МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет)_____
Кафедра _____

«До захисту в ЕК»		**«До захисту допущено»**	
Директор інституту(декан факультету)		Завідувач кафедри	
_____	__Форсюк А.В.___	_____	_Чумаченко С.М.__
(підпис)	(прізвище та ініціали)	(підпис)	(прізвище та ініціали)
«___» _____ 2020 р.		«___» _____ 2020 р.	

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

зі спеціальності___122 «Комп'ютерні науки та інформаційні технології»_____
(код та назва спеціальності)
освітньо-професійної програми___Комп'ютерні науки _____
на тему:___Створення електронного глосарію абревіатур у сфері інформаційних технологій. ___

Виконав: здобувач_4__ курсу, групи_7_

____Романович Роман Богданович_____ _____
(прізвище, ім'я, по батькові повністю) (підпис)

Керівник __Литвинов Валерій Андроникович_____ _____
(прізвище , ім'я та по батькові повністю) (підпис)

Консультанти_____ _____
(прізвище та ініціали) (підпис)

_____ _____
(прізвище та ініціали) (підпис)

_____ _____
(прізвище та ініціали) (підпис)

Рецензент *Власенко Лідія Олександрівна* _____
(прізвище та ініціали) (підпис)

Засвідчую, що в цій кваліфікаційній роботі немає запозичень із праць інших авторів без відповідних посилань.

Здобувач_____
(підпис)

Київ – 2020 р.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) __Автоматизації і комп'ютерних систем _____
Кафедра __Інформаційних систем _____
Освітній ступінь___Бакалавр_____
Спеціальність_122 «Комп'ютерні науки та інформаційні технології"_____
(код і назва)
Освітньо-професійна програма _____Комп'ютерні науки_____
(назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри Чумаченко С.М.
_____
"____" _____20___року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

_____Романович Роман Богданович_____
(прізвище, ім'я, по батькові)

1. Тема роботи Створення електронного глосарію абревіатур у сфері інформаційних технологій. _____
_____
_____

керівник роботи__Литвинов В. А. професор, доктор технічних наук_____,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від "27" _04 2020 року №_КС-269_

2. Строк подання здобувачем роботи_____26 травня 2020 року_____

3. Вихідні дані до роботи

1.Термінологія у сфері IT та IC _____
2.Аналоги ETC словників _____
_____

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)_____
1. Системний аналіз предметної області та постановка задачі _____
2. Проектування системи _____
3. Охорона праці _____
_____

5. Перелік графічного матеріалу

1. Логічна модель _____
2.Фізична модель _____
3. Діаграма бази даних _____
4.Скріншоти інтерфейсу програми _____
_____

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Вступ	Литвинов В. А.	27.04.20	30.04.20
Розділ 1	Литвинов В. А.	30.04.20	02.05.20
Розділ 2	Литвинов В. А.	02.05.20	17.05.20
Розділ 3	Литвинов В. А.	17.05.20	25.05.20
Висновки	Литвинов В. А.	25.05.20	26.05.20
Джерела	Литвинов В. А.	18.05.20	20.05.20
Додатки	Литвинов В. А.	18.05.20	20.05.20

7. Дата видачі завдання_________

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
	Системний аналіз діяльності підприємства	27.04.20 - 30.04.20	Виконано
	Постановка задачі на розроблення системи	30.04.20 - 02.05.20	Виконано
	Проектування бази даних	03.05.20 - 07.05.20	Виконано
	Створення додатку користувача	07.05.20 - 15.05.20	Виконано
	Написання інструкції користувача	15.05.20 - 17.05.20	Виконано
	Оформлення пояснювальної записки	18.05.20 - 26.05.20	Виконано
	Оформлення презентації	25.05.20 - 29.05.20	Виконано

Здобувач _____ ____Романович Р.Б._____
(підпис) (прізвище та ініціали)

Керівник роботи _____ ____Литвинов В. А._____
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Виконавцем цієї кваліфікаційної роботи є Романович Роман Богданович.

Темою даної роботи є «Створення електронного глосарію абревіатур у сфері інформаційних технологій».

Кваліфікаційна робота містить 2 основні розділи.

Перший розділ це «СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ».

