

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ  
Кафедра комп'ютерних наук

**КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА**

на тему: **«РОЗРОБКА ФОРМАЛЬНОЇ МОДЕЛІ  
ПРЕДСТАВЛЕННЯ ЗНАНЬ В ПРЕДМЕТНІЙ  
ОБЛАСТІ «ШКІЛЬНА ІНФОРМАТИКА»**

Виконала: студентка 2 курсу, групи 8.1228

спеціальності 122 комп'ютерні науки  
(шифр і назва спеціальності)

освітньої програми комп'ютерні науки  
(назва освітньої програми)

О.Ю. Діденко

(ініціали та прізвище)

доцент кафедри комп'ютерних наук, к.ф.м.н.

Керівник Єрмолаєв В.А  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

декан математичного факультету, професор, д.т.н.

Рецензент Гоменюк С.І.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

Факультет математичний

Кафедра комп'ютерних наук

Рівень вищої освіти магістр

Спеціальність 122 комп'ютерні науки  
(шифр і назва)

Освітня програма комп'ютерні науки

**ЗАТВЕРДЖУЮ**

Завідувач кафедри комп'ютерних наук, к.т.н., доцент

Борю С.Ю.  
(підпис)

« 30 » травня 2019 р.

**ЗАВДАННЯ**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ**

Діденко Олексині Юріївні

(прізвище, ім'я та по батькові)

1. Тема роботи Розробка формальної моделі представлення знань в предметній області «Шкільна інформатика»

керівник роботи Єрмолаєв Вадим Анатолійович, к.ф.м.н., доцент

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

затверджені наказом ЗНУ від « 29 » травня 2019 року № 811-с

2. Строк подання студентом роботи 16.12.2019

3. Вихідні дані до роботи 1. Постановка задачі.

2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Постановка задачі.

2. Основні теоретичні відомості.

3. Розробка формальної моделі представлення знань відповідно до теми кваліфікаційної роботи

4. Визначення ефективності створеної онтології

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Презентація

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 30.05.2019 р.**КАЛЕНДАРНИЙ ПЛАН**

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	31.05.2019	
2.	Збір вихідних даних.	15.06.2019	
3.	Обробка методичних та теоретичних джерел.	11.09.2019	
4.	Розробка першого та другого розділу.	14.10.2019	
5.	Розробка третього розділу.	5.12.2019	
6.	Оформлення та нормоконтроль кваліфікаційної роботи.	12.12.2019	
7.	Захист кваліфікаційної роботи.	14.01.2020	

Студент

\_\_\_\_\_

(підпис)

О.Ю.Діденко

\_\_\_\_\_

(ініціали та прізвище)

Керівник роботи

\_\_\_\_\_

(підпис)

В.А.Єрмолаєв

\_\_\_\_\_

(ініціали та прізвище)

**Нормоконтроль пройдено**

Нормоконтролер

\_\_\_\_\_

(підпис)

О.Г. Спиця

\_\_\_\_\_

(ініціали та прізвище)

## РЕФЕРАТ

Кваліфікаційна робота магістра «Розробка формальної моделі представлення знань в предметній області «Шкільна інформатика»: 77 с., 9 рис., 6 табл., 16 джерел, 5 додатків.

ДОМЕН, ІНФОРМАТИКА, МЕТОДИКА ВИКЛАДАННЯ, ОНТОЛОГІЯ, ПРЕДМЕТНА ОБЛАСТЬ, РЕЛЕВАНТНІСТЬ, ТЕРМІНОЛОГІЧНЕ НАСИЧЕННЯ, ФОРМАЛЬНА МОДЕЛЬ ПРЕДСТАВЛЕННЯ.

Об'єктом дослідження є предметна область «шкільна інформатика».

Предмет дослідження: аналіз термінологічного насичення текстів вищезазначеного домену.

Мета роботи: розробити формальну модель представлення знань в предметній області «шкільна інформатика».

Метод дослідження: метод аналізу і синтезу, моделювання та класифікації, метод дедукції та формалізації.

У результаті роботи була створена онтологія предметної області «шкільна інформатика». Розроблена формальна модель знань з обраного домену була також проаналізована за двома напрямками: змістові лінії, що активно обговорюються та реформуються світовою спільнотою вчителів з інформатики, а також методика та дидактичні засади викладання вищезазначеного шкільного курсу. В результаті аналізу обох напрямків був зроблений висновок щодо повноти розкриття питань змісту освіти з інформатики та її методичних засад, а також були розроблені рекомендації, щодо вдосконалення змісту конференції міжнародного рівня з питань інформатики в молодшій та середній школі для покращення якості освіти в Україні та світі шляхом реформування системи освіти з інформатики.

## SUMMARY

Master's Qualification Thesis "Development of the Formal Knowledge Representation Model for the Domain of " School Informatics ": 77 pages, 9 figures, 6 tables., 16 sources, 5 applications.

DOMAIN, FORMAL MODEL OF REPRESENTATION , INFORMATICS, ONTOLOGY, RELEVANCE, SUBJECT MATTER, TEACHING METHOD, TERMINOLOGICAL SATURATION.

The object of the study is the subject area "school computer science".

Subject: analysis of terminological saturation of texts of the above domain.

Aim: To develop a formal model of knowledge representation in the subject area "school computer science".

Method: method of analysis and synthesis, modeling and classification, method of deduction and formalization.

As a result, the ontology of the subject area "school informatics" was created. The developed knowledge model for the selected domain was also analyzed in two directions: content lines, which are actively discussed and reformed by the world community of teachers in informatics and the methodology and didactic principles of teaching the above-mentioned school course. As a result of the analysis of both directions, a conclusion was reached on the completeness of disclosure of the content of informatics education and its methodological principles, as well as recommendations were developed on improving the content of the international conference on informatics in junior and secondary school to improve the quality of education in Ukraine and the world by reforming the computer science education system.

## ЗМІСТ

Завдання.....	2
Реферат .....	4
Summary.....	5
Вступ.....	8
1 Сучасний стан формування навчальних програм з інформатики в середній школі в Україні та світі.....	10
1.1 Інформатика як наука та шкільний курс .....	10
1.2 Огляд нормативної та методичної бази для шкіл України.....	11
1.3 Огляд нормативної та методичної бази для шкіл Європи та світу ...	24
1.4 Огляд і аналіз тематичного наповнення програм з інформатики ....	29
1.5 Огляд і аналіз методичних матеріалів для викладання інформатики в Україні і світі .....	39
1.6 Здобуття термінів з текстів як метод тематичного дослідження домену .....	41
1.7 Мета і завдання роботи, постановка мети роботи та завдань .....	42
1.8 Висновок .....	43
2 Збір та обробка насиченої термінологічної інформації у домені “шкільна інформатика” .....	44
2.1 Пошук та вибір колекції наукових статей для термінологічного аналізу .....	44
2.2 Розробка процедурного пайплайну для здобуття насиченого набору термінів.....	45
2.3 Підготовка колекції для термінологічного аналізу. Розробка каталогу, пошук кількостей цитувань, перетворення у текст.....	48
2.4 Здобуття насиченого набору термінів .....	52
2.5 Групування термінів.....	57
2.6 Висновок .....	58

3 Розробка онтології “шкільна інформатика” та аналіз терміно-логічних розбіжностей між світовими національними програмами з інформатики .....	59
3.1 Методика і програмне середовище розробки онтології .....	59
3.2 Трансформування груп термінів в онтологію у Protege .....	63
3.3 Побудова і аналіз покриття онтології тематикою національних програм з шкільної інформатики.....	66
3.4 Висновок .....	67
Висновки.....	68
Перелік посилань .....	70
Додаток А Результати оцінки насиченості колекції ISSEP .....	73
Додаток Б Розміщення файлів статей по директоріях на Google Drive .....	74
Додаток В Каталог статей .....	75
Додаток Г Скриншот розробленої онтології .....	76
Додаток Д Рекомендації.....	77

## ВСТУП

**Актуальність теми.** Питання якості освіти турбує людей завжди. Щоб дізнатися наскільки якісною є освіта з інформатики та наскільки відповідає рівень викладання світовим тенденціям в освіті треба дослідити цю предметну область. Шкільний курс інформатики в Україні був введений в систему загальної освіти майже 30 років назад і за ці роки реформувався багато разів. Ми вирішили дослідити наскільки правильним є підхід до його викладання в нашій країні за допомогою аналізу знань в даній предметній області. Ми обрали напрямок у бік освітнього процесу з інформатики у середніх класах. Питаннями вдосконалення освіти з інформатики займаються міжнародному рівні. Фахівці-делегати з різних країн світу зустрічаються на міжнародній конференції з питань інформатики ISSEP – International Conference on Informatics in Secondary Schools – Evolution and Perspectives (Інформатика в середніх школах – еволюція та перспективи). Це форум для дослідників та практиків у галузі інформатики, як в початкових, так і в середніх класах загальноосвітніх шкіл, на якому вони обговорюють та розглядають питання про цілі та завдання цього предмету, його навчальні програми, різні парадигми та теми викладання та навчання, а також зв'язки із повсякденним життям, включаючи різні способи розвитку інформатики в школах. Матеріали з цієї конференції знаходяться у вільному доступі, тому було вирішено дослідити їх та за їх допомогою дослідити рівень якості освіти з інформатики в світі та порівняти його з відомостями про якість викладання цього курсу в Україні.

**Новизна.** Елемент наукової новизни даного дослідження полягає в тому, що дослідження якості освіти в предметній області «шкільна інформатика» проводиться на основі аналізу термінологічних розбіжностей між світовими та українськими підходами до викладання, на основі



поступових етапів збору і обробки термінології з даного домену та побудови відповідної формальної моделі представлення знань в цій області.

**Практична цінність.** Результати даної роботи представляються у вигляді готової онтології термінів та розроблених рекомендацій, щодо удосконалення дидактичних засад у викладанні інформатики в українських школах.

**Структура роботи.** Дипломна робота складається з 3 основних розділів, вступу, в якому описані актуальність теми, новизна та практична цінність роботи, з висновку, де описуються результати роботи та додатків.

У першому розділі досліджувалося питання сучасного стану формування навчальних програм з інформатики в середній школі України та світу. Розглянуті нормативно методичні бази для шкіл України і світу, проаналізоване тематичне наповнення програм курсу, розбіжності у методичних підходах до викладання інформатики в українських школах та у школах різних країн Європи. Також в цьому розділі описані алгоритми здобуття термінів з текстів, встановлена мета робота та сформульовані завдання.

У другому розділі описаний алгоритм пошуку та вибору колекції наукових статей для термінологічного аналізу, розглянута розробка процедурного пайплайну для здобуття насиченого набору термінів, описана процедура підготовки колекції для термінологічного аналізу, здобуття і вимірювання насичення, а також обґрунтована насиченість та описаний процес групування термінів.

Третій розділ був присвячений темі розробки онтології “шкільна інформатика” та аналізу термінологічних розбіжностей між світовими і національними програмами з інформатики. В цьому розділі були описані методика і програмне середовище розробки онтології, процес трансформування груп термінів в онтологію у Protege та безпосередньої побудови онтології, проведений аналіз покриття тематикою національних програм з шкільної інформатики.

# 1 СУЧАСНИЙ СТАН ФОРМУВАННЯ НАВЧАЛЬНИХ ПРОГРАМ З ІНФОРМАТИКИ В СЕРЕДНІЙ ШКОЛІ В УКРАЇНІ ТА СВІТІ

## 1.1 Інформатика як наука та шкільний курс

В 60-ті роки минулого століття разом із стрімким розвитком ЕОМ, почала своє становлення і розвиток *кібернетика*. Спочатку кібернетика була наукою про управління і зв'язок у різних системах(штучних, біологічних, соціальних) і згодом стала більш загальною наукою про перетворення інформації. Під *інформацією* у кібернетиці розуміють будь-яку сукупність сигналів, впливів або відомостей, які деяка система сприймає від навколишнього середовища (вхідна інформація), видає у навколишнє середовище (вихідна інформація), а також зберігає у собі (внутрісистемна інформація) [1].

Синонімом до слова кібернетика став термін *інформатика* (нім. *Informatik*), який був використаний та введений в статті «*Informatik: Automatische Informationsverarbeitung*» (Інформатика: Автоматична обробка інформації) 1957 року німецьким кібернетиком, інженером з телекомунікацій, спеціалістом з теорії інформації Карлом Штайнбухом. В деяких джерелах вважають, що слово інформатика (*Informatik*) утворено злиттям слів інформація (*Information*) та автоматичний (*Automatik*); інший погляд полягає в тому, що інформатика (*Informatik*) – це неологізм, утворений зі слова інформація (*Information*) за аналогією зі словом математика (*Mathematik*) [2].

*Інформатика* – це наука про інформацію та інформаційні процеси в природі та суспільстві, методи та засоби пошуку, збирання, одержання, опрацювання, зберігання, подання, передавання інформації та управління інформаційними процесами. Судячи з визначення можна зробити висновок, що інформатика це більш загальне поняття, а кібернетика є її складовою.

*Інформатика* – це динамічна наука. Вона стрімко розвивається та має суттєвий вплив на розвиток інших наук і технологій. Інформатика сьогодення впливає майже на всі сфери людської діяльності. Однією з основних сфер, що зазнали її впливу є система освіти.

*Інформатика* як шкільний курс повинен виконувати загальноосвітні функції, повинен відображати фундаментальні поняття і відомості, які розкривають сутність науки, забезпечувати учнів знаннями, вміннями, навичками, необхідними для вивчення основ інших наук в школі, а також, повинен готувати освітян до життя в сучасному інформаційному суспільстві.

*Мета* навчання шкільного курсу інформатики – формування основ інформаційної культури школярів, тобто формування сукупності знань, умінь, навичок, які забезпечують учням можливість застосовувати комп'ютерну техніку в навчальній, а згодом у професійній діяльності [3].

## 1.2 Огляд нормативної та методичної бази для шкіл України

Методична система навчання будь-якого предмета – це сукупність п'яти основних компонентів:

- а) цілі;
- б) зміст;
- в) методи;
- г) засоби;
- д) організаційні форми навчання.

*Інформатика як навчальний предмет* – це педагогічно адаптована і предметно специфікована система знань, **навчальним об'єктом** якої є предмет інформатики як наукової дисципліни, а **предметом** – результат дидактичного опрацювання наукових знань, які належать до навчального об'єкта, відповідно до цілей навчання. Створення та розвиток методичної системи навчального предмета інформатики відіграє ключову роль у

становленні шкільного курсу інформатики. На відміну від інших методичних систем навчання предметів зі змісту загальної середньої освіти України, які формувалися протягом десятиліть в основному емпірично, перевірялись шкільною практикою і помітно змінювались кожні 10 – 15 років, Інформатика має високий динамізм становлення її методичної системи навчання.

*Основною метою* предмета “Інформатика” є:

а) сформувати знання, уміння і навички, необхідні для раціонального використання засобів сучасних інформаційно-комунікаційних технологій при розв’язуванні задач, пов’язаних з опрацюванням інформації, її пошуком, систематизацією, зберіганням, поданням, передаванням;

б) ознайомити учнів із роллю нових інформаційно-комунікаційних технологій у сучасному виробництві, науці, повсякденній практиці, з перспективами розвитку комп’ютерної техніки;

в) започаткувати основи інформаційної культури учнів.

*Зміст* шкільного предмета “Інформатика” значною мірою залежить від рівня інформатизації навчального процесу, розробки інформаційно-комунікаційних технологій навчання та їх використання при вивченні різних навчальних предметів, змістового наповнення інших навчальних предметів у різних освітніх галузях [3].

Принципи навчання інформатики.

Принципи навчання – це керівні ідеї, нормативні вимоги до організації і здійснення освітнього процесу.

Принцип науковості і посиленої складності.

Принципом науковості і посиленої складності в навчанні передбачається ретельний відбір істотного змісту науки. Саме мета і завдання уроку, сформульовані лаконічно і в доступній для учнів формі, дають перші уявлення учням про ту інформацію, на яку вони повинні спрямувати свою увагу і пізнавальні можливості на уроці.

Принцип послідовності і систематичності навчання.

Головним у цьому принципі є логічна побудова змісту освіти, а також обґрунтована послідовність етапів освітнього процесу.

Для реалізації цього принципу на уроках інформатики доцільно поділяти навчальний матеріал на логічно зв'язані розділи і блоки. Не перевантажувати заняття другорядними фактами; навчати учнів користуватися довідниками, словниками, енциклопедіями.

Принцип наочності змісту і діяльності.

Практичне використання комп'ютерної техніки в навчальному процесі, робота кожного учня з комп'ютером є реалізацією принципу наочності, важливою умовою досягнення загальноосвітніх цілей навчання інформатики.

Наочність – невід'ємна риса навчання інформатики внаслідок гнучкості змісту самого поняття «інформація», способів і засобів її подання: одну і ту саму інформацію можна подати у вигляді багатьох графічних образів.

Принцип активності і самостійності.

Принцип активності в навчанні інформатики передбачає розумову активність учнів у процесі оволодіння діяльністю. В методиці навчання інформатики розрізняють інтелектуальну, емоційну активність. Інтелектуальна активність забезпечується постановкою проблемних завдань, які стимулюють пізнавальні процеси. Емоційна активність пов'язана з питанням цікаво чи не цікаво учням вивчати ті чи інші питання. Позитивне емоційне ставлення до предмета відіграє велику роль у досягненні успіхів.

Самостійність учня також є метою і умовою успішного вивчення інформатики. При переході до творчої діяльності самостійність реалізується повністю.

Принцип свідомості.

Цим принципом передбачається цілеспрямований добір навчального матеріалу, який забезпечує розвиток пізнавальних здібностей учнів. Принцип свідомості реалізується також через усю організацію навчання протягом якого відбувається перехід від усвідомлення правил виконання дії до її

автоматизованого виконання, виформування окремих елементів діяльності до їх об'єднання.

Принцип міцності і системності знань.

Міцність засвоєння навчального матеріалу з інформатики може бути забезпечена в навчальному процесі шляхом підвищення змістовності навчального матеріалу, підвищення внутрішньої мотивації вивчення матеріалу, пошуку конкретних асоціацій, систематичного повторення того, що зберігається в пам'яті, виконання великої кількості творчих завдань, систематичного контролю знань матеріалу і вмінь володіння ним.

Принцип індивідуалізації і колективності навчання.

Тільки організувавши колективну або групову роботу, можна знайти час для занять з сильнішими і слабшими учнями.

