

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра мультимедійних систем факультету інформатики



Реалізація бази знань за допомогою системи PROTEGE

Текстова частина до курсової роботи
за спеціальністю «Інженерія програмного забезпечення»- 121

Керівник курсової роботи

к.ф.-м.н., доцент

Жежерун О.П.

(підпис)

“ ____ ” _____ 2020 р.

Виконав студент ІІЗ-3:

Коваленко А.Ю.

“ ____ ” _____ 2020 р.

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем факультету інформатики

ЗАТВЕРДЖУЮ
Зав. кафедри мультимедійних систем,
доцент, к.ф.-м.н.
_____ Жежерун О.П.
(підпис)
„____” _____ 2020 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студенту Коваленко Андрію Юрійовичу факультету інформатики 3 курсу

ТЕМА Реалізація бази знань за допомогою системи PROTEGE

Вихідні дані:

Зміст ТЧ до курсової роботи:

Індивідуальне завдання

РОЗДІЛ 1: Теоретична частина

РОЗДІЛ 2: Опис алгоритму, який пропонується в роботі

РОЗДІЛ 3: Реалізація алгоритму

Список використаної літератури

Програмні додатки (тексти програм)

Висновки

Список використаної літератури

Додатки

Дата видачі „____” _____ 2020 р. Керівник _____
(підпис)

Завдання отримав _____
(підпис)

Календарний план виконання роботи:

Тема: Реалізація бази знань за допомогою системи PROTEGE

№	Назва етапу	Термін виконання	Примітка
1.	Отримання теми курсової	09.10.2019	
2.	Пошук тематичної наукової літератури	20.10.2019	
3.	Ознайомлення з науковою літературою	11.12.2019	
4.	Розбір предметної області	18.12.2019	
5.	Перші спроби реалізації	01.01.2020	
7.	Визначення структури програми	05.02.2020	
8.	Написання 1 частини курсової роботи	25.04.2020	
9.	Написання 2 частини курсової роботи	30.04.2020	
10.	Написання 3 частини курсової роботи	2.05.2020	
11.	Написання висновків курсової роботи	7.05.2020	
12.	Перегляд змісту роботи з керівником	7.05.2020	
13.	Внесення змін до роботи	9.05.2020	
14.	Завантаження курсової роботи	11.05.2020	

Студент Коваленко А. Ю.

Керівник Жежерун О. П.

“ ”

Зміст

Календарний план виконання роботи:	3
Зміст	4
Анотація	5
Вступ	6
1. Теоретична частина	8
1.1 Онтологія та бази знань	8
1.2 База знань традиційного типу її структура та класифікація, мета створення.	9
1.3 Сфера та приклади застосування бази знань	10
1.4 Визначення поняття експертної системи	12
2. Опис алгоритму, який пропонується в роботі	15
2.1 Загальний опис програми, що потребується	15
2.2 Варіанти алгоритмів вирішення задачі	15
2.3 Огляд можливих систем для використання у програмі.....	16
2.4 Алгоритм для роботи програми.....	18
3. Реалізація алгоритму	19
3.1 Структура програми та її класів	19
3.2 Загальний опис роботи програми та її компонентів	19
3.3 Опис використаних технологій.....	24
Висновок.....	26
Список використаної літератури	27

Анотація

У курсовій роботі наведені теоретичні відомості про бази знань традиційного типу та бази знань побудовані на основі онтології. Проведена порівняльна характеристика та визначення переваг і недоліків між цими типами .

Представлено опис сфер діяльності та приклади їх застосування. Проведено вивчення предметної області. Наведено опис написання алгоритму, що потребує програма на основі онтології Protégé та опис структур ,використаних API та технологій для його реалізація.

В роботі використовується система Protégé, OWL parser, Java 1.8

Ключові слова : База знань, онтологія, експертна система, штучний інтелект, OWL, редактор онтологій, Protégé, геометрія.

Вступ

Актуальність теми. У сучасному світі людство намагається автоматизувати більшість процесів своєї життєдіяльності. З дитинства людину привчають вирішувати всі задачі своїм розумом, самим знаходити розв'язок, шляхом аналізу теорем, правил, задач, а також намагається зрозуміти те що передає інша людина усним мовленням. Кожен сприймає та обробляє інформацію з різною швидкістю, швидкість ще інколи залежить від способу надходження, хочеться завжди прискорити цей процес, або навіть щоб хтось інший виконав ці етапи, тоді можна одразу перейти до рішення. Коли стає зрозумілим, що витрачений час буде дуже великим, або вектор вирішення питання не є видимим, починаються пошуки варіантів, хто виконає поставлену задачу. Виконавцем може бути інша людина, яка зіткнеться з тими самими етапами, або автоматизована система, програма що одразу дасть відповідь. Користувач може вбити свою задачу та побачити готовий розв'язок, це надасть йому бачення, яким чином було вирішене поставлене питання. Для того щоб програма зрозуміла задачу потрібно розбити її на фрагменти, зрозуміти вхідні дані та питання.

Мета дослідження – реалізувати базу знань на основі онтології Protege, написати простий аналізатор тексту, спробувати ототожнити всі задачі з геометрії 7 класу під один формат, розробити початкові класи та вбити в них початкові формули для подальшої розробки програми, що буде вирішувати завдання формату ЗНО.

