://www.setlab.net/?view=Terelkina-Furman-Tytenko.pdf-2020	0.24%
https://es.scribd.com/document/391463842/Tecnologia-y-Accesibilidad	0.21%
https://nubip.edu.ua/sites/default/files/u284/pidpriemnictvo_torgivlya_ta_birzhova_diyalnist_24001.pdf	0.2%
http://elartu.tntu.edu.ua/bitstream/lib/36757/2/2_%D0%9C%D1%96%D1%80%D0%BE%D1%88%D0%BD%D0%B8%D0%BA	2 джерела 0.17%
https://ela.kpi.ua/bitstream/123456789/44869/1/Metody_doslidzhen.pdf	3 джерела 0.17%
http://dspace.wunu.edu.ua/bitstream/316497/3270/1/dis_melnyk.pdf	0.16%

https://core.ac.uk/download/pdf/323532279.pdf	8 джерел	0.16%
https://ua-referat.com/uploaded/ekonomichne-obruntuvannya-napryamiv-pidvishennya-efektivnosti/index1.html		0.15%
https://researchr.org/publication/SchwabSTFHHSKKP17		0.14%
https://ipze.kpi.ua		0.12%
https://r-libre.telug.ca/623		0.1%
https://umrs.org.ua/society/society-participants	23 джерела	0.09%
https://cyberleninka.ru/article/n/accounting-and-analysis-of-settlements-with-suppliers-and-contractors	3 джерела	0.09%
https://ela.kpi.ua/bitstream/123456789/32011/1/Bukhanenko_magistr.docx	2 джерела	0.09%
https://gking.harvard.edu/publications/boocio-education-system-hierarchical-concept-maps		0.08%
http://ihvs.kiev.ua/wp-content/uploads/2019/05/Avt_ref_Kovalchuk_A_I.pdf		0.08%
https://dspace.dsau.dp.ua/bitstream/123456789/5710/1/%D0%A1%D0%B0%D0%B4%D0%BE%D0%B2%D0%B8%D0%B9%20%D0%...		0.07%
https://eprints.whiterose.ac.uk/123840/1/SEQA2017_Libro_de_resumenes.pdf		0.07%
https://worldwidescience.org/topicpages/m/market+structure+behavior.html		0.07%
http://dspace.oneu.edu.ua/jspui/bitstream/123456789/10474/2/%D0%A0%D0%B5%D1%84%D0%B5%D1%80%D0%B0%D1%...	8 джерел	0.07%
http://dspace.oneu.edu.ua/jspui/bitstream/123456789/6758/1/%D0%A0%D0%B8%D0%B1%D1%87%D0%B5%D0%BD%D0%BA%D...		0.07%
https://osvita.kpi.ua/sites/default/files/opfiles/122_OPPB_SMSHl_2022.pdf		0.07%
https://conference-service.com/conferences/neural-networks.html		0.07%
https://www.public.asu.edu/~kvanlehn/distrib/ANDES.pdf	4 джерела	0.07%
http://elartu.tntu.edu.ua/bitstream/lib/35714/1/%D0%91%D0%90%D0%9A_%D0%9A%D1%83%D0%B7%D1%8C%D0%BE_%...	3 джерела	0.07%
http://dspace.onua.edu.ua/bitstream/handle/11300/1966/Denisenko_diss.pdf	5 джерел	0.07%
http://eprints.iain-surakarta.ac.id/1954/1/Eisha%20Jamila%20Qomariyah%20Ikhwan.pdf		0.07%
http://oneu.edu.ua/wp-content/uploads/2018/01/OSNOVNI-VIMOGI-do-organizatsiyi-pidgotovki-ta-zahistu-kvalifikatsiyovoyi	5 джерел	0.07%