Індивідуалізація визначається як організація навчального процесу, під час якого при доборі способів, прийомів, темпу навчання враховуються індивідуальні особливості учнів, рівень розвитку їхніх здібностей до навчання.

Принцип зв'язку теорії з практикою.

Для реалізації принципу зв'язку теорії з практикою вчителю необхідно:

а) намагатися не допускати в навчанні розходження: це – знання, а це – життя.

б) розуміти, що процесі практичної діяльності учень розширює і поглиблює свої теоретичні представлення, «нарощує теорію».

в) навчати дітей технології розумової діяльності. будь-яку виконувану учнями діяльність пропонувати їм для наступного усвідомлення і теоретичного структурування у вигляді моделі.

Принцип гармонійного розвитку особистості.

При проведенні уроків з інформатики необхідно концентрувати увагу на їх виховних можливостях, для чого реалізовувати такі задачі:

а) ставити виховну мету на кожному уроці;

б) використовувати зміст уроку з виховною метою;

- в) цілеспрямовано формувати узагальнені прийоми розумової діяльності учнів;
- г) формувати вміння користуватися набутими знаннями і розширювати їх під час самостійного вивчення;
- д) розвивати самостійність і пізнавальні здібності учнів, готувати їх до творчої діяльності;
- е) формувати критичне та творче мислення;
- є) прищеплювати любов до праці;
- ж) формувати вміння виділяти основне при вивченні навчального матеріалу

Принцип виховуючого навчання.

Принцип виховуючого навчання інформатики реалізується при такій організації навчального процесу, яка забезпечує учням можливість проявити себе як особистість, гармонійно і всебічно розвинутися соціально, удосконалити здібності, сформувати пізнавальні мотиви як домінуючі в навчальній діяльності.

Методи організації і здійснення навчально – пізнавальної діяльності:

- а) за джерелом передачі і сприйняття навчального матеріалу:
  - 1) словесні;
  - 2) наочні;
  - 3) практичні;
- б) за логікою передачі і сприйняття інформації:
  - 1) індуктивні;
  - 2) дедуктивні;
- в) за ступінню самостійності мислення учнів при оволодінні знаннями:
  - 1) репродуктивні;
  - 2) пошукові;
- г) за ступінню управління навчальною роботою:
  - 1) навчальна робота під керівництвом вчителя;

2) самостійна робота учнів.

Під час навчання інформатики слід використовувати з загальнодидактичні принципи, методи, засоби та форми організації роботи.

В українській школі інформатика з'явилась у 1985 році і отримала назву “Основи інформатики та обчислювальної техніки”. Спочатку це був курс обсягом 100 годин, основна увага в межах якого приділялась алгоритмізації та програмуванню [4].

У 1990-х роках розпочалась тенденція скорочення вивчення програмування на інформатиці та відведення все більшої кількості годин на так званий курс користувача.

З 2003 до 2009 року інформатика вивчалась у 10-11 класі, обсягом 70 годин. З них від 3 до 12 годин становило програмування. Існувала можливість вивчати інформатику глибше, обсягом 140 годин, з яких 48 становив розділ програмування.

З 2009 року інформатику почали вивчати у 9-11 класах, загальним обсягом від 105 до 385 годин залежно від профілю (див. рис.1.1).

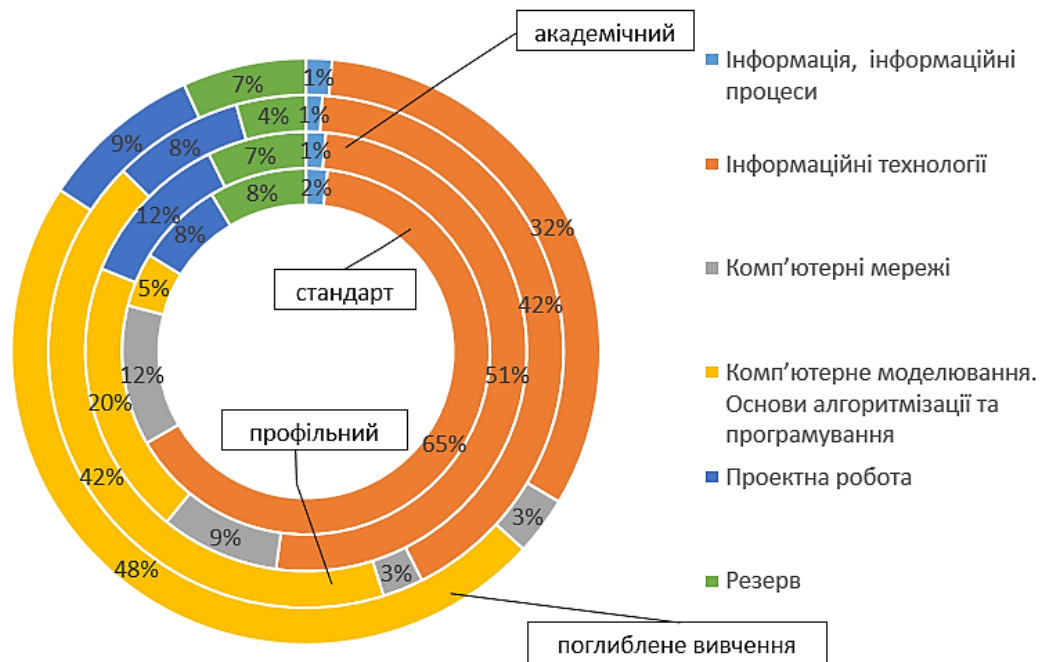


Рисунок 1.1 – Розподіл годин на вивчення інформатики в школах України (у відсотках)



Програмування при цьому вивчається від 5 до 175 годин у класах різних профілів. Переважна більшість учнів вивчає інформатику за найпростішою програмою рівня стандарту (105 годин всього, 5 годин програмування). Багато шкіл України мають профільні класи з інформатики, котрі передбачають вивчення предмету протягом 385 годин на рік, з яких майже половину становить розділ програмування. Учні деяких профілів (наприклад, математичні класи), вивчають інформатику 175 годин, з яких 28 годин становить програмування – це академічний рівень.

З 2013 року інформатику почали вивчати одночасно в 5 та 2 класі. Ці програми передбачають дещо ширше вивчення програмування. Зокрема 2-4 класи, 105 годин інформатики, розділ алгоритмізації 17 годин. 5-9 класи, 245 годин, з яких від 46 до 70 годин відведено на вивчення програмування (див. рис. 1.2).

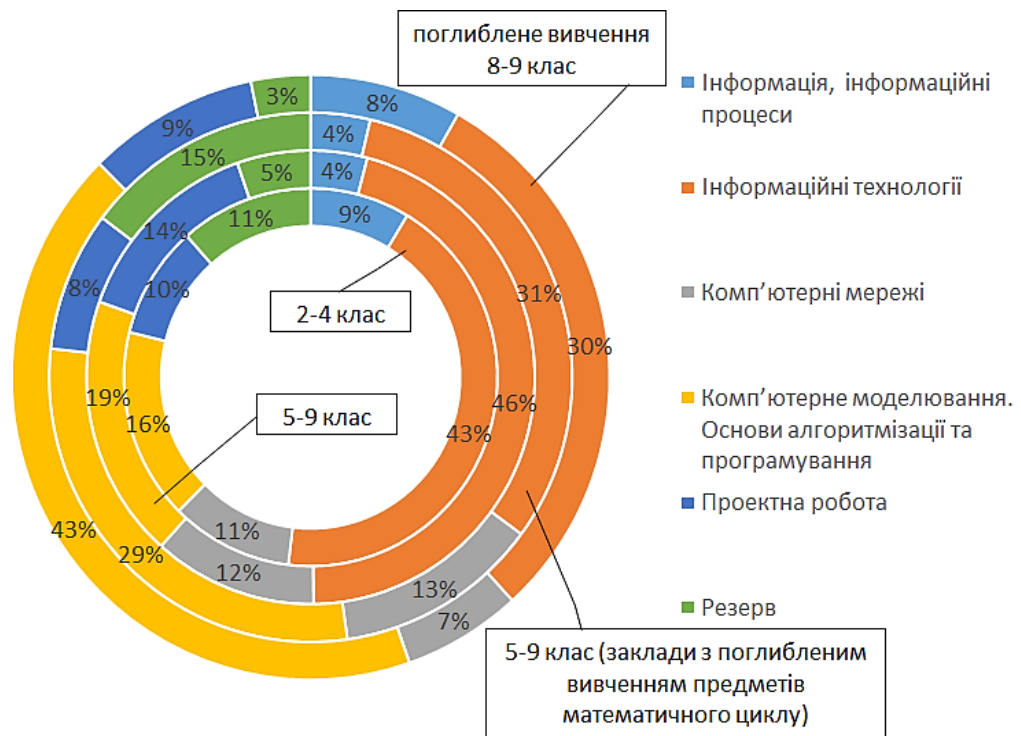


Рисунок 1.2 – Розподіл годин на вивчення інформатики в школах України (у відсотках)

У 2016 році почали вивчення інформатики у 5 класах учні, котрі вже вивчали її у 2-4 класі. Це призвело до чергової зміни змісту курсу [4].

У 2017–2018 н.р. до 10 класу прийшли учні, котрі вивчали інформатику в 5–9 класах, тому профільні курси для старшої школи теж піддалися корекції.

У 2019–2020 навчальному році курс «Інформатика» вивчається учнями загальноосвітніх шкіл України починаючи з 2 класу. Для середньої ланки навчальний курс сумарно розрахований на 245 годин і вивчається в межах інваріантної частини навчального плану (див. таблицю 1.1).

Таблиця 1.1 – Розподіл годин на вивчення курсу інформатики за класами

Клас	Кількість годин на тиждень	Загальна кількість годин
5 клас	1	35
6 клас	1	35
7 клас	1	35
8 клас	2	70
9 клас	2	70
Усього		245

*Зміст курсу інформатики* включає сукупність двох взаємопов'язаних компонентів: теоретичного і практичного [3].

*Теоретична частина* курсу спрямована на формування в учнів основ інформаційної культури, навичок аналізу і формалізації предметних задач, ознайомлення з такими поняттями як інформатизація, повідомлення, властивості інформації, інформаційні процеси, алгоритм, виконавець алгоритму, структура алгоритму, величина, типи величин.

*Практичний аспект* пов'язаний з виробленням навичок роботи з готовим програмним забезпеченням, написанням програм однією з

конкретних мов програмування, використання глобальної мережі Інтернет для обміну інформацією та повідомленнями, її пошуку.

На добір змісту шкільного курсу інформатики впливають дві групи основних чинників:

*Науковість і практичність.* Зміст навчального курсу інформатики повинен іти від науки інформатики (тобто не суперечити сучасному стану науки і бути методологічно витриманим); вивчення предмета повинно давати такий рівень фундаментальних знань учнів, який дійсно міг би забезпечувати підготовку учнів до майбутньої професійної діяльності в різних сферах (практична мета).

*Доступність і загальноосвітність.* Матеріал, який включається до курсу інформатики, має бути доступним для засвоєння учнем. Курс інформатики повинен, крім того, відображати найбільш загальнозначущі, загально культурні, загальноосвітні відомості з відповідної галузі наукових знань.

Для 10 –11 класів загальноосвітніх навчальних закладів була створена навчальна програма[5], розрахована на вивчення інформатики як вибірково-обов'язкового предмету навчального плану в обсязі до 105 годин, з яких 35 годин складає інваріантний базовий модуль (див. таблицю 1.2).

Таблиця 1.2 – Розподіл годин на вивчення курсу інформатики за класами

<b>Клас</b>	<b>Кількість годин на тиждень</b>	<b>Загальна кількість годин</b>
10 клас	1	35
11 клас	2	70
Усього		105

Програма має модульну структуру і складається з двох частин – базового та вибіркового (варіативних) модулів.

Модуль – структурна одиниця навчальної програми, подана як організаційно-методичний блок, що містить цілісний набір компетенцій, необхідних для засвоєння учнями протягом його вивчення.

Основою навчання інформатики в 10-11 класах є базовий модуль, зміст якого може бути розширений за рахунок вибіркового модуля. Базовий модуль, на вивчення якого відводиться 35 годин, завершує формування в учнів предметних і ключових компетентностей щодо використання сучасних інформаційно-комунікаційних технологій на рівні, визначеному чинним Державним стандартом базової і повної загальної середньої освіти. Цей модуль є мінімально допустимою нерозривною структурною одиницею програми, рознесення вивчення базового модуля на два роки не допускається.

Вибіркові модулі для розширення курсу учитель добирає, відповідно до профілю навчання навчального закладу, запитів, індивідуальних інтересів і здібностей учнів, регіональних особливостей, матеріально-технічної бази та наявного програмного забезпечення. Реалізація профільного навчання під час викладання курсу може здійснюватися як шляхом розширення змісту окремих тем, так і добором профільно-орієнтованих навчальних завдань.

*Організаційні форми навчання інформатики* в школі варіюються залежно від можливостей школи надати достатній арсенал засобів навчання. На сьогоднішній день існує 3 варіанти викладання інформатики у загальноосвітній школі:

- 1 варіант – “безмашинний”;
- 2 варіант – “частково машинний”;
- 3 варіант – “машинний”.

Це зумовлено наявністю комп’ютерної техніки у школі взагалі, або наявністю в школі комп’ютерів різних типів (машини старих поколінь і комп’ютери, що відповідають сучасним вимогам), а також кількістю комп’ютерів [3].

**1 варіант – “безмашинний”.**

**Мета курсу:** Формування знань, необхідних для майбутнього раціонального використання нових інформаційних технологій при розв’язуванні задач, що пов’язані з обробкою інформації, її пошуком, систематизацією, зберіганням, передачею інформації; знайомство учнів з роллю нових інформаційних технологій в сучасному виробництві, з перспективами розвитку обчислювальної техніки.

При виборі цього варіанту учні знайомляться з теоретичною частиною курсу на рівні представлення та орієнтації у поняттях, дізнаються про можливості програмного забезпечення ПК.

## **2 варіант викладання інформатики – “частково машинний”.**

**Мета курсу:** Формування знань, **практичних вмінь та навичок**, необхідних для майбутнього раціонального використання нових інформаційних технологій при розв’язуванні задач, що пов’язані з обробкою інформації, її пошуком, систематизацією, зберіганням, передачею інформації; знайомство учнів з роллю нових інформаційних технологій в сучасному виробництві, з перспективами розвитку обчислювальної техніки.

Окремо складаються програми теоретичного та практичного курсів, але обов’язково слід витримувати відповідність тематики теоретичних та практичних занять. Тому в залежності від наявності комп’ютерів певного класу можна робити деякі перестановки в чергуванні окремих тем.

При виборі такого варіанту викладення учні отримують теоретичні знання із шкільного курсу інформатики, а також частково оволодівають практичними вміннями та навичками роботи з операційною системою DOS та можливостями програми-оболонки Norton Commander, знайомляться з мовою програмування Basic чи Pascal та практикуються у написанні цими мовами та тестуванні програм. В учнів також є можливість познайомитися з операційною системою Windows та стандартними програмами, що входять до її складу для роботи з різними типами файлів. Окрім цього, якщо апаратне забезпечення кабінету інформатики є більш сучасним, то учні також

оволодівають знаннями та вміннями роботи з мережевими та інтерактивними технологіями.

### **3 варіант викладання інформатики – “машинний”.**

**Метою курсу** є формування теоретичної бази знань учнів з основ інформатики та практичних навичок використання засобів сучасних інформаційних технологій у повсякденній практичній, зокрема навчально-пізнавальній, діяльності учнів.

До **теоретичної бази знань** відносяться: уявлення про інформацію, її властивості, інформаційні процеси та інформаційні системи, загальні принципи розв'язування задач за допомогою комп'ютера з використанням прикладного програмного забезпечення загального та спеціального призначення, формулювання проблем і постановку задач, побудову відповідних інформаційних (зокрема, математичних) моделей, основи алгоритмізації і програмування, принципи будови та дії комп'ютера, можливості глобальної мережі Інтернет, пошук потрібної інформації.

До **практичних навичок** відносяться: навички роботи з пристроями введення-виведення інформації, прикладним програмним забезпеченням загального та цільового призначення: редакторами текстів, графічними редакторами, електронними таблицями, системами управління базами даних, інформаційно-пошуковими системами, педагогічними програмними засобами, програмами-браузерами для перегляду гіпертекстових сторінок; програмами для роботи з електронною поштою і телеконференціями; пошук інформації в глобальній мережі Інтернет; навички складання, описування та реалізації найпростіших алгоритмів і програм з використанням засобів навчальної алгоритмічної мови та реальних мов програмування і операційних систем.

Мета курсу досягається через практичне оволодіння учнями навичками роботи з основними складовими сучасного програмного забезпечення ЕОМ, ознайомлення з функціональним призначенням основних пристроїв комп'ютера та принципами їх будови і дії, основами технології розв'язування

задач за допомогою комп'ютера, починаючи від їх постановки й побудови відповідних інформаційних моделей і закінчуючи інтерпретацією результатів, отриманих за допомогою комп'ютера.

Програма курсу розрахована на вивчення основ інформатики за умов достатнього доступу учнів до комп'ютерів. Залежно від варіанту навчального плану, за яким працює школа курс інформатики може вивчатись у 8–9 класах або в 10–11.

Залежно від рівня оснащення комп'ютерною технікою в програмі пропонується два варіанти розподілу основних тем курсу:

**Перший варіант** тематичного планування розраховано на використання операційної системи *Windows*, в ньому також пропонується два варіанти орієнтовного планування теми «Основи алгоритмізації та програмування» — один варіант орієнтовано на вивчення процедурного програмування (мови *Basic* чи *Pascal*), другий — на об'єктне програмування (*VisualBasic* чи *Delphi* (меншою мірою)).

**Другий варіант** планування пропонується для навчання інформатики на базі комп'ютерів, які працюють під управлінням операційної системи *MS-DOS*.

Залежно від типу обчислювальної техніки, складу наявного науково-методичного та програмного забезпечення учитель може самостійно добирати методичні шляхи розв'язування освітніх завдань курсу, вносити необхідні корективи в порядок вивчення тем, а також змінювати кількість годин, необхідних для засвоєння програмного матеріалу. Окремі питання програми можуть вивчатися тільки в порядку ознайомлення. Відповідно до обраної методики вивчення курсу вчитель може дібрати відповідні навчальні посібники та дидактичне забезпечення, віддаючи перевагу тим чи іншим з них або ж певним чином поєднуючи їх.

Особливість викладання інформатики при машинному варіанті - вивчення даного предмету відбувається лише при умові розподілу учнів на

підгрупи на всіх уроках інформатики. Тобто немає уроків теорії та практики, а на кожному уроці учні працюють в комп'ютерному класі.