Досягнення мети передбачає виконання низки завдань:

- визначення поняття бази знань та сфери її застосування;

- вивчення предметної області, а саме геометрії 7 клас;
- опис алгоритму який пропонується в роботі;
- опис реалізованого алгоритму с технічними подробицями;
- опис структури, використаних API та технологій, необхідних для реалізації бази знань, заснованої на онтологіях Protégé.

Об'єкт дослідження – база знань, створена на основі онтологій системи Protégé.

Предметом дослідження є способи та технології вирішування задач з геометрії 7 класу на основі онтологій Protégé, переваги його застосування.

Структура курсової роботи така: вступ, три розділи ,список використаної літератури та програмні додатки.

У першому розділі наведено визначення поняття бази знань традиційних типів , класифікацій , мета створення, місце та сфера застосування, надано визначення поняття онтологія

В другому розділі розповідається про загальні потреби від програми та шляхи вирішення поставлених задач, описані різні алгоритми та підходи, огляд системи Protégé.

У третьому розділі наведено детальний опис структури і функціонування алгоритму розробленого на основі онтології Protégé.

1.Теоретична частина

1.1 Онтологія та бази знань

Онтологія може бути структурою бази знань, тому доречно буде почати, з визначення онтології, та що вона собою являє.

Онтологія - це ієрархічно структурування безлічі термінів, що описують предметну область, яка може бути використана як вихідна структура для бази знань.[4]

Компонентами онтології можуть бути:

1. Клас
2. Відносини
3. Функції
4. Аксиоми
5. Екземпляри

Онтології у програмуванні та не тільки, потрібні для опису термінів предметної області та їх зв'язків. Крім розробників штучного інтелекту, онтології набувають популярності серед експертів в різних предметних областях. У всесвітній мережі WWW онтології вже стали звичайним явищем наприклад: категоризації веб-сайтів (як на веб-сайті Google.com), категоризація товарів, що продаються і їх характеристик (як на веб-сайті Rozetka.com). У багатьох областях зараз розробляються стандартні онтології для спрощення знаходження та доповнення вже існуючих термінів і зв'язків експертами, що полегшує роботу користувача з великим об'ємом даних у кожній окремій предметній області.

Наприклад, в області медицини створені великі стандартні, структуровані словники, такі як SNOMED і семантична мережа Системи Уніфікованого Медичної Мови (Unified Medical Language System, UMLS). Також з'являються

великі спільноцільові онтології. Наприклад, Програма ООН з розвитку (the United Nations Development Program) і компанія Dun & Bradstreet об'єднали зусилля для розробки онтології UNSPSC, яка надає термінологію товарів і послуг (unspsc.org).[4]

1.2 База знань традиційного типу її структура та класифікація, мета створення.

База знань - це база даних розроблена для оперування знаннями (методами) та містить структуровану інформацію для використання кібернетичним пристроєм або людиною з конкретною метою.

Структуру баз знань можна представити у вигляді таких основних під баз: база даних і база правил, база фактів, база цілей, база мета знань, база процедур.

При створенні бази знань акцентують увагу на виді інформації, яку вона має містити. Це модель знань, які змінюються сильно рідше ніж дані, або факти які швидко змінюються при виконанні певної задачі. Знання зберігає інформацію про самого себе, з них можуть впливати наступні факти, правила, гіпотези. Можна назвати це метазнанням - знанням системи про себе, зазвичай окремий блок бази знань, в який система не буде заходити, якщо не використовуються правила які в середині. Для пошуку рішення пробігаються по всім блокам і знаходиться найбільш ефективна стратегія.

Час обробки поданої інформації сильно ранжується від розміру систематизованої бази знання. Система повинна весь час зменшувати область шуканої відповіді та цілеспрямовано виконувати наступні кроки, аналізуючи вже отриманні результати або просто виконувати так званий «брут ворс» - повний перебір всіх знань.

База даних -це сукупність структурованих взаємопов'язаних динамічна обновлюваних даних певної предметної області.

Предметною областю може бути частина реального світу, що підлягає вивченню з метою її автоматизації або реорганізація існуючої автоматизованої системи.

В ній зберігається інформація про розв'язування на об'єкті завданнях і дані, що відносяться до зазначеної предметної області.

База правил визначає правила і обмеження які відображають зв'язки між елементами даних, що зберігаються в базі даних на основі моделей представлених знань про предметну область та методи активізації цих знань.

Класифікація баз знань :

В залежності від рівня складності системи розрізняють такі бази знань:

- Всесвітнього масштабу
- Національні
- Галузеві
- Організаційні
- Експертних систем
- Фахівців

Метою створення бази знань є подальше її використання в експертних системах для вирішення та визначення методу вирішення експертних завдань за допомогою правил логічного виведення.

1.3 Сфера та приклади застосування бази знань

Застосовуються бази даних для збереження даних про організацію, створення експресивних систем, створення системи штучного інтелекту.