https://ae.fea.kpi.ua/ae-files/doc/diplom/Recommend_DP.pdf

0.07%

Вилучення по Бібліотеці акаунту

701

Студентська робота	ID файлу: 1012959500	Навчальний заклад: National Technical University of Ukraine	275 Джерело	9.82%
Студентська робота	ID файлу: 1008267455	Навчальний заклад: National Technical University of Ukraine	17 Джерело	7.67%
Студентська робота	ID файлу: 1009452212	Навчальний заклад: National Technical University of Ukraine	12 Джерело	3.97%
Студентська робота	ID файлу: 1008267323	Навчальний заклад: National Technical University of Ukraine "Kyiv Po...		3.33%
Студентська робота	ID файлу: 1012937489	Навчальний заклад: National Technical University of Ukraine	49 Джерело	3.5%
Студентська робота	ID файлу: 1012937558	Навчальний заклад: National Technical University of Ukraine	5 Джерело	2.64%
Студентська робота	ID файлу: 1012937547	Навчальний заклад: National Technical University of Ukraine "Kyiv Po...		2.32%
Студентська робота	ID файлу: 1009447263	Навчальний заклад: National Technical University of Ukraine	4 Джерело	1.98%
Студентська робота	ID файлу: 1008294135	Навчальний заклад: National Technical University of Ukraine	2 Джерело	1.73%
Студентська робота	ID файлу: 1009447346	Навчальний заклад: National Technical University of Ukraine "Kyiv Po...		1.26%
Студентська робота	ID файлу: 1009403038	Навчальний заклад: National Technical University of Ukraine	5 Джерело	1.22%
Студентська робота	ID файлу: 1009483837	Навчальний заклад: National Technical University of Ukraine "Kyiv Po...		0.9%
Студентська робота	ID файлу: 1009495857	Навчальний заклад: National Technical University of Ukraine	6 Джерело	0.82%
Студентська робота	ID файлу: 1009433921	Навчальний заклад: National Technical University of Ukraine "Kyiv Po...		0.73%
Студентська робота	ID файлу: 1009497499	Навчальний заклад: National Technical University of Ukraine "Kyiv Po...		0.72%
Студентська робота	ID файлу: 1000022016	Навчальний заклад: National Technical University of Ukraine	24 Джерело	0.72%
Студентська робота	ID файлу: 1000050446	Навчальний заклад: National Technical University of Ukraine	4 Джерело	0.71%
Студентська робота	ID файлу: 1009503380	Навчальний заклад: National Technical University of Ukraine "Kyiv Po...		0.7%
Студентська робота	ID файлу: 1009475753	Навчальний заклад: National Technical University of Ukraine "Kyiv Po...		0.69%
Студентська робота	ID файлу: 1005666703	Навчальний заклад: National Technical University of Ukraine	2 Джерело	0.63%

Студентська робота	ID файлу: 1009476854	Навчальний заклад: National Technical University of Ukraine	3 Джерело	0.62%
Студентська робота	ID файлу: 1009504601	Навчальний заклад: National Technical University of Ukraine	4 Джерело	0.61%
Студентська робота	ID файлу: 1000039714	Навчальний заклад: National Technical University of Ukraine	7 Джерело	0.61%
Студентська робота	ID файлу: 1000035048	Навчальний заклад: National Technical University of Ukraine	17 Джерело	0.6%
Студентська робота	ID файлу: 1000047160	Навчальний заклад: National Technical University of Ukraine	6 Джерело	0.6%
Студентська робота	ID файлу: 1011340965	Навчальний заклад: National Technical University of Ukraine "Kyiv Po...		0.59%
Студентська робота	ID файлу: 1000043043	Навчальний заклад: National Technical University of Ukraine "Kyiv Po...		0.58%
Студентська робота	ID файлу: 1011422372	Навчальний заклад: National Technical University of Ukraine	2 Джерело	0.42%
Студентська робота	ID файлу: 1009477281	Навчальний заклад: National Technical University of Ukraine "Kyiv Po...		0.4%
Студентська робота	ID файлу: 1000748509	Навчальний заклад: National Technical University of Ukraine	3 Джерело	0.27%
Студентська робота	ID файлу: 1011365840	Навчальний заклад: Lviv Polytechnic National University	51 Джерело	0.27%
Студентська робота	ID файлу: 1000755753	Навчальний заклад: National Technical University of Ukraine "Kyiv Po...		0.27%
Студентська робота	ID файлу: 1000735925	Навчальний заклад: National Technical University of Ukraine	17 Джерело	0.27%
Студентська робота	ID файлу: 1000756067	Навчальний заклад: National Technical University of Ukraine	2 Джерело	0.26%
Студентська робота	ID файлу: 1005757762	Навчальний заклад: National Technical University of Ukraine	2 Джерело	0.25%
Студентська робота	ID файлу: 1000782438	Навчальний заклад: National Technical University of Ukraine	9 Джерело	0.24%
Студентська робота	ID файлу: 1009496565	Навчальний заклад: National Technical University of Ukraine "Kyiv Po...		0.23%
Студентська робота	ID файлу: 1011424341	Навчальний заклад: National Technical University of Ukraine	8 Джерело	0.23%
Студентська робота	ID файлу: 1000751205	Навчальний заклад: National Technical University of Ukraine	13 Джерело	0.21%
Студентська робота	ID файлу: 1002982941	Навчальний заклад: National Technical University of Ukraine "Kyiv Po...		0.21%
Студентська робота	ID файлу: 1000755809	Навчальний заклад: National Technical University of Ukraine	2 Джерело	0.19%
Студентська робота	ID файлу: 1005756412	Навчальний заклад: National Technical University of Ukraine	7 Джерело	0.18%