Планування даного варіанту вивчення інформатики відбувається відповідно до нової програми, але слід враховувати, що під час вивчення окремих тем курсу учні не зможуть працювати на комп'ютері, якщо техніка не відповідає сучасним вимогам. Наприклад, в комп'ютерному класі не має виходу до мережі Internet тощо. Тобто деякі теми слід вивчати на рівні ознайомлення.

### **1.3 Огляд нормативної та методичної бази для шкіл Європи та світу**

Дослідивши частково нормативно-методичну базу для шкіл країн Європи та світу можна стверджувати, що урок інформатики – це загальне визначення уроку з предмету, який у різних країнах має різну назву, і частіше за все різне смислове навантаження. В деяких країнах інформатика як шкільний предмет існує досить давно, але є й такі країни, де комп'ютерні технології вивчаються лише як спеціалізований курс в університеті чи коледжі. Розглянемо приклади з історії впровадження дисципліни «Інформатика» в навчальні програми в різних країнах Європи.

**Фінляндія.** Тільки в 2014 році Міністерство освіти Фінляндії заявило про плани по введенню основ програмування в програму початкової школи до 2016 року. На сьогоднішній день учні молодших класів опановують основні команди і структури, віддаючи команди іншим людям. У середній школі учні вивчають середу візуального програмування, таку як Scratch - це візуальна об'єктно-орієнтоване середовище програмування для навчання школярів молодших і середніх класів. Scratch створений як продовження ідей мови Logo і конструктора Lego. І тільки в старшій школі учні спробують роботу зі справжніми мовами програмування.



**Естонія.** Всі естонські школи були підключені до інтернету ще в кінці 90-х. У країні можна платити податки і голосувати через Інтернет і взагалі досягнуті цілком відчутні успіхи в побудові «Електронного Уряду». Але, не дивлячись на це, тільки в 2012 в Естонії була запущена програма вивчення основ програмування з 1 по 12 клас школи. Початковими учасниками програм стали 20 шкіл. У початковій школі діти вивчають основні логічні концепції і програмування в середовищі Scratch, а в середній та старшій ланці вдосконалюють свої знання, вміння та навички.

**Польща.** З вересня 2013 року в Польщі діє програма «Майстри Кодування», спрямована на навчання учнів початкової школи основам програмування. Для того, щоб взяти участь в програмі в школі повинно бути принаймні два вчителі, готових до участі в програмі, а також все необхідне обладнання. У перший рік дії цієї програми в ній взяли участь 120 шкіл і понад 6000 учнів. Для школярів 4-6 класу основою для вивчення програмування також стало середовище розробки і виконання алгоритмів Scratch.

**Великобританія.** Основним із основних центрів зародження інформатики як шкільної дисципліни серед європейських країн була Великобританія.

Британську освіту вважають однією з найкращих за якістю, тому що учні британських шкіл часто вражають результатами із загальних тестів. Двоє школярів з Великобританії зуміли перевершити видатних вчених Стівена Хокінга і Альберта Ейнштейна в тесті на визначення IQ-рівня. Це є своєрідною візиткою британських шкіл, що підтверджує заслужене першість в рейтингах.

На сучасний стан системи шкільної освіти Великобританії суттєво вплинув період, коли при владі перебувала партія консерваторів на чолі з М. Тетчер. Саме він став початком нового підходу до освіти, основними характеристиками якої стали гнучкість, мобільність і централізм [6].

Це було обумовлено тим, що за період з 1972 по 1977 рр. безробіття серед молоді у віці 16 і 17 років збільшилася на 120 відсотків і урядові кола, а також британські вчені в галузі освіти, були стурбовані тим, що школи і коледжі недостатньо добре готують випускників до сучасних професій, пов'язаних, зокрема, з використанням ІТ.

У зв'язку з цим, починаючи з 1979 р, Департамент Освіти і Науки (Department of Educational and Science, DES) проводив ряд пілотних проектів, метою яких була розробка єдиної Національної програми навчання для дітей у віці від 11 до 16 років. В результаті цієї роботи зросла переконаність в тому, що всі учні повинні навчатися за загальною програмою до досягнення шістнадцятирічного віку; при цьому вони повинні отримати різноманітні знання, підготовку до життя в суспільстві і необхідний для цього досвід.

З боку держави був також зроблений і ряд ініціатив по впровадженню в школах нових технологій. Перша з них відноситься до 1979 року і пов'язана з мікро-комп'ютерами. З тих пір в школах було запроваджено кілька схем з оновлення обладнання, в тому числі: стандартні настільні комп'ютери, портативні комп'ютери та CD-ROM. Як відзначають британські дослідники, вивчення курсу «Інформаційних технологій» (ІТ) з самого початку мало істотні відмінності від традиційних предметів, що вивчаються в початкових і середніх школах тим, що воно ніколи не відбувалося у відриві від них.

З тих часів Національна програма навчання реформувалася ще декілька разів разом із тим змінюючи методичні рекомендації та наповнення шкільних курсів, зокрема ІТ. При цьому найбільш повно ІТ вивчалися в предметній області «Технологія» і представляли самостійний розділ. У зв'язку з цим британськими методистами, вчителями та адміністрацією шкіл накопичений великий досвід вивчення та використання ІТ саме в різних предметах, а впровадження нового стандарту тільки зміцнило становище ІТ в школі, а не зруйнувало вже створену методичну систему в рамках всієї країни в галузі вивчення ІТ. Саме тому фахівців завжди цікавило питання про те, що ефективніше: вивчати ІТ, як окремий предмет з окремим викладачем, або

вивчення ІТ можливо якісно здійснювати в рамках інших предметів шкільної програми?

З точки зору британських фахівців немає єдино правильного шляху вивчення та використання ІТ в школі. Підходи, що успішно застосовуються в одній школі, можуть виявитися неефективними в іншій. Але проведені ними дослідження показали, що існують загальні проблеми, з якими стикаються всі школи в процесі освоєння вивчення ІТ. Прийнявши новий стандарт навчання школам дозволялося вводити новий для них предмет «Інформаційні технології» поступово, протягом декількох років. Згодом зміст цього предмету був розширений і отримав назву ІСТ (ІКТ – Інформаційні та комунікаційні технології).

У вересні 2014 року у школах Великобританії навчальний курс ІКТ був замінений на Computing (обчислення, технологія обчислень). Цей курс був розроблений за участі кваліфікованих спеціалістів з Microsoft та Google, а також робочої групи з Королівської інженерної академії та був введений на всіх ключових етапах державної навчальної програми.

Це відбулося через ряд причин. Уроки з ІКТ використовувалися виключно для навчання комп'ютерній грамоті, тобто учні оволодівали знаннями, як працювати з текстовим процесором, як створювати та редагувати електронні таблиці за допомогою табличного редактора і т.д. Цей підхід не враховував особливостей використання програмного забезпечення, яке вже застаріло.

Через вищезазначений факт уроки з ІКТ перетворювалися на нудні та некорисні, тому що в більшості випадків учні вже володіли багатьма знаннями та навичками роботи із ПК.

У шкільній програмі з ІКТ не було розділу «Програмування», тому школярів не знайомили з базовими алгоритмічними структурами, не наділялося достатньо уваги на розвинення в них логічного поетапного мислення та розвиток вмінь та навичок у написанні власних програм або створенні комп'ютерних ігор.

Технологічні компанії Великобританії скаржилися на нестачу достатньої кількості

Курс Computing має 3 основні складові [7]:

- а) computer science(CS) – комп'ютерна наука;
- б) information technology (IT) – інформаційні технології;
- в) digital literacy (DL) - цифрова грамотність.

Computer science (CS) – це розділ курсу Computing , де відбувається наукове та практичне вивчення обчислень, яке спрямоване на дослідження того,що можна обчислювати, як обчислювати та як обчислення можуть бути застосовані для вирішення повсякденних задач.

Information technology (IT) – розділ, в якому учні вивчають, як комп'ютери та телекомунікаційне обладнання працюють, як вони можуть бути застосовані при зберіганні, пошуку, передачі та маніпуляції з даними.

Digital literacy (DL) – це розділ курсу Computing, в якому школярі отримують змогу оволодіти здатністю ефективно, відповідально, безпечно та критично орієнтуватися, оцінювати і створювати цифрові одиниці матеріальної культури, використовуючи ряд цифрових технологій.

У програмі курсу Computing є три чіткі етапи [8]. Ключовий етап 1 (5 – 6-річні учні). Діти дізнаються про те, що таке алгоритми, не завжди залучаючись при цьому до роботи за комп'ютерами. На цьому етапі учні також створюють та налагоджують власні прості програми, розвиваючи навички логічного мислення та роблять перші кроки у використанні пристроїв для створення, організації, зберігання, маніпулювання та отримання цифрового контенту.

Ключовий етап 2 (7–11 років). Трохи старші діти початкової школи створюють та налагоджують складніші програми з конкретними цілями та знайомляться з поняттям змінної та вивчають особливості використання послідовності, вибору та повторення в програмному коді. Учні продовжують розвивати навички логічного міркування та поповнюють свій багаж знань та на практиці.

Ключовий етап 3 (11–14 років). У старшій школі учні використовують дві або більше мов програмування, принаймні одна з яких є текстовою, для створення власних програм. Школи та вчителі можуть обирати конкретні мови та засоби кодування. Учні також вивчають просту Булеву логіку (наприклад, оператори AND, OR чи NOT), працюють з двійковими числами та вивчають, як апаратне та програмне забезпечення комп'ютера працюють разом.

На всіх ключових етапах учні також знайомляться з поняттями безпеки для комп'ютерів та в мережі Інтернет, дізнаються, як повідомляти про занепокоєння щодо "вмісту чи контакту" в Інтернеті.

«Плюси» реформи. Перше – новий підхід до навчання: зміна результату підготовки зі значення «користувач» на значення «програміст».

По-друге, відтепер учень, який вивчає програмування з дитинства отримує величезну перевагу в порівнянні з тим, хто почав займатися цим лише в старших класах школи. Очевидно, що такому учневі буде легше отримати вищу освіту в цій галузі і вигідно продати свої знання на ринку високих технологій.

Третій аспект нової реформи – це логічне мислення, що розвивається у тих, хто навчається програмуванню з початкової школи. Воно стане в нагоді дітям незалежно від того, з якою професією вони заохочуть пов'язати своє життя. Такі навички, як уміння побудувати послідовність дій, оптимізувати свою діяльність, розбити задачу на кілька підзадач, не будуть зайвими ні для кого.

#### **1.4 Огляд і аналіз тематичного наповнення програм з інформатики**

Для дослідження даного питання були використані матеріали з навчальної програми шкільного курсу «Інформатика» для середньої та

старшої ланки загальноосвітніх шкіл в Україні та дані з державного навчального плану до курсу «Computing» для учнів початкової та середньої ланки в школах Великобританії.

**Україна.** Міністерство освіти і науки України затверджує типові освітні програми, які спрямовані на реалізацію мети та завдань освітньої галузі. Заклади загальної середньої освіти можуть використовувати типові або інші освітні програми. Освітні програми, що розробляються на основі типових освітніх програм, не потребують окремого затвердження центральним органом забезпечення якості освіти. На основі освітньої програми заклад освіти складає та затверджує навчальний план, який конкретизує організацію освітнього процесу [9].

Створюючи календарно-тематичне планування уроків з інформатики, вчитель спирається на запропоновану Міністерством освіти і науки України навчальну програму з даного курсу. З нею можна ознайомитися на сайті міністерства в розділі «Навчальні програми», обравши відповідно підрозділ, що орієнтований на конкретні класи (наприклад: Навчальні програми для 5-9 класів, або Навчальні програми для 10-11 класів).

Навчальна програма з курсу «Інформатика» існує в двох варіантах, через те, що одна програма розрахована на учнів, які до 5 класу не вивчали інформатику, інша – для учнів, що вивчали цей предмет з 2 класу. Кожна програма містить пояснювальну записку, в якій визначено мету та завдання навчання інформатики, а також зміст навчального матеріалу та вимог щодо рівня навчальних досягнень учнів.

Відповідно до Державного стандарту базової і повної загальної середньої освіти курс «Інформатика» будується за такими змістовими лініями [10]:

- а) інформація, інформаційні процеси, системи;
- б) комп'ютер як універсальний пристрій для опрацювання даних;

в) комп'ютерні мережі, інформаційні технології створення та опрацювання текстових документів, графічних зображень, числових даних, об'єктів мультимедіа, комп'ютерних презентацій;

г) комп'ютерне моделювання;

д) основи алгоритмізації та програмування.

Навчальна програма до курсу Інформатика побудована *лінійно-концентрично*. Зміст понять поступово розширюється і доповнюється. Лінійність реалізується шляхом ознайомлення учнів з поняттями інформації, інформаційних процесів, систем і технологій, інформаційної моделі та комп'ютерним моделюванням, операційною системою й прикладними програмами захисту та архівування даних, графічним редактором, текстовим і табличним процесорами, засобами створення та опрацювання публікацій, редакторами презентацій й об'єктів мультимедіа, системою управління базами даних, сервісами Інтернету, поняттям алгоритму, базовими структурами алгоритмів, навчальним середовищем виконання алгоритмів. Змістова лінія «Основи алгоритмізації та програмування» є наскрізною для всього курсу.

Концентричність реалізує ознайомлення учнів з поняттями інформатики і інформаційно-комунікаційними технологіями за всіма змістовими лініями на різних рівнях складності, поступово доповнюючи і розширюючи їх зміст залежно від рівня сформованості загально-навчальних навичок, вивченого навчального матеріалу з інших предметів і вікових особливостей розвитку учнів відповідних класів. Таким чином забезпечується поступове нарощування складності матеріалу, його актуалізація, повторення, закріплення, що сприяє формуванню предметної ІКТ-компетентності та ключових компетентностей і способів діяльності на більш високому рівні:

Перший рівень (5–7 класи) – ознайомлення з базовими поняттями курсу, формування орієнтувальної основи дій щодо роботи з персональним комп'ютером, комп'ютерними мережами, інформаційними технологіями,

навчальним середовищем виконання алгоритмів, формування предметної ІКТ-компетентності та ключових компетентностей під час виконання репродуктивних і проблемних завдань, зокрема індивідуальних навчальних проєктів та компетентнісних задач, виконання яких передбачає використання однієї з інформаційних технологій або програмного середовища.

Другий рівень (8–9 класи) – формування предметної ІКТ-компетентності та ключових компетентностей при виконанні репродуктивних, проблемних і евристичних (частково-пошукових) завдань, зокрема індивідуальних і групових проєктів, компетентнісних задач, виконання яких передбачає використання кількох різних інформаційних технологій або програмних середовищ.

У програмі конкретизовано зміст навчального матеріалу для кожного класу і подано відповідні вимоги до навчальних досягнень учнів. Перелік вимог зорієнтує вчителя на досягнення мети навчання за кожною темою програми, полегшить планування мети і завдань навчання на уроках, надасть змогу виробити адекватні методичні підходи до проведення навчальних занять, поточного й тематичного оцінювання. Зміст навчання інформатики структуровано за темами із визначенням кількості годин на їх вивчення (див. таблицю 1.3). Такий розподіл змісту і навчального часу є орієнтовним. Учителю та авторам підручників надається право коригувати послідовність вивчення тем залежно від методичної концепції та конкретних навчальних ситуацій, від рівня підготовки учнів і сформованості у них предметної ІКТ-компетентності, вибудовуючи найбільш доречну для конкретного навчального закладу або класу траєкторію навчання.

Таблиця 1.3 – Розподіл навчальних годин на вивчення розділів програми

№	Назва розділу	Класи і кількість годин					Всього
		5 кл.	6 кл.	7 кл.	8 кл.	9 кл.	
1	Інформація, інформаційні процеси, системи, технології	4	–	–	3	2	9



## Продовження таблиці 1.3

№	Назва розділу	Класи і кількість годин					Всього
		5 кл.	6 кл.	7 кл.	8 кл.	9 кл.	
2	Комп'ютер як універсальний пристрій для опрацювання даних	10	6	–	5	–	21
3	<b>Інформаційні технології</b>						
3.1	Створення та опрацювання текстових документів	–	8	–	6	5	19
3.2	Створення та опрацювання графічних зображень	9	–	–	–	6	15
3.3	Створення та опрацювання об'єктів мультимедіа	–	4	–	6	–	10
3.4	Створення та опрацювання комп'ютерних презентацій	9	–	–	–	6	15
3.5	Створення та опрацювання числових даних	–	–	8	10	–	18
4	Комп'ютерні мережі	–	8	4	–	16	28
5	Моделювання	–	–	3	–	6	9
6	Основи алгоритмізації та програмування	–	7	9	28	10	54
7	Розв'язування компетентнісних задач, виконання індивідуальних і групових навчальних проєктів	–	–	8	9	15	32
8	Резерв	3	2	3	3	4	15
Всього	35	35	35	70		70	245

Водночас учитель не може порушувати порядок вивчення тем, між якими є змістові залежності (див. рис. 1.3). Запропонована кількість часу на вивчення кожної теми також є орієнтовною, вчитель може її змінювати (до 15 % від загального навчального часу для кожного класу).

При цьому вчитель має забезпечити рівень навчальних досягнень учнів, зазначених у програмі з кожної теми.

**Великобританія.** Обчислювальна техніка має глибокі зв'язки з математикою, наукою, а також з дизайном і технологіями, і дає розуміння як природних, так і штучних систем.