- У 1994 році в рамках проекту PiES WorkBench був розроблений продукційно-фреймовий мова представлення знань PILOT / 2. Особливістю даної мови є уявлення декларативного опису предметної області за допомогою фреймів, а процедурного - за допомогою продукцій. Приклад коду:
Child is_a Age restr_by (>0 && <12);
- Психодіагностика системою «Орієнтир» має набір тестів, в залежності від відповідей формує рекомендації щодо вибору особи до конкретного колективу виду діяльності.
- Wolfram alpha – це обчислювальний калькулятор на основі бази знань, працює шляхом отримання результату, роблячи аналіз та обчислення використовуючи свою базу знань Wolfram, замість пошуку в Інтернеті та повернення посилань
- I&W (indications and warnings) - це система основана на базі знань, розроблена американськими вченими та аналітиками. Основною розробленої системи є проведення розвідувальних заходів, цілю яких є виявити та попередити про загрозу розвідку США щодо ворожих або терористичних намірів, які можуть бути спрямовані проти США, та мати на меті завдати шкоди військовим, політичним чи економічним інтересам, або громадянам США за кордоном. Система повинна передбачувати наступні дії чи наміри противника; неминучість бойових дій; повстання; ядерна / неядерна атака на Сполучені Штати, його заморські сили або країни союзників та / або коаліції; ворожі реакції на розвідувальні дії США; напади терористів; та інші подібні події.
- MYCIN - це експертна система першого покоління, яка почала використовувати штучний інтелект для виявлення бактерій, які є причиною виникнення важких інфекцій, такі як бактеремія та менінгіт, система

спроможна рекомендувати певні антибіотики, дозування яких визначається за масою тіла пацієнта. Ще однією задачею системи MYCIN може бути те, що вона використовується для діагностики захворювань згортання крові. MYCIN був розроблений протягом п'яти-шести років на початку 1970-х років в Стенфордському університеті. Вона була написана на мові програмування Лісп, як докторська дисертація Едварда Шортліффа під керівництвом Брюса Г. Бьюкенана, Стенлі Н. Коена та інших.

1.4 Визначення поняття експертної системи

Моє завдання полягає в якомусь сенсі розробити експертну систему. Тому слід дати визначення, що собою являє експертна система.

Експертна система – це спосіб використання алгоритму в інтелектуальній комп'ютерній програмі, яка має певну базу знань, основу експертами, з предметної області, та здатна використовуючи знання зробити певні логічні відповіді. Експертна система може вирішити специфічне питання швидше та точніше, замість одного експерта чи групи.

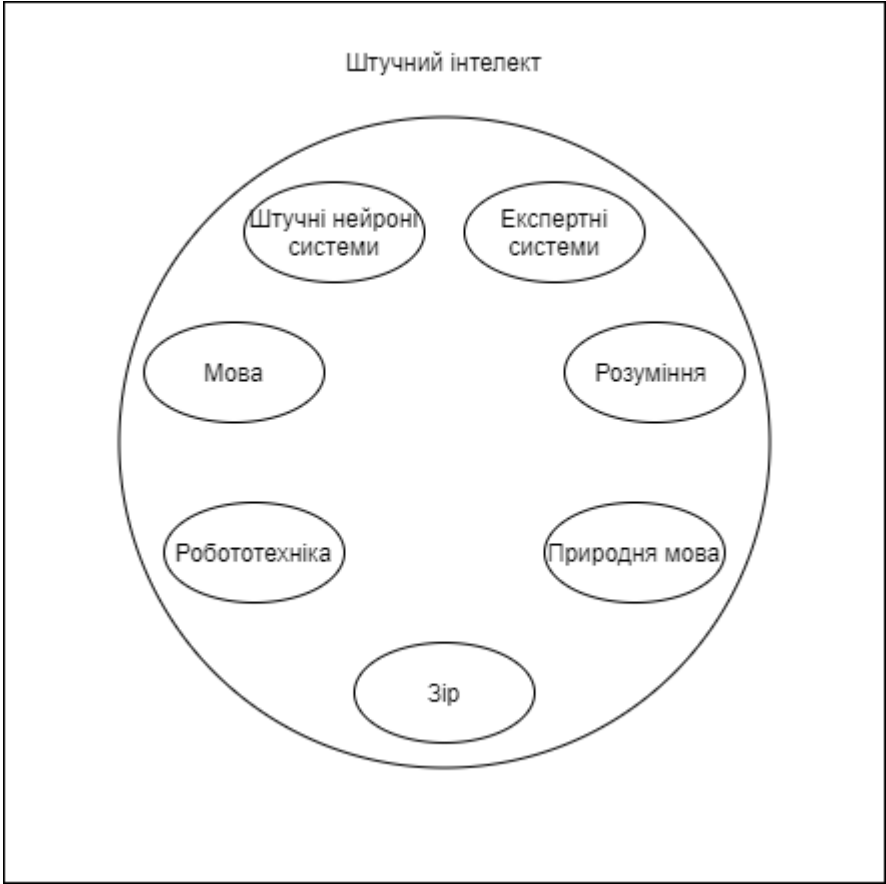
Експертна система є дуже гарним використанням та реалізації технології штучного інтелекту, існує також сумісні системи, які використовують методи експертних систем у додаток до інших. Системи які являють собою використання штучного інтелекту називають інтелектуальними системами або автоматизованими.