Другий розділ це «ПРОЕКТУВАННЯ СИСТЕМИ»

Результатом написання кваліфікаційної роботи є створення електронного глосарію абревіатур у сфері інформаційних технологій

КЛЮЧОВІ СЛОВА: ІНФОРМАЦІЙНА СИСТЕМА, ГЛОСАРІЙ, АБРЕВІАТУРА, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ІНТЕРФЕЙС КОРИСТУВАЧА.

ANNOTATION

The executor of this qualification work is Romanovich Roman Bogdanovich.

The topic of this work is "Creating an electronic glossary of abbreviations in the field of information technology."

Qualification work contains 2 main sections.

The first section is "SYSTEM ANALYSIS OF THE SUBJECT AREA AND PROBLEM STATEMENT".

The second section is "SYSTEM DESIGN"

The result of writing a qualifying work is the creation of an electronic glossary of abbreviations in the field of information technology

KEY WORDS: INFORMATION SYSTEM, GLOSSARY, ABBREVIATION, SOFTWARE, USER INTERFACE.

ЗМІСТ

ВСТУП

В епоху сучасних інформаційних технологій існує безліч різноманітних прикладних програм. На ринку програмного забезпечення представлені різноманітні програмні засоби, які з кожним днем все більше і більше охоплюють дуже велике коло задач що виникають у сучасному житті та на виробництві.

Комп'ютери проникли в усі сфери діяльності, починаючи від початкового освіти і до вивченням технологій, вивчення нових видів матерії, невідомих поки людству. Застосування комп'ютерних технологій полегшує процес освіти у середніх та вищих навчальних закладах як самих учнів, студентів, і робочого персоналу.

Завдяки розмаїттям програмного і апаратного забезпечення сьогодні можливо використання всіх можливих комп'ютерних технологій. Це дозволяє зберігати дуже багато інформації, займаючи мінімум місця. Також комп'ютерні технології дозволяють швидко обробляти інформацію і тримати її в захищеному вигляді.

Метою цього дипломного проекту є створення електронного глосарію абревіатур у сфері інформаційних технологій.

РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Терміни і термінологія

1.1.1 Визначення понять «термін» і «абревіатура»

Термін - слово або словосполучення, точно і однозначно іменує поняття і його співвідношення з іншими поняттями в межах спеціальної сфери.

Абревіатура (італ. Abbreviatura від лат. Brevis «короткий») - слово, утворене скороченням слова або словосполучення і читають за алфавітною назвою початкових букв або за початковими звуками слів, що входять в нього.

На даний момент скорочення міцно увійшли в англійську термінологію з інформатики. Переважна більшість абревіатур користується значно більшою популярністю, ніж вихідні терміни і термінологічні поєднання, на базі яких вони були побудовані. З усього різноманіття способів освіти скорочень ініціальна абревіація, усічення, стяжіння, акроніми і гібридних утворень безсумнівно, безсумнівно, лідируюче місце належить і буде належати ініціальній абревіації. Це найпростіший і надзвичайно поширений спосіб утворення абревіатур. У цьому випадку скорочена форма утворюється лише за початковими літерами компонентів термінологічного поєднання або терміну.

Слова, створені таким способом часто незручні для вимови, і тому вимовляються незв'язно, по буквах, які не сприймаючись як єдине слово.

BCC - block check character символ контролю блоку даних

Зрозуміло, немає правил без винятків, і з допомогою ініціальної абревіації випадково можуть бути утворені зручні для вимови, витончені слова.

TIPS - technical inlormation processing system - автоматизована система обробки технічної інформації.

Структурні особливості та функціонування термінів комп’ютерних наук та інформаційних технологій

На кожному етапі розвитку людства можна виокремити сфери життя, які розвиваються особливо бурхливо. Створення кібернетичних машин, їх удосконалення, мініатюризація спричинили формування та розвиток комп'ютерної терміносистеми, а впровадження комп'ютерів у всі сфери людської діяльності викликало перехід лексичних інновацій з вузькоспеціальної комп'ютерної мови до загальнолітературної і, врешті-решт, глобальна комп'ютеризація сприяла інтернаціоналізації комп'ютерної лексики, зокрема і її термінологічних одиниць.