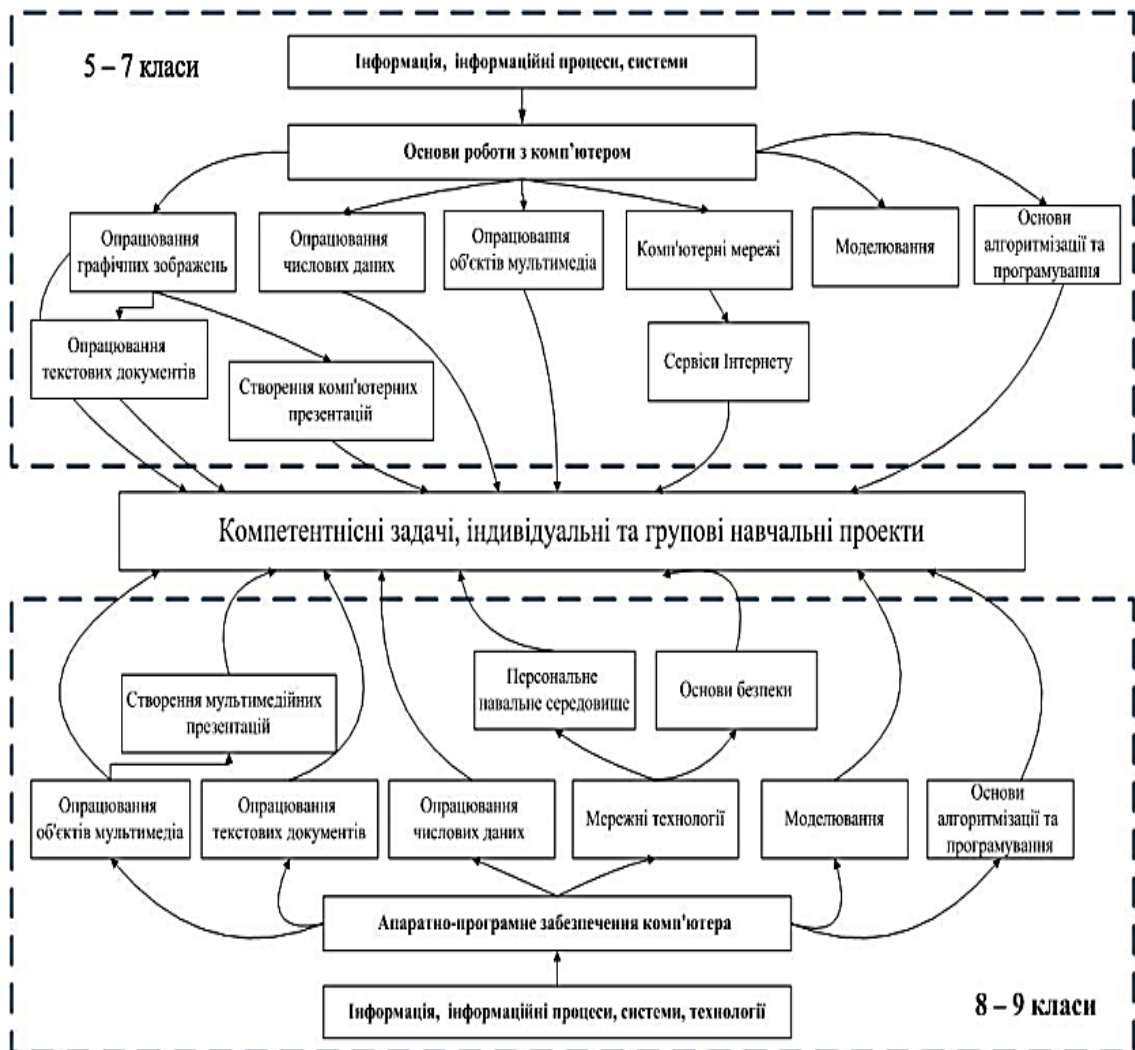


Рисунок 1.3 – Змістові залежності між темами курсу інформатики

Основою обчислювальної техніки є інформатика, в якій учнів навчають принципам інформації та обчислень, як працюють цифрові системи та як ці знання використовувати для програмування. Обчислювальна техніка також забезпечує цифрову грамотність учнів, коли школярі вміють користуватися та висловлювати себе та розвивати свої ідеї за допомогою інформаційно-комунікаційних технологій на придатному для майбутнього робочого місця та як активних учасників цифрового світу рівні.

Національна навчальна програма з курсу «Computing» має на меті забезпечити:

- а) розуміння та застосовування основних принципів та концепції інформатики, включаючи абстрагування, логіку, алгоритми та представлення даних;
- б) вміння аналізувати проблеми в обчислювальному відношенні та неодноразовий практичний досвід написання комп'ютерних програм для вирішення таких проблем;
- в) можливість оцінювати та застосовувати інформаційні технології, включаючи нові або незнайомі технології, аналітично для вирішення проблем;
- г) виховання відповідальних, компетентних, впевнених та креативних користувачів інформаційно-комунікаційних технологій.

Після закінчення кожного ключового етапу учні повинні знати, застосовувати та розуміти питання, вміння та процеси, визначені у відповідній програмі навчання.

Для учнів початкової (див. таблицю 1.4) та середньої (див. таблицю 1.5) школи в державному навчальному плані для шкіл Великобританії передбачений такий предметний зміст курсу «Computing» з точки зору 3 основних напрямків курсу: комп'ютерної науки, ІТ та цифрової грамотності (див. таблицю 1.4).

Програма навчання є мінімальним набором рекомендацій, який не обмежує тематичного охоплення курсу «Computing». Програма навчання не є схемою роботи, тобто вчителю надається свобода у виборі контенту, порядку, контексту та з ресурсів, які наповнюватимуть навчальний процес. Схеми роботи, зазначені в національній програмі не є планами уроків.

Цей рівень планування розроблюється пізніше вчителем самостійно, залежно від ідеї, конкретних цілей та ресурсів. Зараз набагато більша увага приділяється вивченню комп'ютерів та технологій обчислення, а не навчанню до використання технологій.

Таблиця 1.4 – Опис змістового наповнення розділів програми з курсу “Computing” для учнів початкової школи

	KS1	KS2
CS	<p>Зрозуміти, що таке алгоритми; як вони реалізуються як програми на цифрових пристроях; і що програми виконуються, дотримуючись точних та однозначних інструкцій. Створення та налагодження простих програм. Використовуйте логічні міркування для прогнозування поведінки простих програм</p>	<p>Розробляти, записувати та налагоджувати програми, які досягають конкретних цілей, включаючи контроль або моделювання фізичних систем; розв’язувати задачі, розкладаючи їх на менші частини.</p> <p>Використовувати послідовність, вибір та повторення в програмах; робота зі змінними та різними формами введення та виводу.</p> <p>Використовуйте логічні міркування, щоб пояснити, як працюють деякі прості алгоритми, а також виявити та виправити помилки в алгоритмах та програмах.</p> <p>Розуміти комп’ютерні мережі, включаючи Інтернет; як вони можуть надавати декілька послуг, таких як всесвітня павутина</p> <p>Оцініть спосіб вибору та ранжирування результатів [пошуку]</p>
IT	<p>Використовувати технологію цілеспрямовано для створення, організації, зберігання, маніпулювання та отримання цифрового вмісту</p>	<p>Використовуйте пошукові технології ефективно. Вибір, використання та комбінування різноманітного програмного забезпечення (включаючи Інтернет-сервіси) на різноманітних цифрових пристроях для розробки та створення цілого ряду програм, систем та контенту для досягнення поставлених цілей, включаючи збір, аналіз, оцінку та представлення даних та інформації</p>

## Продовження таблиці 1.4

KS1		KS2
DL	<p>Визнати поширене використання інформаційних технологій поза школою</p> <p>Використовувати технологію безпечно та з повагою, зберігаючи особисту інформацію приватною; визначте, куди звертатися за допомогою та підтримкою, коли у них є проблеми щодо вмісту чи контактів в Інтернеті чи інших інтернет-технологіях</p>	<p>Розуміти можливості [мереж], що пропонуються для спілкування та співпраці</p> <p>Будьте вибагливі в оцінці цифрового контенту</p> <p>Використовувати технологію безпечно, шанобливо та відповідально; визнати прийнятною / непринятною поведінку; визначити цілий ряд способів повідомляти про занепокоєння щодо вмісту та контактів</p>

Таблиця 1.5 – Опис змістового наповнення розділів програми з курсу “Computing” для учнів середньої ланки

KS3	
CS	<p>Розробити, використовувати та оцінити обчислювальні абстракції, що моделюють стан та поведінку проблем у реальному світі та фізичних систем</p> <p>Зрозуміти декілька ключових алгоритмів, що відображають обчислювальне мислення [наприклад, алгоритми сортування та пошуку]; використовувати логічні міркування для порівняння корисності альтернативних алгоритмів для тієї ж проблеми</p> <p>Використовуйте дві або більше мов програмування, принаймні</p>

## Продовження таблиці 1.5

KS3	
CS	<p>одна з яких є текстовою для вирішення різноманітних обчислювальних задач; правильно використовувати структури даних [наприклад, списки, таблиці або масиви]; розробляти та розробляти модульні програми, що використовують процедури чи функції</p> <p>Зрозуміти просту логічну логіку [наприклад, AND, OR і NOT] та деякі її використання в схемах та програмуванні; зрозуміти, як числа можуть бути представлені у двійкових, і бути в змозі виконувати операції над двійковими числами [наприклад, двійкове додавання та перетворення між двійковими і десятковими]. Зрозуміти апаратні та програмні компоненти, що складають комп'ютерні системи, і те, як вони спілкуються між собою та з іншими системами</p> <p>Зрозуміти, як інструкції зберігаються та виконуються в комп'ютерній системі; зрозуміти, як дані різних типів (включаючи текст, звуки та зображення) можуть бути представлені та оброблені цифровим шляхом у вигляді двійкових цифр</p>
IT	<p>Виконувати творчі проекти, які передбачають вибір, використання та комбінування декількох додатків, бажано на різних пристроях, для досягнення складних цілей, включаючи збір та аналіз даних та задоволення потреб відомих користувачів</p> <p>Створення, повторне використання, перегляд та повторне призначення цифрових артефактів для даної аудиторії з урахуванням надійності, дизайну та зручності використання</p>
DL	<p>Розуміти цілий ряд способів безпечного, відповідального та безпечного використання технології, включаючи захист їх ідентичності та конфіденційності в Інтернеті; розпізнавати невідповідний вміст, контактувати та вести та знати, як повідомити про проблеми</p>

## **1.5 Огляд і аналіз розбіжностей методичних матеріалів для викладання інформатики в Україні та Світі**

Для підготовки до проведення уроків з інформатики в середній ланці загальноосвітньої школи в Україні вчитель повинен мати необхідні йому ресурси. До них відносяться і методичні рекомендації щодо проведення уроків з конкретної теми, і наочний та роздатковий матеріал, і посібники з інформатики, до яких можна віднести і підручники з цього предмету для різних класів. Ніхто не має права обмежувати вчителя у його прагненні застосувати обрані ним методичні матеріали. Тому для побудови конспекту уроку або позакласного заходу з інформатики вчителі можуть використовувати будь-які ресурси.

В ті часи, коли інтернет не був настільки доступним, вчителі користувалися послугами бібліотек і власною фантазією. Сьогодні ж, коли є можливість вільного доступу до мережі інтернет вчителям стало набагато легше працювати. По-перше, на просторах інтернету можна знайти готові конспекти до уроків, які підкріплюються мультимедійною презентацією. Це значно полегшує підготовку вчителя та скорочує час на неї. По-друге, набагато простішим став алгоритм розробки роздаткового матеріалу та наочності для уроків з інформатики, тому що на сьогодні завантажити зображення або схему, або скриншот чи логотип програми набагато легше, аніж робити все це вручну. Багато хто з вчителів навіть розробляють власні веб-ресурси, де викладають матеріали зі своїх уроків, які можуть стати у пригоді їх колегам (див. рис. 1.5).

Такий метод підготовки до уроків може використовуватися і вчителями з інших країн. Досліджуючи нормативно-методичні документи для викладання курсу “Computing” у Вєдикобританії ми помітили, що вчителям надаються рекомендації та навіть активні посилання на ресурси безпосередньо в тексті гїду по національному плану для вчителів середньої

ланки (Computing in the national curriculum. A guide for secondary teachers)  
(див. рис.1.6).

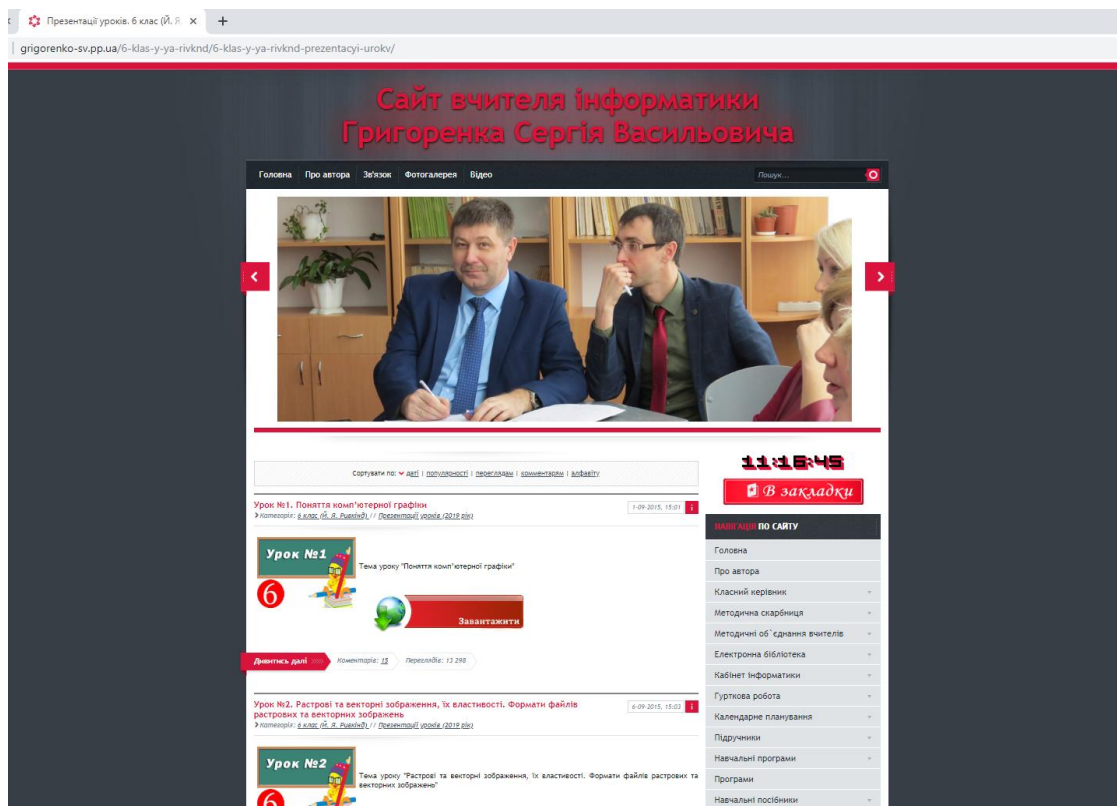


Рисунок 1.5 – Інтерфейс сайту вчителя з інформатики

В цьому офіційному документі вчителі можуть знайти багато корисної інформації і про зміст предмету, який вони викладають, і про підходи до викладання (методи, принципи, форми організації навчального процесу та ін.). За допомогою активних посилань в тексті, вчителі британських шкіл можуть навіть перейти на запропонований сайт із софтом та завантажити необхідні дистрибутиви.



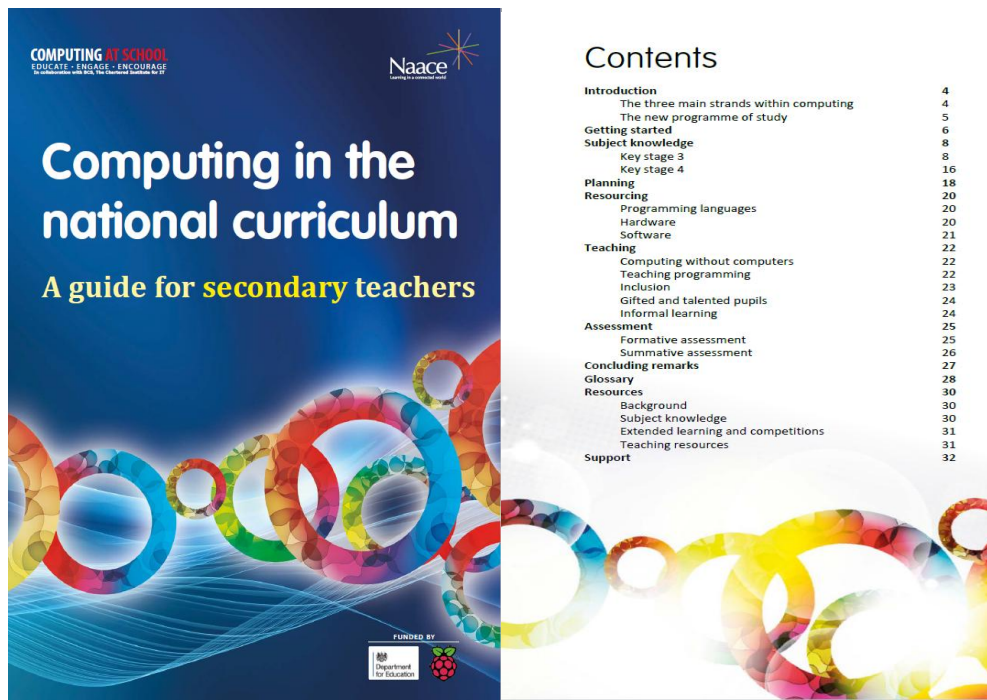


Рисунок 1.6 – Computing in the national curriculum. A guide for secondary teachers

## 1.6 Здобуття термінів з текстів як метод тематичного дослідження домену

Здобуття набору термінів із колекції документів, що описує предметну область, є важливим початковим кроком у з'ясуванні того, що існує повний набір вимог побудови онтології для домену. Результат цього кроку буде корисним лише у тому випадку, якщо вихідна колекція документів буде достатньо повною. В іншому випадку важливі терміни можуть бути пропущені.

Простий спосіб зібрати повну колекцію - це отримати всі доступні документи. На жаль, це не є можливим через різні доступності та величезну кількість джерел у реальних областях. Спосіб зменшення розміру колекції, що підлягає обробці, зберігаючи при цьому повноту набору термінів – це здобуття термінологічно насиченого піднабору документів, підколекції, яку називають *термінологічним ядром* [11].

OntoElect, як методологія, дозволяє максимізувати відповідність онтології тому, що стейкхолдери знань у домені думають про цей домен. Відповідність вимірюється як «голоси» стейкхолдерів – це показник, який дозволяє оцінити прихильність стейкхолдерів (stakeholder commitment) до розробленої онтології – відображаючи, наскільки добре враховується їх думка про потреби. Чим більше голосів зібрано, тим вищою буде їх прихильність. Якщо буде отримана критична маса голосів (скажімо, 50%+1, тобто проста більшість голосів), то онтологія вважається задовільною відповідним потребам. Важливою особливістю, яка повинна бути забезпечена для здобуття знань з текстових колекцій, є те, що набір документів (dataset) повинен бути статистично репрезентативним, щоб повністю задовольняти думки стейкхолдерів знань з предметної області. OntoElect пропонує метод вимірювання термінологічної повноти колекції документів шляхом аналізу насиченості термінологічних слідів інкрементально збільшуваних частин колекції документів [12]. Детальний алгоритм роботи за вказаною методологією описаний в пункті 2 та 3 другого розділу.