Як вже було згадано раніше експертній системі для роботи потрібні певні початкові нароби, базу знань та/або базу даних. Наприклад використання системи для розпізнавання образів, так само як це робить звичайне людське око та мозок, та з автоматизованими системами для знаходження відповіді, аналізу неявних закономірностей та подальшим занесенням нових правил, знань до інтелектуальної бази даних.

Однією з прикладів використання системи розпізнавання образів, є програма системи безпеки аеропортів. Задача системи полягає у розпізнаванні обличчя потенційних підозрюваних. Програма отримує дані через камери, аналізує їх порівнюючи з даними що міститься в системі, та приймає рішення чи треба сповістити органам правопорядку аеропорту. Система має право тільки аналізувати данні що в нею занесені. Інший же приклад коли програма може доповнювати власну базу знань.

Система створення відкриттів є гарним прикладом області штучного інтелекту. Система являє собою комп'ютерну програму, що може знаходити нові відкриття в різних галузях, залежності від знань що занесені до неї. Прикладом такої може слугувати Automated Mathematician – автоматизований математик. Програма змогла відкрити декілька нових теорем, та підтвердила декілька вже існуючих щодо простих чисел. BACON 3 програма яка виконувала математичні обрахунки, щодо руху об'єктів в космічному просторі підтвердила 3 версію закону планетарного руху Кеплера. Далі на діаграмі (див. рис. 1) представлені ще різні сфери використання штучного інтелекту.

Рисунок 1 - Діаграма штучного інтелекту:



[11]

2. Опис алгоритму, який пропонується в роботі

2.1 Загальний опис програми, що потребується

Основним завданням моєї курсової роботи є розробка бази знань для вирішення задач з геометрії 7 класу на основі системи онтологій Protégé. З самого початку було заплановано виконати цю задачу разом з іншими студентами 3 курсу ІПЗ, які також пишуть курсову роботу на цю ж тему, однак обставини карантину, самоізоляція, та недостатній зв'язок з членами команди та керівником, примусили відмовитись від цієї ідеї. Кожен член команди розробляв свою частину, свій розділ з геометрії. Моя частина це вирішення задач з трикутником.

Наприклад (див. рис. 2):

Рисунок 2

200.° Периметр рівнобедреного трикутника дорівнює 28 см, а бічна сторона — 10 см. Знайдіть основу трикутника.

Програмі на вхід повинна подаватися задача з підручника «Геометрія 7 клас А.Г. Мерзляк, В.Б. Полонський, М.С. Якір» з розділу Трикутники. Програма повинна проаналізувати умови, питання, та надати відповідь до задачі.

2.2 Варіанти алгоритмів вирішення задачі

Перший варіант вирішення та написання програми полягав у тому, що існують певні загальні схеми подання задач. Існувала теорія, що існує близько десяти варіантів постановки задачі, та постановки питання. Таким чином для першого етапу зчитування інформації не потрібно було дуже сильного лексичного аналізатора. Ця теорія була швидко спростована глобальним вивченням всього підручника. На превеликий жаль автори робили кожну другу задачу з зовсім

іншим типом надання інформації щодо дано, та що знайти. Спільний тип був лише у задач що біли для класної роботи та домашньої. Наприклад (див. рис. 3):

Рисунок 3

201.° Знайдіть сторони рівнобедреного трикутника, периметр якого дорівнює 32 см, а основа на 5 см більша за бічну сторону.

202.° Знайдіть сторони рівнобедреного трикутника, периметр якого дорівнює 54 см, а основа в 4 рази менша від бічної сторони.

За задумкою таких однотипних за формулюванням умови задач повинно бути десь 80/750. Однак ця думка була швидко спростована, спробую написання хоча б однієї загальної бази. Порядок підмету, присудку числових значень сильно різниться від задачі до задачі.

Одночасно з проробкою подання задач була спроба написання різних видів конструкторів для задання даних трикутнику. Після аналізу підручнику, з'ясувалося, що конструктори повинні бути під кожну окрему задачу, що вже ставало десь біля 300. Що робило код не надто стислим та гнучким. Тому прийшлося відмовитися від даного варіанту виконання задачі.

Наступний варіант написання був таким. В класах кожен елемент може стати доступний через геттер або сеттер. Для аналізу тексту використовується аналізатор. Були наступні варіанти побудови на основі таких систем: OpenCyc, KIF, Openlingvia. На жаль жоден з цих варіантів не було знайдено opensource. Єдиний аналітор тексту що вдалося запустити безкоштовно Pullenti.

2.3 Огляд можливих систем для використання у програмі

OpenCyc - це проект штучного інтелекту, основна задача якого зібрати глобальну базу онтологій та знань, про те як працює весь світ. Перевагу відає неявним

знанням, що може відрізнятися від загальновідомих фактів. Таким чином міркування стає схожим на людське, і їй легше занести нові правила в базу.

Це була одна з найцікавіших систем для побудови програми, однак безкоштовно її отримати не вийшло навіть в учбових цілях.