За визначенням лінгвістичного енциклопедичного словника за редакцією В.Н. Ярцевої термін – це слово або словосполучення, що позначає поняття спеціальної галузі знань чи діяльності. Він належить до загальної лексичної системи мови, але лише через конкретну термінологічну систему. Головними вимогами до термінології, яка відповідає міжнародним стандартам, наголошує Т.Р. Кияк, є: відсутність синонімії в одній галузі, обмеження багатозначності, точність і повнота терміну, відсутність експресивності, евфонія, оптимальний рівень інтернаціоналізації та, особливо, вмотивованість термінів. Це мовне явище є актом відображення однієї або багатьох ознак предметів у його назві засобами мови, показником "виправданості" найменування.

Важливими рисами термінів будь-якої галузі є те, що вони точно виражають поняття, процеси та назви речей, які є характерними для тієї або іншої сфери виробництва, співвідносяться з точно визначеними поняттями, спрямовані на однозначність в межах однієї терміносистеми. Одиницями комп'ютерної терміносистеми є терміни – слова або словосполучення, які мають спеціальне, строго визначене значення у комп'ютерній сфері.

З огляду на походження та функціонування комп'ютерна терміносистема складається із термінів, які можна поділити на такі групи:

1. Корелятивні терміни – терміни, що утворюються внаслідок набуття загальновживаних слів специфічних для комп'ютерної галузі значень. У цьому випадку значення терміну є одним із значень загальновживаного слова.

Наприклад, комп’ютерний термін file “об’єм інформації, що має своє позначення – ім’я” сформувався на базі відповідного слова загальновживаної лексики, яке має значення “досьє; підшиті папери”.

2. Загально технічні терміни, що функціонують не лише у межах комп’ютерної терміносистеми, але й в інших галузях науки і техніки. Так, наприклад, термін driver, який у комп’ютерному контексті означає “програма, що керує введенням та виведенням інформації”, в інших галузях має ще більш десятка значень.

3. Спеціальні терміни, характерні лише для комп’ютерної галузі знань. Прикладом можуть слугувати такі терміни, як cybernetics, gigadisc, hardware, software. В таких випадках значення слова і значення терміну співпадають, тому що дане слово слугує лише для вираження одного спеціального поняття, тобто являє собою термін і семантика слова адекватна значенню терміну.

4. Терміни, що мають два і більше значень у комп’ютерній галузі. Так, наприклад, термін server є назвою комп’ютера як устрою, за допомогою якого можна з’єднуватись з Інтернетом, а також програми, що забезпечує цей процес.

Комп’ютерна лексика англійської мови має специфічні особливості. Одна з них – притаманність окремим термінам образності. Хоча в цілому науково-технічні терміни характеризуються стилістичною нейтральністю, окремі комп’ютерні терміни мають яскраво виражений відтінок образності, наприклад, термін mouse, що позначає устрій для регуляції пересування курсору, зовні дійсно нагадує мишу, а назва програми Windows базується на схожості принципу надання інформації у вигляді вікон на дисплеї комп’ютера.

Часте використання скорочень у текстах з проблем комп’ютерних наук та інформаційних технологій, є наслідком прагнення до економії мовних засобів. Серед скорочень, що є одиницями комп’ютерної терміносистеми, можна виділити, наприклад, абревіатури PC – personal computer (персональний комп’ютер), VR – virtual reality (світ, штучно створений за допомогою комп’ютерної техніки); акроніми: CLS – Clear Screen (клавіша, що виконує наказ

“очистити екран”); TELEX – teletypewriter exchange (телекс), DIVOL – digital-tovoice translator (переклад цифрового коду на мовлення); скорочення, що являють собою перші склади слів: DEL – delete (клавіша, що призначається для ліквідації знака), INS – insert (клавіша для вставки символу), avail – availability (коефіцієнт готовності); скорочення, що являють собою “стислі” утворення, наприклад, cnsl – console (пульт управління), comptr – comptometer (комптометр), TPWR – typewriter (пристрій для друкування). Скорочення слова electronic до літери е зустрічається у термінах e-mail, e-cash, e-data.

Скорочення є ширшим поняттям, ніж акронім або абревіатура. Скорочення є способом словотвору, суть якого полягає у відсіканні частини основи, що або збігається зі словом, або є словосполученням, об’єднаним загальним змістом. Скорочення прийнято класифікувати на лексичні й графічні. До лексичних відносять усічені слова (clipped or stump words) і акроніми (initial words or acrol1yms). Скорочуватися можуть будь-які фрагменти слова не залежно від морфемних границь.