### **1.7 Мета і завдання роботи, постановка мети роботи та завдань**

**Метою** даної роботи є дослідження повноти охоплення питань з методики викладання інформатики та її дидактичних засад на основі структурного аналізу термінологічного наповнення тексту. Вхідними даними стали збірники статей за всі роки проведення міжнародної конференції ISSEP.

Відповідно до мети визначаються наступні завдання дослідження:

а) проаналізувати сучасний стан і проблеми вивчення і використання потенціалу курсу інформатики, що розглядаються на

міжнародному рівні для вітчизняної системи освіти так і для освіти за кордоном;

б) на основі комплексного аналізу науково-педагогічної літератури, нормативних документів та інших джерел виявити фактори, під впливом яких формувалися методичні підходи до вивчення інформаційних технологій в українських школах та школах інших країн Європи, зокрема Британії;

в) провести порівняльний аналіз змістових ліній освітньої галузі «інформатика» і предмета «інформаційні технології», визначених у стандартах України та Великобританії ;

г) побудувати термінологічну онтологію, дослідити її на повноту, зробивши висновок про глибину вивчення питань викладання «інформатики», як шкільного предмета, проаналізувати повноту висвітлення питань методологічного підходу у вітчизняному викладанні шкільного курсу «інформатика» та закордоном.

## **1.8 Висновок**

В першому розділі було розглянуто історію становлення інформатики як науки та шкільного курсу. Були висвітлені питання, що стосуються різного підходу до викладання та змістового наповнення курсу інформатики в українських загальноосвітніх школах та у школах країн Європи, зокрема Великобританії.

В даному розділі був проаналізований аспект здобуття термінів з текстів як метод тематичного дослідження домену Також була встановлена чітка мета роботи та аргументовані завдання до роботи.

## **2 ЗБІР ТА ОБРОБКА НАСИЧЕНОЇ ТЕРМІНОЛОГІЧНОЇ ІНФОРМАЦІЇ У ДОМЕНІ “ШКІЛЬНА ІНФОРМАТИКА”**

### **2.1 Пошук та вибір колекції наукових статей для термінологічного аналізу**

Дана робота спрямована на вивчення особливостей змістового наповнення та методичних підходів до викладання шкільного курсу «Інформатика» на основі розробленої формальної моделі представлення знань в вищезазначеній предметній області. Для аналізу були обрані міжнародної конференції з питань інформатики ISSEP.

ISSEP – International Conference on Informatics in Secondary Schools – Evolution and Perspectives (Інформатика в середніх школах – еволюція та перспективи) – це форум для дослідників та практиків у галузі інформатики, як в початкових, так і в середніх школах. Конференція надає можливість викладачам та дослідникам обговорити та розглянути питання про цілі та завдання цього предмету, його навчальні програми, різні парадигми та теми викладання та навчання, а також зв'язки із повсякденним життям - включаючи різні способи розвитку інформатики в школах. Це більше стосується дидактики інформатики. ISSEP надає можливість підбити підсумки в цій галузі та наскільки ці напрацювання є актуальними для повсякденного шкільного життя. Ця конференція об'єднує делегатів з різних країн для вирішення нагальних питань в галузі інформатики. На основі цих складних напрямків учасники розглядають, як впоратися з майбутніми потребами освіти з інформатики. Вчителі, ті, хто працює в освіті вчителів і професійному навчанні, а також у шкільній адміністрації, беруть участь у семінарах та діляться своїм досвідом чи баченням на міжнародній платформі.

Конференція ISSEP розпочалась у 2005 році в Австрії, Клагенфуртському університеті (нині Клагенфуртський університет імені

Альпен-Адрія), де конференція була частиною святкування 20 років викладання інформатики в австрійських середніх школах. Наступного року ISSEP відбулася у Вільнюсі, Литва, відзначивши 20 років інформатики в литовських середніх школах. У 2008 році ISSEP був у Торуні, Польща, а у 2010 конференція ISSEP була організована у Цюриху. Далі слідували Братислава, Словаччина, у 2011 році, Ольденбург, Німеччина, у 2013 році, Стамбул, Туреччина, у 2014 році, Любляна, Словенія, у 2015 році та Мюнстер, Німеччина, у 2016 році. Міжнародна конференція з інформатики в школах: ситуація, еволюція та перспективи (ISSEP) відсвяткувала своє 10-річчя в Гельсінкі, столиці Фінляндії. 2017 рік також відзначив річницю Фінляндії. Через місяць після конференції 6 грудня Фінляндія відзначила свій 100-й рік незалежності.

Кожного разу в ISSEP приймає участь все більше і більше спеціалістів, що висвітлюють проблемні питання в області інформатики. Теми, що пропонуються на обговорення дуже різні і стосуються полярно різних аспектів процесу викладання інформатики як дисципліни і самих процесів підготовки вчителів до уроків з інформатики, яка включає спочатку і їх педагогічну освіту з а цим напрямком.

## **2.2 Розробка процедурного пайплайну для здобуття насиченого набору термінів**

Підхід до вимірювання термінологічної насиченості базується на використанні алгоритму THD, розробленого на основі методології OntoElect для поліпшення онтологій.

Повні тексти документів із ретроспективної колекції групуються у набори даних в порядку їх часових міток. Перший набір даних D1 містить першу частину (inc) документів (див. рис. 2.1). Другий набір даних D2 містить перший набір даних D1, а також другу інкрементальну частину (inc)

документів. Нарешті, останній набір даних  $D_n$  містить всі документи з колекції [12].

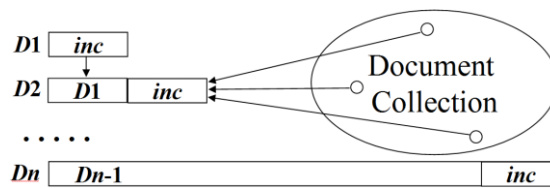


Рисунок 2.1 – Інкрементальне збільшення наборів даних у OntoElect [12]

На наступному кроці робочого процесу OntoElect, набори багаторазових термінів  $B_1, B_2, \dots, B_n$  здобуваються з наборів даних  $D_1, D_2, \dots, D_n$ , за допомогою програмного забезпечення TerMine, разом зі значеннями їх важливості (C-value). Ці оцінки значною мірою співвідносяться з частотами термінів (term frequencies), тобто як часто зустрічається термін у наборі даних. Приклад здобутого набору термінів за допомогою TerMine (див. рис. 2.2).

C-Value	Frequency	Term
1587.8446	1590	temporal logic
1306.8312	1308	de nition
777.0000	778	temporal representation
725.5102	727	computer science
659.5835	661	speci cation
649.4886	652	temporal constraint

Рисунок 2.2 – Приклад наборів термінів, здобутих TerMine [12]

На наступному етапі кожний здобутий набір термінів  $B_i, i = 1, \dots, n$  обробляється наступним чином:

а) для кожного окремого терміна обчислюються нормалізовані значення важливості (normalized scores) термінів:

$$n\text{-score} = C\text{-value} / \max(C\text{-value});$$

б) обчислюється порог важливості (eps) окремих термінів, щоб відкинути ті терміни, які не входять до складу більшості голосів. Сума  $n$ -

scores зі значеннями вище  $eps$  складає більшість голосів, якщо ця сума вище, ніж  $\frac{1}{2}$  суми всіх  $n$ -scores;

в) виконується відсікання (cut-off) при  $n$ -score  $< eps$ ;

г) результат зберігається в  $T_i$ .

Після цього кроку лише суттєві терміни, чії  $n$ -scores представляють більшість голосів, зберігаються у наборах термінів.

Потім  $T_i$  оцінюють для насичення шляхом вимірювання парної термінологічної різниці між наступними наборами  $T_i$  та  $T_{i+1}$ ,  $i = 0, \dots, n-1$ . Це робиться шляхом застосування алгоритму THD (див. рис. 2.3)

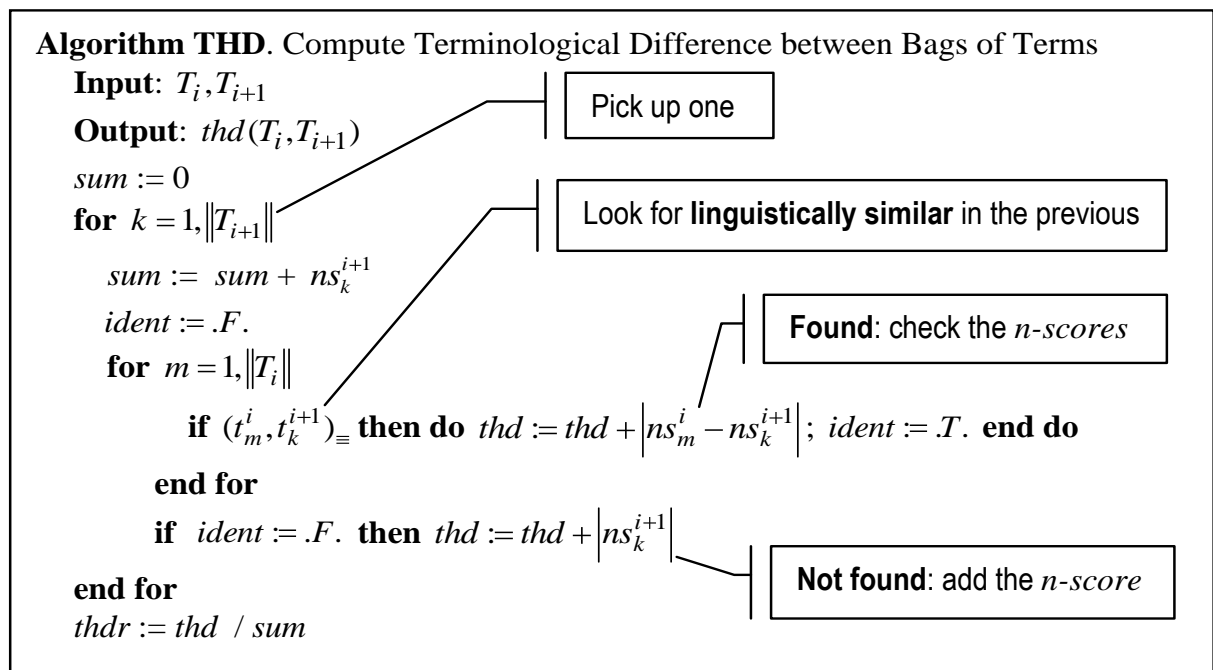


Рисунок 2.3 – THD алгоритм для порівняння пар наборів термінів [12]

Фактично, THD накоплює розбіжності у  $n$ -scores для набору  $T_{i+1}$  у значенні  $thd$ , якщо в  $T_i$  та  $T_{i+1}$  були лінгвістично однакові терміни. Якщо в  $T_i$  не було такого ж терміну, він додає  $n$ -score до значення  $thd$  з  $T_{i+1}$ . Після того, як  $thd$  було обчислено, відносна термінологічна різниця  $thdr$  отримує своє значення як  $thd$ , поділене на суму  $n$ -scores у  $T_{i+1}$ .

Абсолютні ( $thd$ ) та відносні ( $thdr$ ) термінологічні відмінності обчислюються для подальшої оцінки, якщо  $T_{i+1}$  відрізняється від  $T_i$  більше, ніж значення індивідуального порогу значущості  $eps$ . Якщо це не так, це

означає, що додавання інкрементних документів у  $D_i$  для отримання  $D_i + 1$  не сприяло ніякій помітній кількості нової термінології. Таким чином, підмножина  $D_i + 1$  загальної колекції документів можливо буде термінологічно насиченою. Однак, для отримання більшої впевненості в насиченні, OntoElect пропонує, що оцінюються ще декілька наступних пар  $T_i$  і  $T_i + 1$ . Якщо спостерігається стабільна насиченість, то процес пошуку мінімальної насиченої підколекції може бути зупинений. Однак іноді, термінологічний пік може мати місце вже після того, як насиченість спостерігалася у попередніх парах  $T$ . Як правило, цей пік вказує на те, що в інкремент був доданий високоінноваційний документ зі значною кількістю нових термінів. Використання OntoElect на прикладі оцінки насиченості для колекції документів ISSEP (див. Додаток А).

Термінологічні піки корелюють з доданими часто цитованими статтями. Варто також помітити, що кількість збережених термінів у  $T$  значно нижча, ніж кількість здобутих термінів у  $B$ .

Підхід OntoElect для оцінки термінологічної насиченості, неконтрольований та незалежний від домену завдяки використанню TerMine для здобуття термінів. Однак залежність OntoElect від цього інструменту є суттєвим недоліком.

### **2.3 Підготовка колекції для термінологічного аналізу. Розробка каталогу, пошук кількостей цитувань, перетворення у текст**

Необхідні для даної роботи статті знаходяться у вільному доступі у вигляді збірників, що мають назву відповідно до теми конференції та року проведення міжнародної конференції ISSEP. Кожен збірник був завантажений та збережений у форматі Portable Document Format (PDF) [13] – відкритий формат файлу, створений і підтримуваний компанією Adobe Systems, для представлення двовимірних документів у незалежному від



пристрою виведення та роздільної здатності вигляді. Для побудови колекції текстів був проведений процес роз'єднання збірників на окремі статті за допомогою програмного продукту Acrobat DC. Кожна окрема стаття була збережена в окремій директорії відповідно до року проведення конференції у хмарному сховищі Google Drive (див. Додаток Б).

Після того, як всі статті були збережені у хмарному сховищі розпочався процес розробки каталогу даних. Для створення каталогу було обрано програмну оболонку Microsoft Excel, як провідну програму для складання електронних таблиць у певній галузі, а також потужний інструмент для візуалізації й аналізу даних.

Каталог містить 204 рядки (див. Додаток В) із даними про кожну статтю з конференції. У стовці “Year” зазначається рік видавництва статті, стовбець “Journal” містить в собі назви збірників, в яких містяться статті, для зазначення на яких саме сторінках у збірнику знаходиться стаття був сформований стовбець “Pages”. У стовбцях “DOI” та “DOI link” вказані значення DOI (digital object identifier) - цифрового ідентифікатора об'єкта, який присвоюється науковим статтям і збірникам, а також активне посилання на сторінку веб-ресурсу SpringerLink, де можна ознайомитися з фрагментом тексту статті. У стовбці “Paper Title” вказані назви статей, стовбець “Abstract” містить тексти передмови статей, а в стовбці “Key phrases” можна знайти всі ключові слова кожної зі статей. Для того, щоб дізнатися точну кількість цитувань тексту кожної окремої статті ми скористалися інструментом Google Scholar або Google Академія – це вільнодоступна пошукова система, яка індексує повний текст наукових публікацій всіх форматів і дисциплін. У стовбці “Citations (GS)” було зазначено кількість цитувань кожної зі статей каталогу. Для розрахунку кількості цитувань на рік, було використано формулу :

$$\frac{\text{кількість цитувань}}{\text{поточний рік} - \text{рік видавництва} + 1} \quad (2.1)$$

Значення отримані в результаті розрахунків були розміщені в стовбці “Citations per year”. Кожному файлу статті надавалося унікальне ім'я, яке було створено за правилом :

$$\begin{aligned} & \langle \text{ID журналу} \rangle + "-" + \langle \text{рік} \rangle + \langle \text{том} \rangle + "(" + \langle \text{випуск} \rangle + ")" \\ & - "(" + \langle \text{сторінки} \rangle + ")" -" + \langle \text{DOI} \rangle .\text{txt}, \end{aligned} \quad (2.2)$$

де  $\langle \text{ID журналу} \rangle$  - назва директорії, в якій розміщено статтю,  $\langle \text{рік} \rangle$  – рік видавництва збірника,  $\langle \text{том} \rangle$  - номер тому збірника,  $"(" + \langle \text{випуск} \rangle + ")"$  – номер випуску збірника,  $+ \langle \text{сторінки} \rangle +$  – сторінки, на яких розміщений текст статті в конкретному збірнику,  $\langle \text{DOI} \rangle .\text{txt}$  – імя файлу статті, сформоване за допомогою її цифрового ідентифікатора об'єкта та розширення “.txt”. Всі значення імен файлів статей були розміщені в стовбці “paper file name (KM DropBox)”, а стовбець “Link” – був створений для розміщення активних посилань на повний текст статей, що розміщені на Google Drive.

Всі файли статей перетворювалися з формату PDF у формат TXT вручну із використанням Acrobat DC. Для перетворення файлів статей у текстовий формат за допомогою програмного продукту Acrobat DC було використано функцію «Експортувати в ...» обраний формат. Колекція була вручну перетворена в звичайний текстовий формат ANSI.

Наступним етапом роботи було формування набору датасетів з текстових файлів статей. Датасет - це текстовий файл, який містить в собі тексти статей, які в нього входять. Кожен наступний датасет  $D_i$  (крім першого) складається з попереднього  $D_{i-1}$  і інкремента. Інкремент складається з текстів кількох статей. З отриманих текстів статей ми створили 21 датасет,  $D_1, D_2, \dots, D_{21}$ , з інкрементами, що складаються з 10 випадкових документів з цієї колекції. Етап генерації наборів даних робочого процесу підтримується модулем Генератор Наборів Даних (Dataset Generator). Цей модуль приймає наступні входи [12]:

а) PFOLDER – ім'я папки, що містить документи TXT для формування наборів даних. Передбачається, що текстові файли в цій папці будуть названі використовуючи вищезазначену умову;

б) ORDER – порядок вибору документів для додавання в набори даних. Можливі чотири різних значення: "chrono" для хронологічного порядку, "rev-chrono" для зворотньо-хронологічного порядку, "bi-dir" для двонаправленого порядку, (iv) "random" для випадкового вибору документів;

в) INCRSIZE – кількість документів, які повинні бути включені в інкремент набору даних.

г) DFOLDER – ім'я папки для зберігання сформованих наборів даних.

Набори даних формуються відповідно до процедури OntoElect.

Етап здобуття термінів підтримується двома програмними засобами: UPM Term Extractor і NaCTeM TerMine.

UPM Term Extractor був розроблений у проекті Dr Inventor. Інструмент приймає текстовий документ на англійській мові (PDF або звичайний текст) і повертає набір здобутих термінів у вигляді CSV файлу. Кожен термін подано в окремому рядку зі своїм значенням c-value.