Pullenti проводить повний лінгвістичний аналіз будь-якого тексту. Дозволяє відокремити та виділити іменовані сутності, морфологію та семантику. Окремо виділяє організації країни, дати, персон, гроші. Може це робити на декількох мовах у тому числі і на українській. Являє собою аналізатором всебічного типу.
[8]

Почавши тестування Pullenti все було добре, аналізатор визначав закінчення, розбирав текст, знаходив головну частину та другорядну. Але він повністю ігнорував числа. Це дало розуміння, що потрібен спеціалізований аналізатор, який буде покривати певну предметну область, а саме геометрії. Таке Pullenti не міг робити.

У результаті не було знайдено жодного аналізатора, що задовольняє умовам для вирішення задачі. Зроблено спробу написати свій лайтовий лексичний аналізатор.

Було використано Protégé (версії 5.5.0) для побудови онтології з геометрії. Protégé - це відкритий для всіх безкоштовний редактор онтології та система управління базою знань. Protégé має графічний інтерфейс, який легко дозволяє додавати нові правила та залежності онтології. Програма написана на Java і широко використовує Swing для створення інтерфейсу користувача.

2.4 Алгоритм для роботи програми

Перший етап роботи програми визначаються в зчитуванні тексту задачі, яка була введена користувачем через графічний інтерфейс користувача, задача повинна бути з геометрії 7 класу про трикутники.

Далі програма повинна використати лексичний аналізатор для розбору тексту і перетворення його в код. На цьому етапі знадобиться онтологія розроблена в базі знань Protégé. Далі всі обчислення виконуються в програмі, та виводиться відповідь користувачу. При цьому можна записувати, які конкретно формули були використані.

3. Реалізація алгоритму

3.1 Структура програми та її класів

Shape – інтерфейс на основі якого будуть реалізовані всі фігури, у моєму прикладі реалізовано лиш один клас.

MyTriangle – клас реалізуючий інтерфейс фігури, та має основні дані трикутника, використовує клас відрізка.

Dot – клас точки.

LineSegment – клас відрізка використовує клас точки.

SyntaxAnalyzer – клас синтаксичного аналізу, що є основним «мозком» програми отримує задачу розпарсену по словам и генерує відповідь.

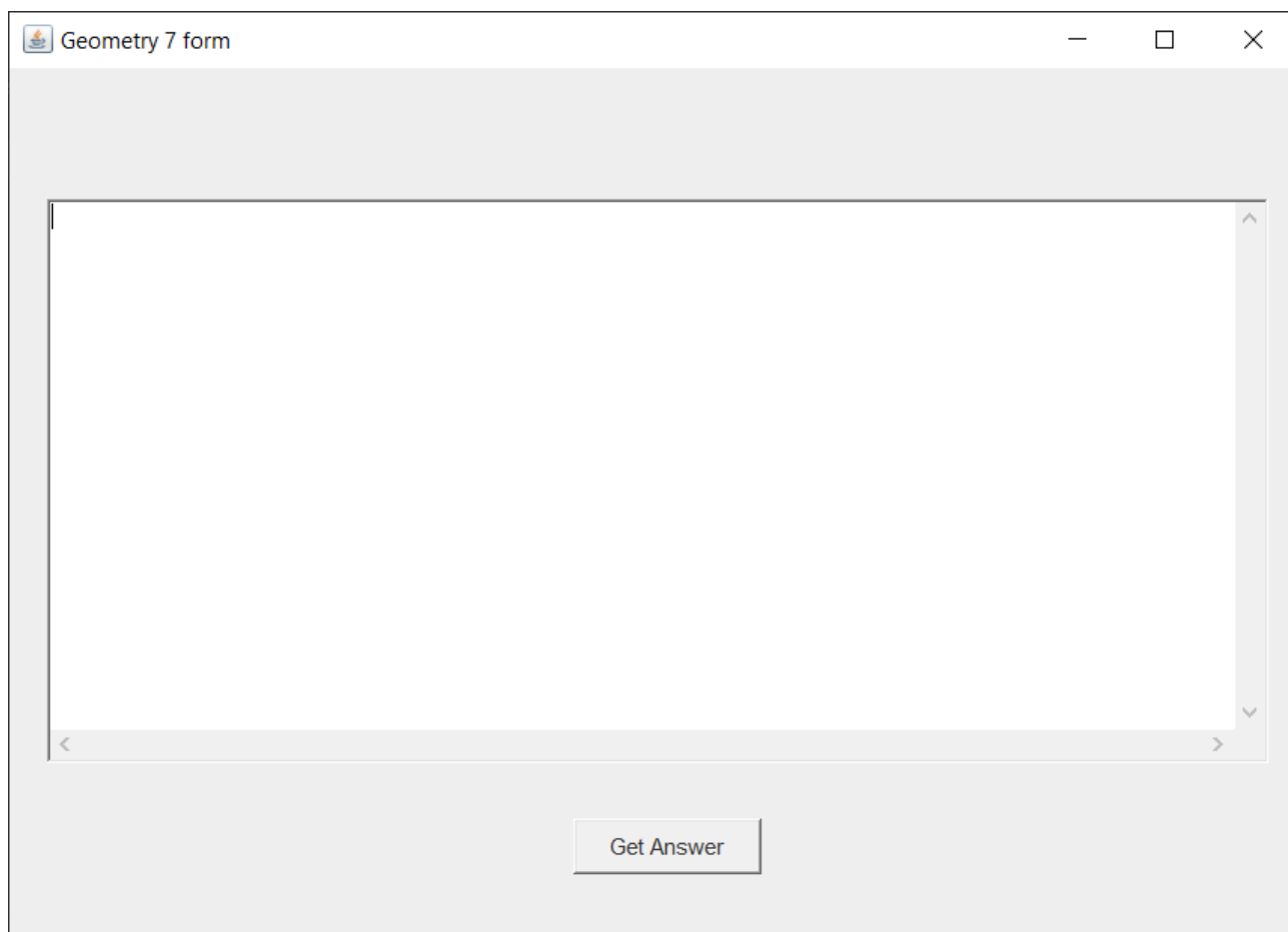
GUI – клас графічного інтерфейсу.