Студентська робота	ID файлу: 1011406612	Навчальний заклад: National Technical University of Ukraine	3 Джерело	0.17%
Студентська робота	ID файлу: 1009466089	Навчальний заклад: National Technical University of Ukraine	4 Джерело	0.16%
Студентська робота	ID файлу: 1009678160	Навчальний заклад: National Technical University of Ukraine	29 Джерело	0.16%
Студентська робота	ID файлу: 1011482871	Навчальний заклад: Yuriy Fedkovych Chernivtsi National Univ	5 Джерело	0.16%
Студентська робота	ID файлу: 1009668032	Навчальний заклад: National Technical University of Ukraine "Kyiv Po...		0.16%
Студентська робота	ID файлу: 1005772707	Навчальний заклад: National Technical University of Ukraine	2 Джерело	0.16%
Студентська робота	ID файлу: 1005742580	Навчальний заклад: National Technical University of Ukraine	2 Джерело	0.16%
Студентська робота	ID файлу: 1000725974	Навчальний заклад: National Technical University of Ukraine "Kyiv Po...		0.16%
Студентська робота	ID файлу: 1011465653	Навчальний заклад: National Technical University of Ukraine "Kyiv Po...		0.15%
Студентська робота	ID файлу: 1005712386	Навчальний заклад: National Technical University of Ukraine	3 Джерело	0.15%
Студентська робота	ID файлу: 1000029931	Навчальний заклад: National Technical University of Ukraine "Kyiv Po...		0.15%
Студентська робота	ID файлу: 1000026860	Навчальний заклад: National Technical University of Ukraine	2 Джерело	0.15%
Студентська робота	ID файлу: 1011412483	Навчальний заклад: National Technical University of Ukraine	4 Джерело	0.14%
Студентська робота	ID файлу: 1012855996	Навчальний заклад: Lviv Polytechnic National University	5 Джерело	0.14%
Студентська робота	ID файлу: 1000711361	Навчальний заклад: National Technical University of Ukraine	13 Джерело	0.12%
Студентська робота	ID файлу: 1000745187	Навчальний заклад: National Technical University of Ukraine	7 Джерело	0.11%
Студентська робота	ID файлу: 1004080738	Навчальний заклад: Lviv Polytechnic National University		0.11%
Студентська робота	ID файлу: 1000051901	Навчальний заклад: National Technical University of Ukraine	2 Джерело	0.09%
Студентська робота	ID файлу: 1005785598	Навчальний заклад: National Technical University of Ukraine	2 Джерело	0.08%
Студентська робота	ID файлу: 1005707596	Навчальний заклад: Yuriy Fedkovych Chernivtsi National University		0.08%
Студентська робота	ID файлу: 1000017130	Навчальний заклад: National Technical University of Ukraine	6 Джерело	0.08%
Студентська робота	ID файлу: 1004091474	Навчальний заклад: National Technical University of Ukraine	2 Джерело	0.08%

Студентська робота	ID файлу: 1003820983	Навчальний заклад: Lviv Polytechnic National University	0.08%
Студентська робота	ID файлу: 1009693978	Навчальний заклад: National Technical University of Ukraine "Kyiv Po...	0.07%
Студентська робота	ID файлу: 1007988878	Навчальний заклад: Lviv Polytechnic National University	0.07%
Студентська робота	ID файлу: 1009288292	Навчальний заклад: National Technical University of Ukraine 2 Джерело	0.07%
Студентська робота	ID файлу: 5975711	Навчальний заклад: National Technical University of Ukraine "Ky 2 Джерело	0.07%
Студентська робота	ID файлу: 1000022969	Навчальний заклад: National Technical University of Ukraine 4 Джерело	0.07%
Студентська робота	ID файлу: 1011427021	Навчальний заклад: National Technical University of Ukraine 19 Джерело	0.07%
Студентська робота	ID файлу: 1012354238	Навчальний заклад: Zhytomyr National Agroecological Univer 2 Джерело	0.07%