NaCTeM TerMine – це загальнодоступний сервіс, який використовується в пакетному режимі. Він використовує текстовий документ на англійській мові (ANSI) як файл для завантаження та повертає набір здобутих термінів у форматі CSV. Кожен термін подається в окремому рядку та супроводжується числовими значеннями його c-value і частоти. Інструмент дозволяє обирати текстовий парсер з двох доступних параметрів: tree tagger і genia tagger.

Етап вимірювання та аналізу насичення [12] підтримуються модулями THD, модулем Converter та модулем Stop Term Remover (видалення стоп-термінів).

## 2.4 Здобуття насиченого набору термінів

Здобуття термінології з текстів – це складний і трудомісткий процес, який потребує значної частини висококваліфікованих людських зусиль. Незважаючи на це, він все частіше використовується у багатьох важливих додатках, наприклад, для побудови онтологій. Отже, для ефективного розвитку онтологій із задовільним охопленням домену важливим є знання найменшої можливої репрезентативної колекції документів для домену [12].

Технології автоматичного здобуття термінів (АЗТ) важливі для практичного застосування. Але, вони ще далекі від того, щоб бути надійними. Запропоновані нові підходи до АЗТ все ще демонструють точність на рівні нижче 80 відсотків. Таким чином, їх важко використовувати в промисловості.

У більшості підходів до АЗТ обробка здійснюється у два послідовні етапи: лінгвістична обробка та статистична обробка. Лінгвістичні процесори, такі як тагери ЧМ (частин мови, POS) або чанкери фраз (phrase chunkers), відфільтровують стоп-слова і обмежують термінів-кандидатів до послідовностей n-gram (n-gram sequences): комбінацій іменники чи іменникові словосполучення, прикметник-іменник та іменник-прийменник-іменник. Потім застосовується статистична обробка для вимірювання рангу термінів-кандидатів. Ці міри є мірами «юнітхуду», які зосереджені на колокативній силі одиниць (слів), що складають певний термін, або міри «термхуду», що вказують на міцність асоціації термінів з концептами домену. Для «юнітхуду» використовуються показники, такі як: взаємна інформація, логарифмічна ймовірність, t-тест, а також поняття "модифіковуваність" та його варіанти. Метрики для «термхуду» базуються або на частоті (неконтрольовані підходи), або на використанні референтних корпусів (напівконтрольовані підходи).

Пізніше було запропоновано гібридні підходи, які поєднують вимірювання «юнітхуду» та «термхуду» у єдиному значенні.

Репрезентативним показником є  $C/nc$ -value. Підходи до АЗТ, які засновані на  $C/nc$ -value, отримали подальшу еволюцію в багатьох роботах. В усіх методах АЗТ лінгвістична обробка організована та реалізована однаково, за винятком того, що деякі з них також включають фільтрування стоп-слів. Стоп-слова (терміни) також можна відфільтрувати і на окремому етапі відсікання (cut off) після статистичної обробки.

У даному варіанті приділяється увага лише другому етапу – статистичній обробці. Статистична обробка іноді додатково розподіляється на дві послідовні підфази: оцінювання термінів-кандидатів та їх ранжування. Оцінювання термінів-кандидатів відображає ймовірність кандидату бути терміном. Відомі способи вимірювання оцінки можна поділити, як такі що спираються на вимірювання (див. таблицю 2.1): частот появ (включаючи словосполучення); оцінку попадань у контексти, а також такі, що використовують: референтні корпуси; моделювання тем.

Процедура відсікання приймає найбільш вагомих кандидатів на підставі оцінок, і тим самим відрізняє значущі терміни від незначних (чи не-термінів). Багато методів відсікання спираються на оцінки, отримані з одного алгоритму підрахунку оцінок, і встановлюють поріг відсікання в той чи інший спосіб. Деякі інші, які збирають оцінки з декількох алгоритмів зарахування, використовують (зважені) лінійні комбінації, голосування або (напів) контрольоване навчання.

Таблиця 2.1 – Способи вимірювання оцінки [12]

Метод [Джерело]	Незалеж- ність від домену (+/-)	Контро- льова- ність (U/SS)	Метрики	Значи- мість терміну	Відсі- кання (+/-)	Точність (GENIA; average)	Трива- лість викона- ння (%/c- value)
TTF [21]	+	U	Term (Total) Frequency	+	-	0.70; 0.35	0.34

Продовження таблиці 2.1

Метод [Джерело]	Незалеж- ність від домену (+/-)	Контро- льова- ність (U/SS)	Метрики	Значи- мість терміну	Відсі- кання (+/-)	Точність (GENIA; average)	Трива- лість викона- ння (%/c- value)
ATF [20]	+	U	Average Term Frequency	+	-	0.71; 0.33	0.37
						0.75; 0.32	0.35
TTF-IDF [22]	+	U	TTF+Inverse Document Frequency	+	-		
						0.82; 0.51	0.35
RIDF [23]	+	U	Residual IDF	-		0.71; 0.32	0.53
						0.80; 0.49	0.37
C-value [13]	+	U	c-value, nc-value	+	-	0.73; <b>0.53</b>	1.00
						0.77; <b>0.56</b>	1.00
Weirdness [12]	+/-	SS	Weirdness	-		0.77; 0.47	0.41
						0.82; 0.48	1.67
GlossEx [18]	+	SS	Lexical (Term) Cohesion, Domain Specificity	-			
						0.70; 0.41	0.42
TermEx [14]	+	SS	Domain Pertinence, Domain Consensus, Lexical Cohesion, Structural Relevance	-	+		
						0.87; 0.46	0.52
PU-ATR [16]	-	SS	nc-value, Domain Specificity	-	+	0.78; 0.57	809.21

**Коментарі:**

**Незалежність від домену:** "+" означає незалежний від домену метод;  
 "-" вказує на те, що цей метод є (або є визначеним його авторами, як)

специфічний для домену, або оцінюється лише в одному конкретному домені. Ми шукаємо метод незалежний від домену.

**Контрольованість:** "U" – неконтрольований; "SS" - напівконтрольований. Ми шукаємо неконтрольований метод.

**Значимість терміну:** "+" - метод повертає значення для кожного збереженого терміну, яке також може бути використане, як міра його значимості в порівнянні з іншими термінами. "-" - означає, що таке значення не повертається або сам метод робить відсікання. Ми прагнемо отримати значення, щоб пізніше розпочати відсікання.

**Відсікання:** "+" - метод сам робить відсікання і повертає лише значимі терміни; "-" - метод не робить відсікання. Ми шукаємо "-".

**Точність і тривалість виконання:** Ці значення базуються на порівнянні двох експериментів з крос-оцінювання. Пусті клітинки в таблиці означають, що в цьому конкретному експерименті не було даних для цього конкретного методу. Дослідження використовувало ATR4S – програмне забезпечення з відкритим кодом, написане в Scala. Автором було перевірено, на 5 різних наборів даних, включаючи GENIA, 13 різних методів, впроваджених у ATR4S. Дослідження використовувало JATE 2.0, вільне програмне забезпечення, написане на Java. Оцінювалися 9 різних методів, впроваджених у JATE, у двох різних наборах даних, включаючи GENIA. Отже, результати по GENIA є базовими для порівняння **Точності**. Для кожного референтного експерименту є два значення: точність на GENIA; середня точність. Обидва експериментували з методом *c-value*, який був у середньому найповільнішим. Таким чином, час виконання для методу *c-value* використовувався як базовий показник для нормалізації решти значень у стовпчику **Тривалість виконання**.

Проаналізувавши дані у таблиці 2.1, був зроблений висновок, що *c-value* є найбільш надійним методом, оскільки він послідовно дає хороші результати в плані точності термінів на двох різних сумішах (mixes) наборів даних. Також слід зазначити, що *c-value* є одним з найповільніших методів у

групі неконтрольованих і незалежних від домену, однак його ефективність порівнянна з найшвидшими. Також *c-value* перевершує залежні від домену методи, іноді значно – як у випадку з PU-ATR.

В даній роботі використовуватиметься метод *c-value*, як АЗТ метод для нашого експерименту з порівнювального оцінювання. [4]

Результуючі набори даних отриманих з колекції ISSEP не вміщували багато шуму (noise). Ми вибрали інкремент в 20 статей для створення наборів даних. Таким чином, на основі доступних текстів ми створили 21 інкрементально збільшені набори даних D1, D2, ..., D21. Після обробки датасетів за допомогою UPM Term Extractor ми отримали набір здобутих термінів у вигляді CSV файлів, в яких кожен термін подано в окремому рядку зі своїм значенням *c-value* (див. рис. 2.4).

terms	c-value
students	1751.294963
computer science	1469.513405
informatics	1292.120572
teachers	1013.460667
computational thinkin	600.4349966
computer	543.1440838
pupils	542.0798783
tasks	510.1373928
programming	505.8913852
education	457.6823869
computer science edu	433.8280852
computers	416.1473596
informatics education	413.5368349
information	392.8444897
information technolog	388.8531406
schools	388.6120157
problems	386.17925

Рисунок 2.4 – Приклад .csv файлу з набором термінів та їх *c-value*

У отриманому насиченому набору терміни записані як пара: строка-кандидат та оцінка ймовірності того, що термін є релевантним для домену: чим вище значення оцінки ймовірності тим більше ймовірність, що даний термін буде значимим для всіх наборів текстів і такі терміни визначаються як **значущі терміни**.



На основі цих даних було визначено алгоритм групування термінів описаної в пункті 2.5 цього розділу.

## 2.5 Групування термінів

Угрупування ознак об'єднує кілька ознак, які лексично відрізняються, але мають еквівалентну семантику. Відповідні випадки включають в себе: множинні і одиничні форми одного і того ж терміну, наприклад «тимчасове обмеження» і «тимчасові обмеження» є одними і тими ж термінами і повинні бути об'єднані. Термін, який втратив або не втратила дволітерні комбінації через особливості їх подання в документах PDF, наприклад «визначення» і «виз\_\_чення», також є одними і тими ж термінами. Значення значущості об'єднаних термінів підсумовуються.

Ієрархії включають в себе концептуальні функції, які є або частинами цілого, наприклад, вихідні дні є частиною тижні, або цілими для їх частин. Обидва типи цих ієрархічних відносин важливі, так як вони впливають на значимість ознак через успадкування властивостей. Дійсно, якщо об'єкт підпорядковується іншому об'єкту, то він успадковує деякі з його властивостей - так що значення цих елементів певною мірою формується цими успадкованими властивостями. Отже, батьківська ієрархія може очікувати, що вона буде винагороджена своїми дітьми через поширення їх значущих балів. OntoElect пропонує, що поширення балів додає одну п'яту балів дітей до балу їх батьків [14]. Приклад обчислення поширення оцінок від дочірніх термінів зображений на рисунку 2.5.

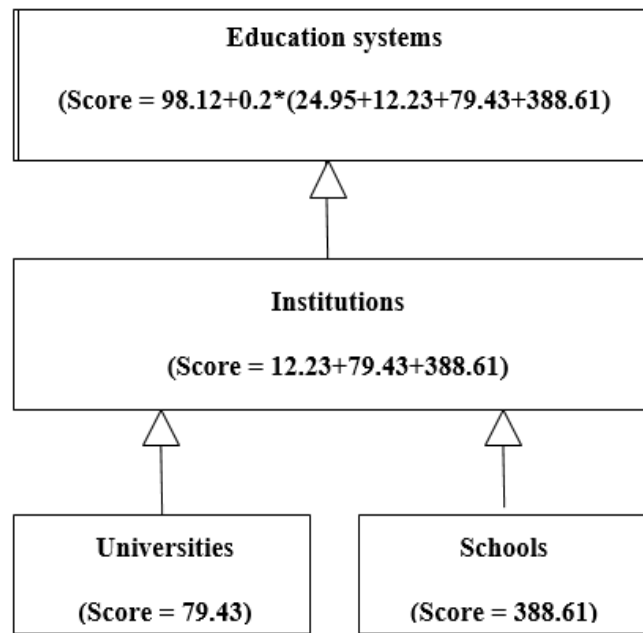


Рисунок 2.5 – Схема обчислення поширення оцінок від дочірніх

## 2.6 Висновок

В даному розділі ми вибрали колекцію наукових статей для термінологічного аналізу, описали послідовність розробки процедурного пайплайну для здобуття насиченого набору термінів із статей, представили результати після здобуття насиченого набору термінів з їх *c-value* і на основі здобутих даних виконали алгоритм групування термінів, щоб вже з остаточними результатами створити онтологію термінів.

### **3 РОЗРОБКА ОНТОЛОГІЇ “ШКІЛЬНА ІНФОРМАТИКА” ТА АНАЛІЗ ТЕРМІНОЛОГІЧНИХ РОЗБІЖНОСТЕЙ МІЖ СВІТОВИМИ І НАЦІОНАЛЬНИМИ ПРОГРАМАМИ З ІНФОРМАТИКИ**

#### **3.1 Методика і програмне середовище розробки онтології**

Онтологія – це загальноприйнята і загальнодоступна концептуалізація певної області знань (світу, середовища), яка містить базис для моделювання цієї області знань і визначає шляхи для взаємодії між агентами, які використовують знання з цієї області, і, нарешті, включає домовленості про представлення теоретичних основ даної області знань. В рамках комп’ютерних наук онтологія – це формальна назва і визначення типів, властивостей і взаємовідносин суб’єктів, що дійсно, або принципово, існують в обраному контексті (предметній області). Таким чином, вони є практичним застосуванням філософського поняття онтології, за допомогою таксономії. Сучасні онтології структурно подібні незалежно від мови, якою вони виражені. Більшість онтології описують осіб (зразки), класи (концепції), атрибути і відносини [15].

Задачі онтології:

- а) спільне використання людьми або програмними агентами загального;
- б) розуміння структури інформації;
- в) можливість повторного використання знань в предметній області;
- г) зробити допущення в предметній області явними;
- г) відділення знань в предметній області від оперативних знань;
- д) аналіз знань в предметній області.

У загальному вигляді структура онтології являє собою набір елементів чотирьох категорій: поняття, відносини, аксіоми, окремі екземпляри.

Поняття розглядаються як концептуалізації класу всіх представників якоїсь сутності або явища (наприклад, Людина, Їжа). Класи (або поняття) є загальними категоріями, які можуть бути впорядковані ієрархічно. Кожен клас описує групу індивідуальних сутностей, які об'єднані на підставі наявності загальних властивостей. Поняття можуть бути пов'язані різного роду відносинами (наприклад, Приналежність, Вага), які пов'язують воедино класи і описують їх. Найпоширенішим типом відносин, що використовується у всіх онтологіях, є ставлення категоризації, тобто віднесення до певної категорії. Цей тип відносин має ряд інших назв, що зустрічається в різних дослідженнях:

- а) таксономічне відношення;
- б) ставлення IS A;
- в) клас підклас;
- г) лінгвістика: гіпонім гіперонім;
- д) відношення; відношення a kind of.

Аксіоми задають умови відповідності категорій і відносин, вони виражають очевидні твердження, що зв'язують поняття і відносини. Під аксіомою можна розуміти твердження, що вводиться в онтологію у готовому вигляді, з якого можуть бути виведені інші твердження. Вони дозволяють висловити ту інформацію, яка не може бути відображена в онтології за допомогою побудови ієрархії понять і установки різних відносин між поняттями. Аксіоми дозволяють надалі здійснювати умовиводи в рамках онтології. Вони можуть постачати дослідників інформацією про правила, що дозволяють автоматично додавати інформацію. Аксіоми можуть також являти собою обмеження, що накладаються на будь-які відносини, які роблять можливим виведення умовиводів. Понятійні обмеження вказують на те, який тип понять може виражати дане відношення (наприклад, властивість Колір може виражатися тільки поняттями категорії Колір). Прикладом числових обмежень є твердження того, що для Людини кількість біологічних

батьків дорівнює 2. Кількість і ступінь деталізації аксіом зазвичай залежать від типу онтології, про що буде докладніше сказано далі.

Поряд із зазначеними елементами онтології в неї також входять так звані «екземпляри». У літературі вони можуть виступати також під назвами:

- а) конкретні екземпляри;
- б) інстанції;
- в) індивідуальні екземпляри.

Примірники – це окремі представники класу сутностей або явищ, тобто конкретні елементи будь-якої категорії.

Складові онтології підкоряються своєрідній ієрархії. На нижньому рівні цієї ієрархії знаходяться екземпляри, конкретні індивіди, вище йдуть поняття, тобто категорії. На рівень вище розташовуються відносини між цими поняттями, а узагальнюючої і сполучною є ступінь правил або аксіом.

Для створення та редагування онтологій ІС розроблено ряд спеціалізованих середовищ розробки, редакторів, парсерів та засобів об'єднання онтологій, найбільш ефективними з яких є: KAON, OilEd, OntoEdit, Ontosaurus, OpenCyc, Protégé.

Серед цих інструментів для побудови предметно-орієнтованої онтології виділимо редактор Protégé-OWL, як гнучке, незалежне від платформи середовище з своїми особливостями та перевагами, яке забезпечує наочний та зручний у використанні графічний інтерфейс користувачу, реалізує масштабованість, тобто модульне нарощування системи в рамках уніфікованої архітектури, дає змогу нарощувати архітектуру за допомогою додатково розроблених підпрограм – плагінів (plug-in). Також Protégé-OWL дає змогу описувати класи з використанням нових можливостей. Зокрема, мова OWL (Ontology Web Language) має великий набір операторів і базується на логічній моделі, яка дозволяє давати визначення поняттям так, як вони описані, тому складні комплексні поняття у визначеннях можуть бути створені з простіших. До того ж логічна модель дає змогу використовувати механізм міркувань (Reasoner), котрий у свою чергу дає змогу перевірити чи

твердження і визначення в онтології є взаємно несуперечливими, а також розпізнати відповідність визначень певним поняттям. Завдяки цьому механізму підтримується правильність ієрархії онтології. Редактор також підтримує більшість реляційних баз даних (Oracle, MySQL, Microsoft SQL Server, Microsoft Access). У редакторі Protégé-OWL забезпечено можливість вибору однієї з трьох розроблених на цей час версій мови OWL: OWL-Lite, OWL-DL, OWL-Full. Версії відрізняються мірою їх виразності. OWL-Lite є найпростішою з погляду синтаксису. Її найкраще використовувати під час побудови простої ієрархії класів та визначення семантичних обмежень. OWL-DL є значно виразнішою підмовою порівняно з OWL-Lite. Вона базується на описовій логіці (description logics) і орієнтована на тих користувачів, які хочуть максимальної прозорості без втрати повноти обчислень. OWL-DL охоплює всі мовні конструкції OWL з обмеженнями, на зразок розділення типів (клас не може бути властивістю, а властивість не може бути об'єктом або класом). Завдяки цьому OWL-DL забезпечує автоматичний логічний вивід (процес міркування). OWL-Full використовують в ситуаціях, де виразність засобів більш важлива, ніж потреба у обчислювальних можливостях мови. Вона забезпечує максимальну синтаксичну свободу стандарту RDF без обчислювальних гарантій. Водночас вона використовує всі переваги попередніх двох підмов.