3.2 Загальний опис роботи програми та її компонентів

Основним завданням моєї курсової було написати програму, яка використовує базу знань Protégé на основі онтологій,

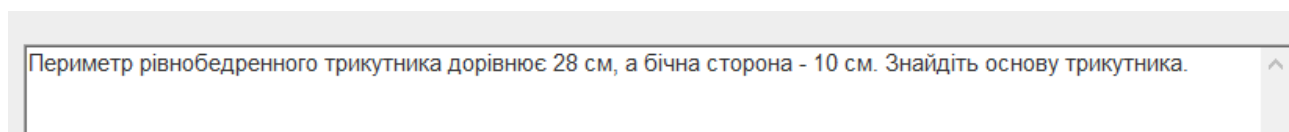
Програма приймає задачу 7 класу з геометрії та надає її рішення. Користувачу достатньо ввести в текстове поле саме завдання. Саме поле вводу не має ніяких специфічних особливостей та готове прийняти будь-який текст. Має такий вигляд. За графічний інтерфейс користувача (див. рис. 4) відповідає клас GUI, в ньому ж знаходиться метод мейн що запускає всю програму, в якому створюється об'єкт класу GUI.

Рисунок 4 - Інтерфейс програми:



Заносимо в поле введення нашу задачу (див. рис. 5).

Рисунок 5



Натискаємо кнопку "Get Answer".

Після цього виконуються зчитування тексту задачі, парситься (див. рис. 6) по словах кожне слово заноситься в стек та передається в наступний клас.

Рисунок 6 - Код лістенера на кнопці:

```
public void actionPerformed(ActionEvent e){
    String text=area.getText();
    // String words[]=text.split("\\s");
    Stack<String> exercise = new Stack<>();
    String[] split = text.split( regex: " ");
    for (String word:split){
        exercise.push(word);
    }
    SyntaxAnalyzer sa = new SyntaxAnalyzer();
    sa.analyze(exercise);
}
```

Далі оскільки не було знайдено нормального аналізатора мови, треба було писати самому. А написання аналізатора є самою по собі дуже важкою задачею. Було наведено рішення лише на декількох. І моя програма повинна працювати лише з трикутниками. Тому в самому початку в мене по любому повинен бути об'єкт фігури (див. рис. 7).

Рисунок 7 - Код створення нового об'єкту:

```
Shape sh=new Shape()
```

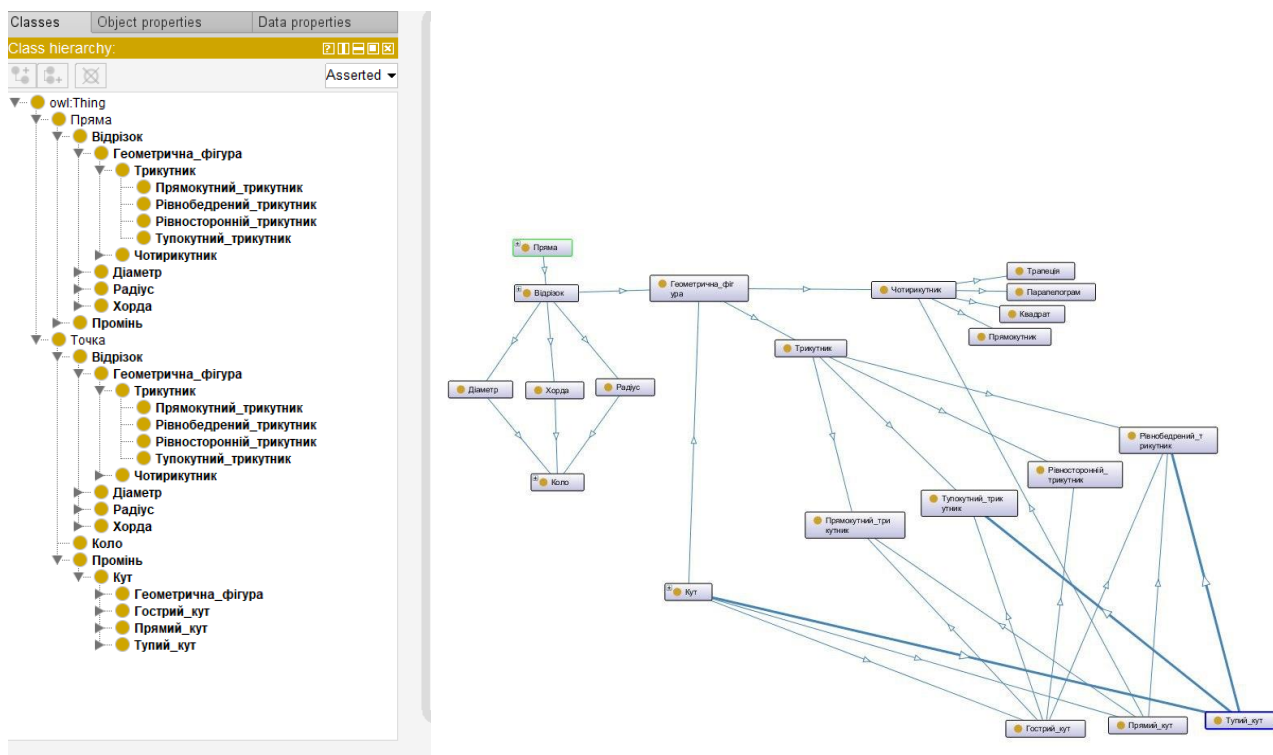
Shape є інтерфейсом, тому можливо слід одразу створювати об'єкт класу MyTriangle, але що це за об'єкт ми дізнаємось тільки в середині аналізатора, і створити його там глобальним не можемо. На цьому етапі ми розуміємо що це трикутник (див. рис. 8).