Для побудови онтології, котра реалізована засобами Protégé-OWL, було зроблено наступне:

- а) визначено область та масштаб онтології;
- б) проаналізовано можливість використання існуючих онтологій;
- в) описано важливі терміни в галузі матеріалознавства;
- г) визначено класи та їх властивості;
- д) внесено екземпляри класів.

Базовим елементом розробленої онтології, як і в більшості інших, є класи, які можна розглядати як множини, що містять об'єкти. Класи в цій онтології організовані в ієрархію клас-підклас, тобто таксономію [16].

### 3.2 Трансформування груп термінів в онтологію у Protege

Після процесу здобуття насиченого набору термінів та його аналізу ми дійшли висновків, що всі терміни умовно можна розділити на 3 великі групи за областю дослідження.

Перша група термінів описує особливості системи освіти

Друга група розкриває питання структури дидактики інформатики як педагогічної науки, описує учасників процесу навчання та викладання та особливості їх взаємодії, а також описує зміст предмету інформатика(теми та розділи, що вивчаються).

Третя група, більша частина із всіх термінів, є загальними термінами, які містять назви частин світу, країн, абстрактні поняття та ін., тобто шум.

Тому для початку ми спробували розподілити терміни на вище вказані групи, а потім впорядкувати їх. Як вже було зазначено в попередніх пунктах цього розділу, кожен термін-кандидат має пару – значення його значимості в домені. Це значенні впливає на те, яким чином будуватиметься онтологія: багато термінів дублюються, чи мають синонімічне значення, тому в онтологію буде вноситися лише один з них, який в собі міститиме сумарне значення всіх схожих.

Для полегшення підготовчого процесу побудови онтології у Protégé OWL всі терміни-кандидати разом із їхніми значеннями доменної значимості були представлені у вигляді формальної ієрархії (дерева термінів) за допомогою програмного засобу Microsoft Office Excel. Ця програмна оболонка була обрана через простоту інтерфейсу і набір функцій для обчислення сумарних значень дочірніх термінів щодо батьківських. Формула для обчислення остаточного результату була описана в пункті 2.5 другого розділу. Приклад побудованої ієрархії зображений у Додатку Г.

Відповідно до отриманого результату були зроблені висновки стосовно напрямку розвитку гілок онтології.

Спочатку створимо клас *science* та його підклас *scientific discipline*. Ці класи мають в арсеналі екземпляри *pedagogy, methodology, chemistry, cryptography, biology, humanities*, а також *informatics*. Таке розташування у гілках ієрархії зумовлено тим, що інформатика в даному аспекті розглядається як наука, а не шкільний предмет.

Створимо клас *discipline*, який увібрав в себе значення всіх схожих термінів (*discipline, disciplines, various school subjects, school subject, school subjects, course, courses*).

Підкласом класу *discipline* зробимо термін *informatics in schools*, який увібрав в себе значення таких термінів, як *computing at school, informatics in education, informatics in schools, informatics lessons, ict, information technologies, information technology, cs courses, computing, computer science course, programming, programming course, programming courses*. Таким чином ми відокремили шкільний курс інформатики від інформатики як науки в цілому, тому що нашою метою є дослідження саме аспектів викладання та навчання шкільного предмету інформатика. Як уже було зазначено в розділі 1 в різних країнах світу вищезазначений предмет може носити різну назву. Термін *informatics* є узагальною назвою.

Наступною гілкою термінів є *world*, який має підклас *country*, який в свою чергу має такі екземпляри: *england, europe, austria, australia, bulgaria, czech republic, new zealand, poland, slovakia, sweden*. Можна зробити висновок, що найбільш популярними є питання представників конференції ISSEP з вищезазначених країн.

Наступним кроком створимо клас *ministry of education*, який став підкласом класу *educational systems*, та який був пов'язаний з екземплярами класу *country*, тому що (за логікою) в кожній країні є Міністерство освіти, яке впроваджує реформи в систему освіти та прописує її стандарти. З вищезазначеного можна зробити висновок, що повинен бути створений клас *educational standards*, який буде пов'язаний зв'язком з попереднім класом



*ministry of education*, а також матиме підклас *curriculum*, як результат діяльності міністерства.

Окремим класом створюємо клас *didactics of informatics*. Цей клас має такі підкласи: *contents, content, method, methods, methods of teaching computer science, principals, principle, principles*.

Ці терміни описують структуру дидактики як науки, яка розглядає інформатику в контексті освітнього процесу, мети та результатів.

Розглянемо структуру підкласів класу *didactics of informatics* детальніше.

Підклас *method* в контексті дидактики розглядається як спосіб ведення уроків, в нашому випадку саме уроків з інформатики. Екземплярами цього класу є *problem solving i project*. Це означає, що найбільш значимими методами, які розглядаються в контексті обраних статей виступають саме метод проблемного пошуку та метод проектів.

Підклас *principle* відповідає за опис принципів навчання інформатики – керівні ідеї, нормативні вимоги до організації і здійснення освітнього процесу.

Підклас *content* спрямований на огляд існуючого контенту про дидактичні засади викладання інформатики. Як нам відомо таку інформацію ми можемо знайти в навчальній програмі, або як її називають англійською *curriculum*. В попередньому пункті ми зазначили, що клас *curriculum* є підкласом *educational standards*. Тобто ми можемо вказати, що підклас *content* з підкласом *curriculum* є еквівалентними поняттями з точки зору дидактики.

У навчальній програмі з інформатики вчитель може знайти інформацію щодо мети курсу та результатів, які повинні отримати всі учасники освітнього процесу. Тому в класі *curriculum* ми створили 2 підкласи *education goals* та *results*, які описують вищезазначені елементи. До результатів навчальної діяльності належать *knowledge, abilities, ability, skills*.

Підклас *skills* є тотожним до класу *basic skills*, який має у екземплярах набір термінів, що описують вміння і навички, що ними повинні оволодіти учні в процесі навчання: *computational thinking skills, design skills, ict skills, information skills, programming skills, basic skills, 21st century skills, thinking skills, children\_s cognitive skills, algorithmic thinking, computational thinking, logic, thinking, way of thinking*.

Також в навчальній програмі описана структура курсу, технологія та послідовність вивчення різних тем, а також вказані данні про те, які саме види робіт повинні використовувати вчителі на уроках.

Ми створили підкласи *structure, section, strategy, technology*. Навчальний процес з інформатики розглядається в аспекті навчальної або викладацької діяльності, в залежності від обраного об'єкту діяльності (вчитель (викладач, той хто навчає) та учень (студент, той хто навчається)).

Для опису вищезазначеної залежності створимо клас *process* який є сумарним значенням діяльності та містить 3 підкласи: *teaching, study, learning*. Наступний кроком створимо класи, що описують суб'єкти процесу навчання – людей (клас *people*), а саме *teacher* та *student*. Ці класи є підкласами класу *people* та є узагальнюючими поняттями.

### **3.3 Побудова і аналіз покриття онтології тематикою національних програм з шкільної інформатики**

Дослідивши результат був зроблений висновок, щодо тематики вивчення інформатики в національних шкільних програмах. В обраному домені знань розглядаються такі розділи:

- а) комп'ютер та його складові;
- б) алгоритми та моделювання;
- в) бази даних та структури даних;
- г) програмне забезпечення комп'ютера;

- г) табличний процесор та таблиці;
- д) інформація та інформаційні технології;
- е) комп'ютерні мережі.

Ці розділи в певній мірі відповідають розділам, що запропоновані в навчальній програмі із інформатики в Україні та світі. Як зазначалося в Розділі першому, є багато особливостей в підході до викладання інформатики в школах різних країн, а також відрізняється і зміст освіти. В деяких країнах вищезазначені розділи розглядаються повністю і мають окрім них додаткові розділи (наприклад: Текстовий процесор та особливості роботи з текстовими документами; Веб-технології та ін). Є країни в яких більшу увагу приділяють програмуванню, починаючи вводити теми з цього розділу вже в початкових класах. Проаналізувавши перелік розділів, що розглядаються в обраному наборі статей та порівнявши значення їх *c-value*, можемо зробити висновок, що найактивніше обговорюються питання вивчення розділу *Компютер та його складові* (2599,69). На другому місці зі значенням (938,07) *Алгоритми та моделювання*, на третьому – (812,94) *Бази даних та структури даних*. На 4 місці розташувався розділ *Інформація та інформаційні технології* (803,94), на п'ятому місці *Табличний процесор та таблиці*(183,19) та на останньому місці за рейтингом розташовується розділ *Комп'ютерні мережі*(97,86).

### **3.4 Висновок**

В цьому розділі ми детально описали алгоритм перетворення у онтологію вже готового насиченого набору термінів-кандидатів та їх позначкою значимості для обраного домену. Проаналізували групи термінів та побудували залежності між класами онтології. На основі отриманих знань зробили висновок стосовно повноти покриття онтологією тематики шкільних програм з інформатики в світі.

## ВИСНОВКИ

Питання вдосконалення освітнього процесу та покращення якості освіти є актуальним завжди. Дипломна робота була спрямована на дослідження якості освіти в предметній області «шкільна інформатика». Дослідження проводилося на основі термінологічного аналізу статей міжнародної конференції з питань інформатики для початкової та середньої ланки школи (ISSEP). Так як дослідити всю літературу в даній предметній області не є можливим, було вирішено відібрати матеріали зі статей ISSEP та провести аналіз термінологічних розбіжностей між світовими та українськими підходами до викладання на основі поступових етапів збору і обробки термінології з даного домену.

В ході роботи було досліджено питання сучасного стану формування навчальних програм з інформатики в середній школі України та світу. , розглянуті нормативно методичні бази для шкіл України і світу, проаналізовані тематичне наповнення програм курсу, розбіжності у методичних підходах до викладання інформатики в українських школах та у школах різних країн Європи, були описані відомі алгоритми здобуття термінів з текстів, розроблений процедурний пай-лайн для здобуття насиченого набору термінів, описана процедура підготовки колекції для термінологічного аналізу, алгоритм розробки каталогу, здобуття і вимірювання насичення, а також обґрунтована насиченість та описаний процес групування термінів.

В останньому розділі роботи був описаний алгоритм розробки онтології “шкільна інформатика” та проведений аналіз термінологічних розбіжностей між світовими і національними програмами з інформатики. Було розглянуто методику і програмне середовище розробки онтології, був описаний процес трансформування груп термінів в онтологію у Protege та

безпосередньої побудови онтології. Після побудови онтології був проведений аналіз її покриття тематикою національних програм з шкільної інформатики.

На основі проведеного експерименту був зроблений висновок, що найактивніше на конференції світового рівня ISSEP обговорюються питання вивчення в курсі шкільної інформатики таких розділів, як Комп'ютер та його складові. Алгоритми та моделювання, Базы даних та структури даних. Інформація та інформаційні технології, Табличний процесор та таблиці, Комп'ютерні мережі. Питання щодо методики проведення уроків з інформатики розглядаються в неповній мірі. Такий висновок був зроблений після дослідження значень релевантності термінів, що описують дидактичні засади викладання обраного шкільного курсу.

На основі вищезазначеного висновку були розроблені рекомендації, щодо удосконалення дидактичних засад у викладанні інформатики в українських школах. З ними можна ознайомитися в Додатку Д.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Інформатика як наука і як навчальний предмет у загальноосвітній школі. Студопедия. URL: [https://studopedia.su/6\\_14703\\_Informatika-yak-nauka-i-yak-navchalniy-predmet-u-zagalnoosvitniy-shkoli.html](https://studopedia.su/6_14703_Informatika-yak-nauka-i-yak-navchalniy-predmet-u-zagalnoosvitniy-shkoli.html) (дата звернення 27.09.2019).
2. Інформатика. Wikipedia. URL: <https://uk.wikipedia.org/wiki/Інформатика#Історія> (дата звернення 29.09.2019).
3. Аналіз методичної системи навчання. Методична система навчання інформатики в середній загальноосвітній школі. Студопедия. URL: [https://studopedia.su/6\\_14704\\_analiz-metodichnoi-sistemi-navchannya.html](https://studopedia.su/6_14704_analiz-metodichnoi-sistemi-navchannya.html) (дата звернення 27.09.2019).
4. Інформатика у школі, Блог про дистанційне та змішане навчання інформатики. Технології та системи дистанційного навчання. Moodle (дата оновлення 27.12.2014). URL: <http://dystosvita.blogspot.com/2014/12/blog-post.html> (дата звернення 29.09.2019).
5. Інформатика. Навчальна програма вибірково-обов'язкового предмету для учнів 10 – 11 класів загальноосвітніх навчальних закладів (рівень стандарту). Сайт Міністерства освіти і науки України. URL: <https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-10-11-klas/2018-2019/informatika-standart-10-11.docx> (дата звернення 02.10.2019).
6. Кравцова А. Ю., Развитие содержательных линий базового курса информатики в общеобразовательной школе на основе анализа опыта Великобритании: дис. ... канд. пед. наук, 13.00.02 / Кравцова Алла Юріївна; Москва, 1998 177 с. Библиотека диссертаций и авторефератов России dslib.net. URL: <http://www.dslib.net/teoria-vospitania/razvitie-soderzhatelnyh-linij-bazovogo-kursa-informatiki-v-obweobrazovatelnoj.html> (дата звернення 30.09.2019).

7. Kemp P. Computing in the national curriculum. [Електронний ресурс]. A guide for secondary teachers/ Dorling M., Humphreys S., Hunt S., Jackson C., Berry M., Jackson A. 32 p., Великобританія: British Library Cataloguing in Publication Data, 2014. Режим доступу: [https://www.computingatschool.org.uk/data/uploads/cas\\_secondary.pdf](https://www.computingatschool.org.uk/data/uploads/cas_secondary.pdf).
8. Coding at school: a parent's guide to England's new computing curriculum. The Guardian. URL:<https://www.theguardian.com/technology/2014/sep/04/coding-school-computing-children-programming> (дата звернення 01.10.2019).
9. Освітні програми. Сайт Міністерства освіти і науки України. URL: <https://mon.gov.ua/ua/osvita/zagalna-serednya-osvita/navchalni-programi> (дата звернення 02.10.2019).
10. Програма курсу інформатика 5 – 9 класи загальноосвітніх навчальних закладів. Сайт Міністерства освіти і науки України. URL: <https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-5-9-klas/onovlennya-12-2017/programa-informatika-5-9-traven-2015.pdf> (дата звернення 02.10.2019).
11. Ермолаєв В. А., Коса В. До Теорії Термінологічного Насичення для Вивчення Онтології з Текстів.
12. Kosa, V., Chaves-Fraga, D., Naumenko, D., Yuschenko, E., Moiseenko, S., Dobrovolskyi, H., Vasileyko, A., Badenes-Olmedo, C., Ermolayev, V., Corcho, O., and Birukou, A.: The Influence of the Order of Adding Documents to Datasets on Terminological Saturation. Technical Report TS-RTDC-TR-2018-2-v2, 21.11.2018, Dept. of Computer Science, Zaporizhzhia National University, Ukraine, 72 p.
13. PDF. Вікіпедія. URL: <https://uk.wikipedia.org/wiki/PDF> (дата звернення 05.10.2019).
14. Ermolayev V.A., OntoElecting requirements for domain ontologies. The case of time domain. EMISA Int J of Conceptual Modeling 13(Sp. Issue), 86–109 (2018).

15. Галушко М. О., Бібліотека наукових публікацій з використанням RDF-сховищ. Сайт кафедри системного проектування КПІ ім. І. Сікорського. URL: [http://cad.kpi.ua/attachments/093\\_2016d\\_Halushko.pdf](http://cad.kpi.ua/attachments/093_2016d_Halushko.pdf)(дата звернення 10.10.2019).

16. Гончаренко С. У. Педагогічні дослідження. Методологічні поради молодим науковцям. Київ – Вінниця : ДОВ «Вінниця», 2008. 278 с.



## ДОДАТОК А

### Результати оцінки насиченості колекції ISSEP

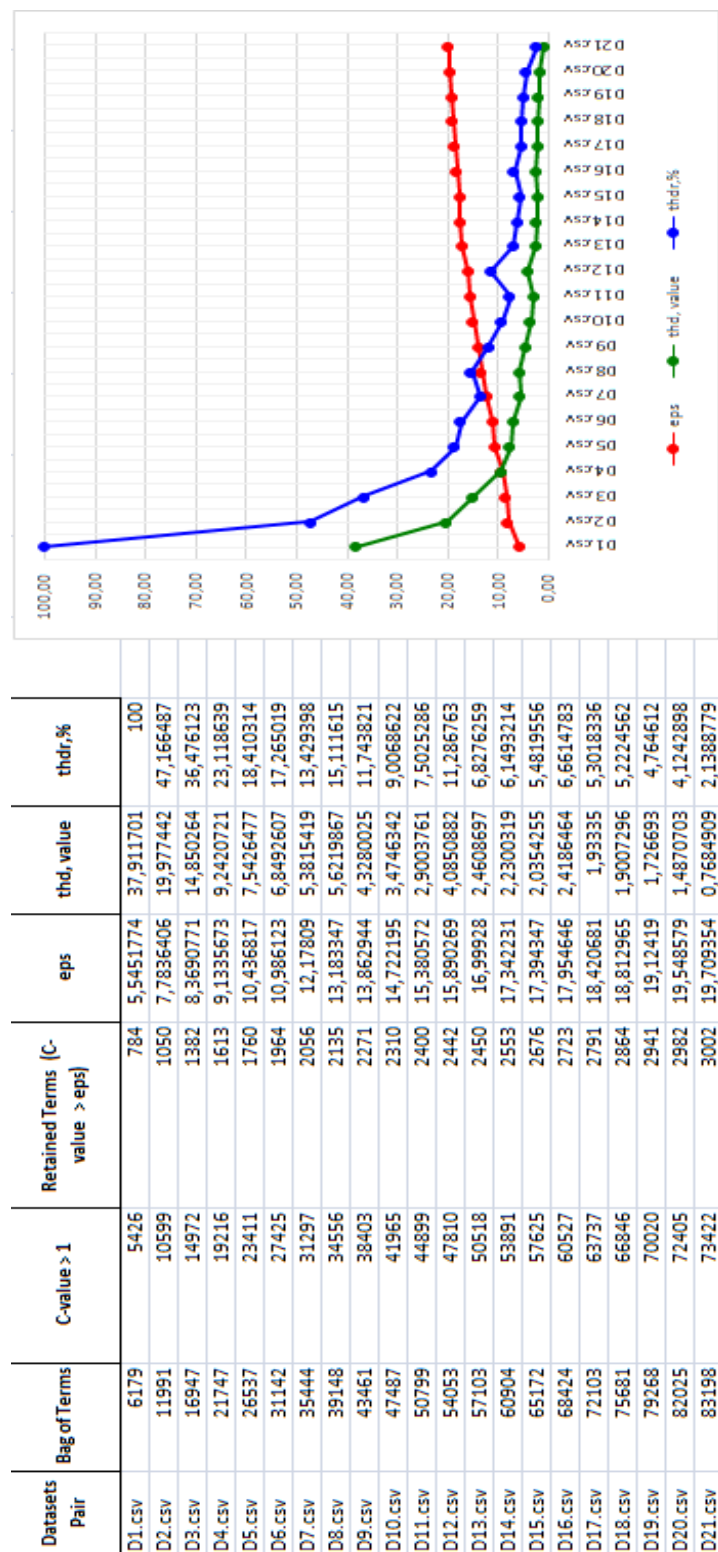


Рисунок А.1

# ДОДАТОК Б

## Розміщення файлів статей по директоріях на Google Drive

The screenshot shows the Google Drive interface. At the top, there is a search bar with the text "Поиск на Диске". Below it, the "Мой диск" (My Drive) section is visible. On the left side, there are navigation options: "Создать" (Create), "Мой диск" (My Drive), "Доступные мне" (Shared with me), "Недавние" (Recent), "Помеченные" (Starred), "Корзина" (Trash), "Резервные копии" (Backups), and "Хранилище" (Storage). The "Хранилище" section shows "Использовано 9,5 ГБ из 15 ГБ" and a link to "КУПИТЬ БОЛЬШЕ МЕСТА". The main area displays a list of folders under "Мой диск". Each folder is named "ISSEP\_2005" through "ISSEP\_2017" and is owned by "я" (me). The last modified date for all folders is "27 сент. 2019 г. я".

Название	Владелец	Последнее изменение
ISSEP_2005	я	27 сент. 2019 г. я
ISSEP_2006	я	27 сент. 2019 г. я
ISSEP_2008	я	27 сент. 2019 г. я
ISSEP_2010	я	27 сент. 2019 г. я
ISSEP_2011	я	27 сент. 2019 г. я
ISSEP_2013	я	27 сент. 2019 г. я
ISSEP_2014	я	27 сент. 2019 г. я
ISSEP_2015	я	27 сент. 2019 г. я
ISSEP_2016	я	27 сент. 2019 г. я
ISSEP_2017	я	27 сент. 2019 г. я

Рисунок Б.1

# ДОДАТОК В

## Каталог статей

A	B	C	D	E	F	G	H	I	J	K
Year	Journal	Pages	DOI	DOI link	Paper Title	Abstract	Key phrases	Citations (GS)	paper file name (KM DropBox)	Link
1	2005	5-19	10.1007/978-3-540-31958-0_2	<a href="https://link.springer.com/doi/10.1007/978-3-540-31958-0_2">https://link.springer.com/doi/10.1007/978-3-540-31958-0_2</a>	Incorporation of 20 Years of Computers	Effects of The way in	Prospective Teaching	6	ISSEP_2005-2005-1(1)-(5)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
2	2005	20-31	10.1007/978-3-540-31958-0_3	<a href="https://link.springer.com/doi/10.1007/978-3-540-31958-0_3">https://link.springer.com/doi/10.1007/978-3-540-31958-0_3</a>	Informatics Education	The contribution	Service	25	ISSEP_2005-2005-1(1)-(5)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
3	2005	32-36	10.1007/978-3-540-31958-0_4	<a href="https://link.springer.com/doi/10.1007/978-3-540-31958-0_4">https://link.springer.com/doi/10.1007/978-3-540-31958-0_4</a>	The Transition from Informatics and ICT in Teaching Information	We investigate During the last	School Level	1	ISSEP_2005-2005-1(1)-(5)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
4	2005	37-45	10.1007/978-3-540-31958-0_5	<a href="https://link.springer.com/doi/10.1007/978-3-540-31958-0_5">https://link.springer.com/doi/10.1007/978-3-540-31958-0_5</a>	Informatics and ICT in Teaching Information	The reform of	Computer	6	ISSEP_2005-2005-1(1)-(5)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
5	2005	46-52	10.1007/978-3-540-31958-0_6	<a href="https://link.springer.com/doi/10.1007/978-3-540-31958-0_6">https://link.springer.com/doi/10.1007/978-3-540-31958-0_6</a>	Educational Standards	In the last two	Subject Area	10	ISSEP_2005-2005-1(1)-(5)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
6	2005	65-69	10.1007/978-3-540-31958-0_7	<a href="https://link.springer.com/doi/10.1007/978-3-540-31958-0_7">https://link.springer.com/doi/10.1007/978-3-540-31958-0_7</a>	Russian Educational	Educational	General	25	ISSEP_2005-2005-1(1)-(5)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
7	2005	70-74	10.1007/978-3-540-31958-0_8	<a href="https://link.springer.com/doi/10.1007/978-3-540-31958-0_8">https://link.springer.com/doi/10.1007/978-3-540-31958-0_8</a>	The Present-Day Study of Informatics	The article	Practical Skill	10	ISSEP_2005-2005-1(1)-(5)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
8	2005	75-83	10.1007/978-3-540-31958-0_9	<a href="https://link.springer.com/doi/10.1007/978-3-540-31958-0_9">https://link.springer.com/doi/10.1007/978-3-540-31958-0_9</a>	Why Teach	Qualified search		10	ISSEP_2005-2005-1(1)-(5)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
9	2005	84-91	10.1007/978-3-540-31958-0_10	<a href="https://link.springer.com/doi/10.1007/978-3-540-31958-0_10">https://link.springer.com/doi/10.1007/978-3-540-31958-0_10</a>	Teaching People to Preparatory	Introductory		4	ISSEP_2005-2005-1(1)-(5)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
10	2005	92-92	10.1007/978-3-540-31958-0_11	<a href="https://link.springer.com/doi/10.1007/978-3-540-31958-0_11">https://link.springer.com/doi/10.1007/978-3-540-31958-0_11</a>	In the recent Spreadsheets	The importance	Programming	0	ISSEP_2005-2005-1(1)-(5)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
11	2005	93-103	10.1007/978-3-540-31958-0_12	<a href="https://link.springer.com/doi/10.1007/978-3-540-31958-0_12">https://link.springer.com/doi/10.1007/978-3-540-31958-0_12</a>	Preparatory	In the recent	Secondary	17	ISSEP_2005-2005-1(1)-(5)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
12	2005	104-115	10.1007/978-3-540-31958-0_13	<a href="https://link.springer.com/doi/10.1007/978-3-540-31958-0_13">https://link.springer.com/doi/10.1007/978-3-540-31958-0_13</a>	A Pragmatic Approach	Spreadsheets	Data	25	ISSEP_2005-2005-1(1)-(5)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
13	2005	116-129	10.1007/978-3-540-31958-0_14	<a href="https://link.springer.com/doi/10.1007/978-3-540-31958-0_14">https://link.springer.com/doi/10.1007/978-3-540-31958-0_14</a>	A Strategy to Introduce Informatic Models in Evolving Boxes as	Users of During the last	Unify	6	ISSEP_2005-2005-1(1)-(5)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
14	2005	130-144	10.1007/978-3-540-31958-0_15	<a href="https://link.springer.com/doi/10.1007/978-3-540-31958-0_15">https://link.springer.com/doi/10.1007/978-3-540-31958-0_15</a>	Informatic Models in Evolving Boxes as	Users of During the last	Logic Program	11	ISSEP_2005-2005-1(1)-(5)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
15	2005	145-155	10.1007/978-3-540-31958-0_16	<a href="https://link.springer.com/doi/10.1007/978-3-540-31958-0_16">https://link.springer.com/doi/10.1007/978-3-540-31958-0_16</a>	The Role of ICT and Informatics Versus	Secondary	General	8	ISSEP_2005-2005-1(1)-(5)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
16	2005	156-165	10.1007/978-3-540-31958-0_17	<a href="https://link.springer.com/doi/10.1007/978-3-540-31958-0_17">https://link.springer.com/doi/10.1007/978-3-540-31958-0_17</a>	The Role of ICT and Informatics Versus	Secondary	Information	8	ISSEP_2005-2005-1(1)-(5)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
17	2005	166-177	10.1007/978-3-540-31958-0_18	<a href="https://link.springer.com/doi/10.1007/978-3-540-31958-0_18">https://link.springer.com/doi/10.1007/978-3-540-31958-0_18</a>	Standard Software as The Future Is Mobile – More than three	Traditionally	Programming	10	ISSEP_2005-2005-1(1)-(5)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
18	2005	178-188	10.1007/978-3-540-31958-0_19	<a href="https://link.springer.com/doi/10.1007/978-3-540-31958-0_19">https://link.springer.com/doi/10.1007/978-3-540-31958-0_19</a>	Standard Software as The Future Is Mobile – More than three	Traditionally	Information	16	ISSEP_2005-2005-1(1)-(5)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
19	2005	189-197	10.1007/978-3-540-31958-0_20	<a href="https://link.springer.com/doi/10.1007/978-3-540-31958-0_20">https://link.springer.com/doi/10.1007/978-3-540-31958-0_20</a>	Standard Software as The Future Is Mobile – More than three	Traditionally	Programming	2	ISSEP_2005-2005-1(1)-(5)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
20	2005	198-201	10.1007/978-3-540-31958-0_21	<a href="https://link.springer.com/doi/10.1007/978-3-540-31958-0_21">https://link.springer.com/doi/10.1007/978-3-540-31958-0_21</a>	Standard Software as The Future Is Mobile – More than three	Traditionally	Information	0	ISSEP_2005-2005-1(1)-(5)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
21	2006	43800	10.1007/11915355_1	<a href="https://link.springer.com/doi/10.1007/11915355_1">https://link.springer.com/doi/10.1007/11915355_1</a>	Evolution of the Cultural Discovering Informatics	The history of Theoretical	Distance Boolean	5	ISSEP_2006-2006-1(1)-(1)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
22	2006	13-24	10.1007/11915355_2	<a href="https://link.springer.com/doi/10.1007/11915355_2">https://link.springer.com/doi/10.1007/11915355_2</a>	Evolution of the Cultural Discovering Informatics	The history of Theoretical	Distance Boolean	0	ISSEP_2006-2006-1(1)-(1)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
23	2006	25-37	10.1007/11915355_3	<a href="https://link.springer.com/doi/10.1007/11915355_3">https://link.springer.com/doi/10.1007/11915355_3</a>	Contributing to General Bridging the Gap	What is For the last two	Computer	10	ISSEP_2006-2006-1(1)-(1)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
24	2006	38-47	10.1007/11915355_4	<a href="https://link.springer.com/doi/10.1007/11915355_4">https://link.springer.com/doi/10.1007/11915355_4</a>	Bridging the Gap	What is For the last two	Computer	52	ISSEP_2006-2006-1(1)-(1)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
25	2006	48-58	10.1007/11915355_5	<a href="https://link.springer.com/doi/10.1007/11915355_5">https://link.springer.com/doi/10.1007/11915355_5</a>	Programming Versus All over Europe	For the last two	Teaching	13	ISSEP_2006-2006-1(1)-(1)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
26	2006	59-70	10.1007/11915355_6	<a href="https://link.springer.com/doi/10.1007/11915355_6">https://link.springer.com/doi/10.1007/11915355_6</a>	Programming Versus All over Europe	For the last two	Teaching	20	ISSEP_2006-2006-1(1)-(1)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
27	2006	71-82	10.1007/11915355_7	<a href="https://link.springer.com/doi/10.1007/11915355_7">https://link.springer.com/doi/10.1007/11915355_7</a>	Databases as a Tool of Handling the Diversity	All over Europe	Database	4	ISSEP_2006-2006-1(1)-(1)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
28	2006	83-93	10.1007/11915355_8	<a href="https://link.springer.com/doi/10.1007/11915355_8">https://link.springer.com/doi/10.1007/11915355_8</a>	Databases as a Tool of Handling the Diversity	All over Europe	Database	8	ISSEP_2006-2006-1(1)-(1)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
29	2006	83-93	10.1007/11915355_8	<a href="https://link.springer.com/doi/10.1007/11915355_8">https://link.springer.com/doi/10.1007/11915355_8</a>	Computer Science in High Schools in	High Schools in	Information	16	ISSEP_2006-2006-1(1)-(1)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>
30	2006	83-93	10.1007/11915355_8	<a href="https://link.springer.com/doi/10.1007/11915355_8">https://link.springer.com/doi/10.1007/11915355_8</a>	Computer Science in High Schools in	High Schools in	Information	16	ISSEP_2006-2006-1(1)-(1)	<a href="https://drive.google.com/file/d/1G11111111111111111111111111111111/view">https://drive.google.com/file/d/1G11111111111111111111111111111111/view</a>

Рисунок В.1

# ДОДАТОК Г

## Скриншот розробленої онтології

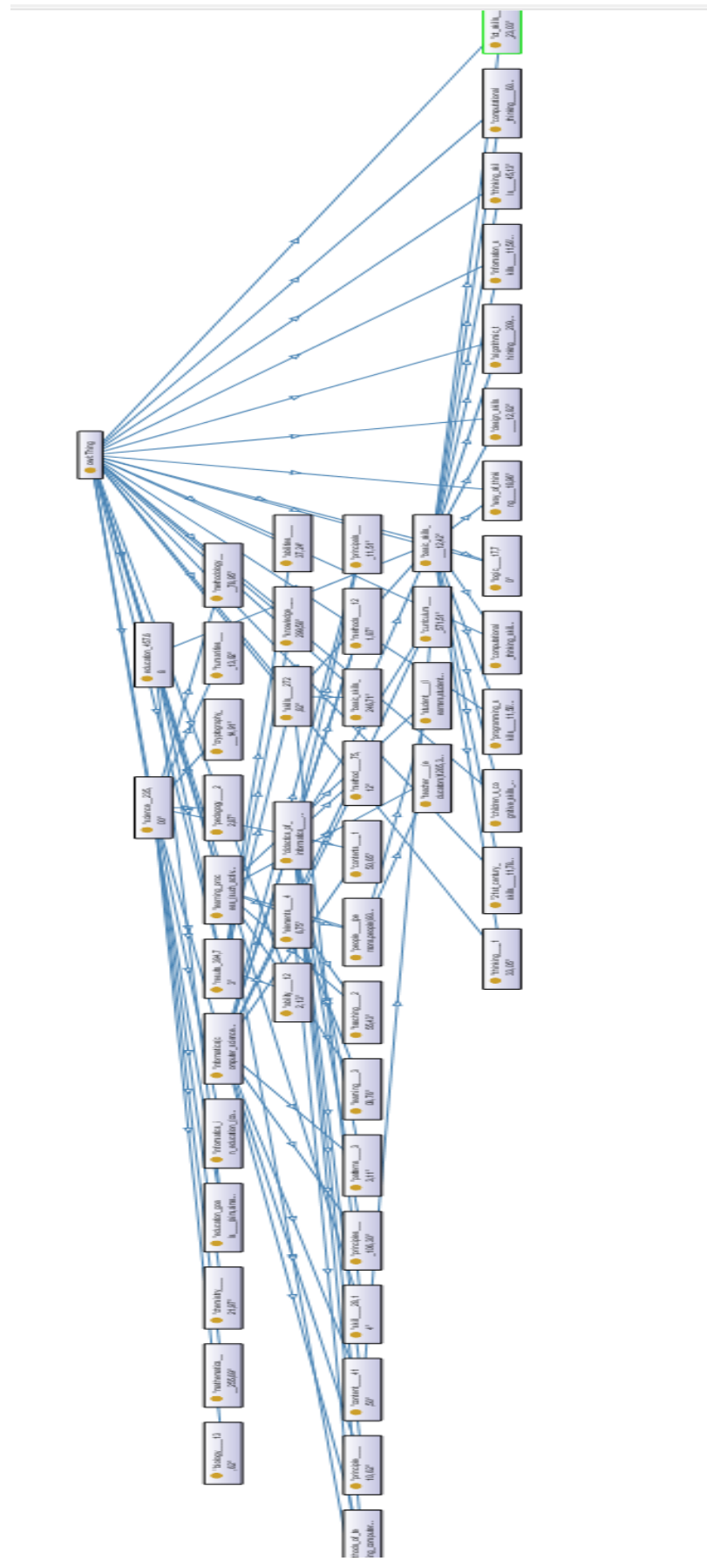


Рисунок Г.1

## ДОДАТОК Д

### Рекомендації

1. Використовувати нестандартні форми організації роботи на уроці. Як показує практика вчителів Європи, нестандартні форми проведення уроків сприяють розвитку нестандартного мислення в учнів.

2. Вводити дидактичні ігри в навчальний процес. Бажано щоб ігри були рухливими та осягали декількох органів чуття одразу, тобто були спрямовані на декілька відчуттів одночасно(дотик, слух, зір і т.д.).

3. Звертати більше уваги на індивідуальну та домашню роботу учнів, бо як показує практика зарубіжних вчителів, якщо учень систематично виконує завдання з інформатики не лише на уроках, а й вдома, тим краще він розумітиметься в цій галузі.

4. Корисним також для вчителя інформатики з України стане дослідження різниці підходу до навчання Інформатиці в світі, тому що досвід показує, що більшість вчителів в українських школах працюють “по-старинці”, тобто використовують вже готове тематичне планування, не враховуючи особливостей учнів різних класів власної школи, проводять уроки без інтерактивних методів і т.д.

5. Не боятися використовувати зарубіжну літературу з предмету, як для самостійного вивчення вчителем, так і для опрацювання учнями. Це потрібно для того, щоб вдосконалювати знання іноземних мов, краще розумітися в термінах з інформатики та для розширення кругозору.

6. Активно використовувати ІКТ, сервіси інтернету та освітні платформи. Готуючись до уроку продумувати варіанти застосування вищезазначених елементів під час його різних етапів. Це якісно підвищить рівень знань учнів, а також буде мотивувати їх відвідувати уроки з бажанням.