Рисунок 8 - Перетворення об'єкта на об'єкт класу трикутника:

```
while (!shape.equals("трикутника")) {
    shape = exercise.get(g++);
}
sh = new MyTriangle();
```

Також було створено систему онтологій, що допомагала в аналізі (див. рис. 9)

Рисунок 9 - Онтологія та її схема:



Всі нові дано зчитувались та передавались у клас трикутника (див. рис. 10)

Рисунок 10 - Код додавання даних до класу трикутника:

```

if(type.equals("рівнобедренного")){
    ((MyTriangle) sh).setTriangleType(Type.Rivnobedrenyi);
}
if(type.equals("прямокутного")){
    ((MyTriangle) sh).setTriangleType(Type.Pryamokutnyi);
} else if ( ((MyTriangle) sh).getTriangleType() == null){
    ((MyTriangle) sh).setTriangleType(Type.Riznostoronni);
}

```

Коли ми доходимо до запитання, клас намагається дістати поле що потребує відповідь, в залежності чи вже це поле відоме, виконується його обрахунок.

Тут можна дописувати усі варіанти та формули за якими можна вирішувати дану задачу, інколи важко вказати комп'ютеру яку з них використовувати, а від цього може залежати точність обрахунку. У лістингу коду нижче показану варіант

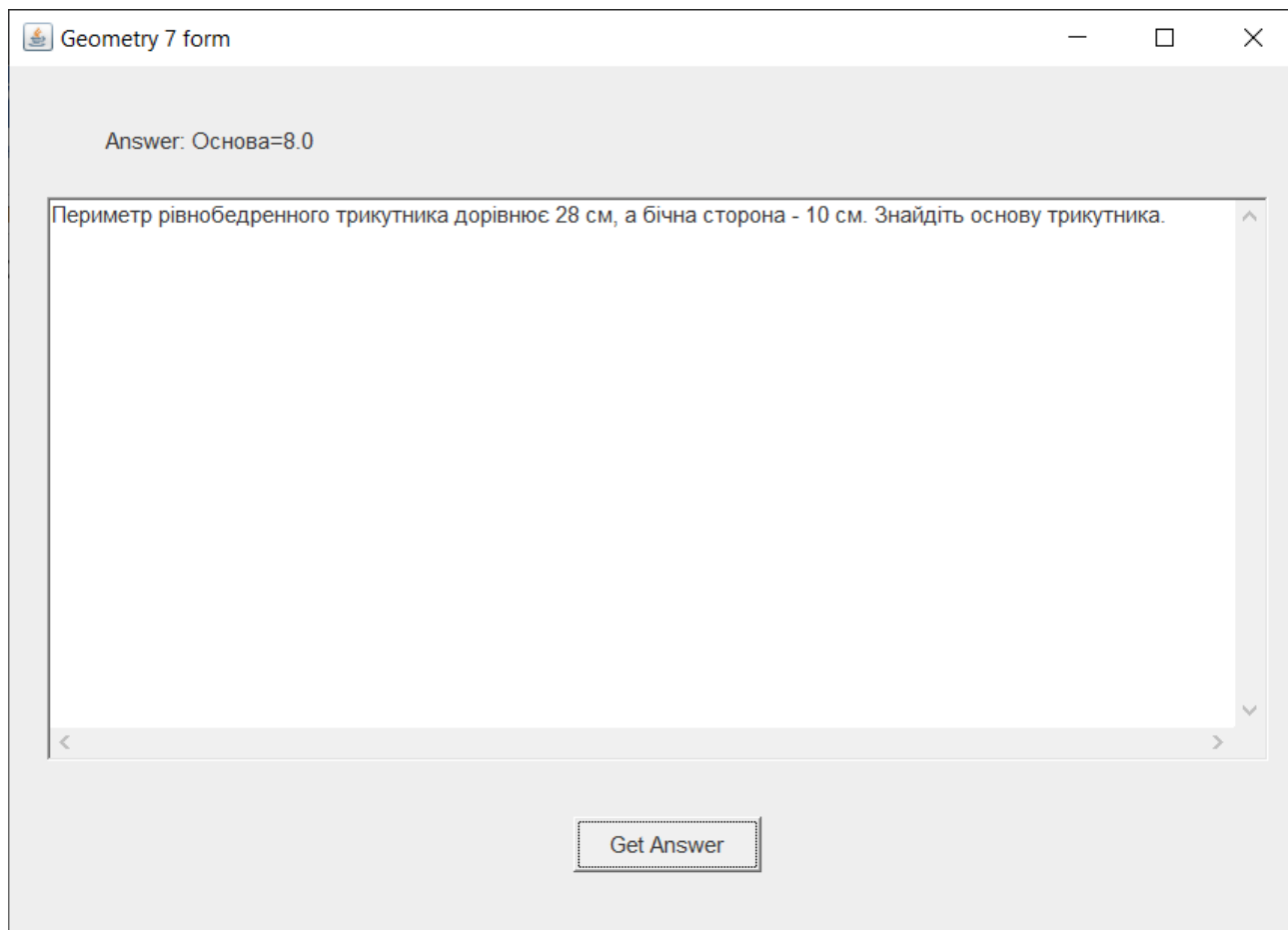
знаходження сторони трикутника у випадки коли невідомо її та відомо периметр (див. рис. 11).

Рисунок 11 - Код отримання сторони трикутника:

```
public double getSideB() {  
    if(sideB.getLength()!=0.0) {  
        return sideB.getLength();  
    }  
    else {  
        if(perimeter!=0.0&sideA.getLength()!=0.0&sideC.getLength()!=0.0){  
            sideB.setLength( perimeter-sideC.getLength()-sideA.getLength());  
            return sideB.getLength();  
        }  
        return 0.0;  
    }  
}
```

Передається далі як відповідь у графічний інтерфейс (див. рис. 12).

Рисунок 12 - Програма з відображеною відповіддю:



Для зв'язку з онтологією було використано бібліотеку Owl API.

3.3 Опис використаних технологій

Protégé (версії 5.5.0) для побудови онтології з геометрії. Джава програма, що дозволила побудувати онтологію у форматі Owl.

Java(версії 1.8) – це мова програмування та створення даної програми є об'єктно-орієнтованою мовою програмування, розробленою Sun Microsystems у 1995.

OWL API (1.3.1) - Java бібліотека з відкритим користуванням, для роботи з файлами формату Owl.

IntelliJ IDEA – середовище розробки для різних мов програмування, у тому числі Java від компанії JetBrains. Існує безкоштовна версія або комерційна з розширеними можливостями, університет надає ліцензію на розширену версію.

Висновок

Отже система основана на базі знань - це будь яка система яка може використовувати готову онтологію, інколи дописувати отриманні правила до неї. Потреба виникла у прискоренні процесів, та заміни експертів людей на машин, які рідше помиляються, та можуть надати нові знання.

Існує проблема, що нема такої гарної та універсальної системи яка може зчитати умову задачі так як її подали у підручнику, і надати вирішення. Таким чином було вирішено спробувати написати дану систему.

Для вирішення проблеми розв'язування задач 7 класу з геометрії, була спроба розробки програми що може це робити в автоматичному режимі. Задача не була вирішена повною мірою через брак ресурсів, а саме гарного синтаксичного аналізатора, як було вказано в фінальному алгоритмі програми, ресурсів вистачало тільки на перший варіант вирішення проблеми.

Було проаналізовано багато систем, та вивчено технічної літератури, написання аналізатора являє собою суттєво іншу задачу. Був написаний простий аналізатор SyntaxAnalyzer, що може читати задачу. Крім цього було надано часткове вирішення проблеми, розроблений клас трикутника, точки та відрізка.

Список використаної літератури

1. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень / С.О. Субботін, 2008. – 341с.
2. Бази знань в інтелектуальній системі
https://studopedia.su/9_60863_komponenti-mashini-vivedennya.html
3. Структура БЗ і взаємодія з іншими компонентами ІС
http://ni.biz.ua/18/18_4/18_40645_struktura-bz-i-vzaimodeystvie-s-drugimi-komponentami-is.html
4. Онтологии и тезаурусы: модели, инструменты, приложения / Б. В.Добров, В. В. Иванов, Н. В. Лукашевич, В. Д. Соловьев, 2009. – 173 с.
5. A translation approach to portable ontologies // Knowledge Acquisition, Gruber T.R. 1993.
6. Гаврилова Т.А. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский. – СПб. 2000. – 384 с.
7. CYC technology <https://www.cyc.com/about-us>
8. Pullenti technology <https://www.pullenti.ru/>
9. Protégé technology <https://protege.stanford.edu/about.php>
10. I&W system https://www.militaryfactory.com/dictionary/military-terms-defined.asp?term_id=2595
11. Экспертные системы. Принципы разработки и программирование. 4-е издание / Джозеф Джарратано, Гари Райли.