Для утворення термінів-акронімів (від грец. акрос – “найвищий, крайній”) використовують частини слів, що входять до складу термінологічних словосполучень, наприклад ALGOL (Algorithic Language) – Алгол; LISP (List Processing) – Ліспп; FORTRAN (Formula Translation) – Фортран.

Акроніми стали настільки популярними, що їхня кількість виправдала публікацію спеціальних словників, до складу яких увійшли скорочення, які вживаються як у загальновживаній лексиці, так і в окремих фахових мовах. Вони становлять особливий інтерес, тому що ілюструють роботу лексичної адаптивної системи мови.

Специфічним англійським підтипом скорочень є напівскорочення, тобто комбінації акроніма одного члена словосполучення із повною основою іншого. Наприклад: A-bomb – atomic bomb, V -day – Victory Day, I-frame – Informational frame (інформаційний кадр), C-BASIK – Commercial BASIK (Бейсік для

комерційного вживання). Слід також звернути увагу на такі способи утворення скорочень, як:

а) використання приголосних (першої і останньої або першої, середньої і останньої) для скорочення слова: ctr (centre), fwd (forward), jr (junior), shtg (shortage), rqs (requirements), ppd (prepaid), rdbl (readable), rd (read), ctg (cotangent), hgt (height);

б) використання початкового складу: libs (liberals), dept (department), nukes (nuclear weapons), recd (received), intrpt (interrupt), inval (invalid);

в) утворення змішаних скорочень: ALGOL (A1gorithic Language), ATM machine (automated teller machine), E-mai1 (Electronic mail), M-way (Motorway).

На відміну від акронімів, для утворення абревіатур використовується лише перші літери слів, що входять до складу початкових термінологічних груп: FOB – free on board; ASAP – as soon as possible. Абревіатура (лат. Abbrevio – скорочую) утворюється з перших літер або з інших частин слів, що входять до складу назви чи поняття. Вони вживаються як в усній, так і писемній мові.

Окрему групу абревіатурних скорочень складають абревіатури, що виникли під час листування у всесвітній мережі Інтернет (Інтернет абревіатури). Найбільш поширені з них такі: BTW – By The Way (до речі); BOT – Back On Topic (повертаючись до теми спілкування); FWIW – For What It’s Worth (справжня ціна); FYI – For Your Information (до вашого відома); GIGO – Garbage In Garbage Out (за що купив за те і продав); HTH – Hope This Helps (сподіваюся, це допоможе). Скорочення доволі часто використовуються в діловому електронному спілкуванні, наприклад: cc (carbon copy – копія); RSVP (абревіатура французького вислову respondez s’il vous p1ait – прошу відповісти); ASAP (as soon as possib1e) – якнайшвидше.

1.1.2 Важливість правильного розуміння термінів для проектантів інформаційних систем, замовників і користувачів

Одна з найбільших проблем в будь-якій сфері діяльності, де використовується специфічна термінологія, - це договір про поняттях. Часто люди використовують слова, взагалі не розуміючи їх значення. І трактують їх якимось своїм особливим чином. Результат - відсутність взаєморозуміння, претензії і невдоволення результатом. Особливо складно пояснювати основні базові поняття. Але в сферах бізнесу і IT технологій без цього не обійтися.

Одна з найпоширеніших проблем при впровадженні IT-систем - дуже високий відсоток провалів.

Спроби пояснити і класифікувати комп'ютерні інформаційні системи привели до появи величезної кількості складних і малозрозумілих термінів. Їх намагаються ділити на класи і підкласи, описувати дивними для широкого кола людей термінами, що призводить до ще більшої плутанини.

Одна з найбільших проблем в IT - відсутність загальної (саме загальної) теоретичної бази

Для спеціалістів у сфері IT дуже важливо знати, і правильно розуміти терміни, щоб правильно робити свою роботу, також щоб вони могли пояснити замовникам, та розробити інструкцію для користувачів, коли будуть розробляти якусь інформаційну систему. Тому термінологічним аспектам присвячено багато нормативно-методичних документів на створення ПЗ та інформаційних систем, зокрема:

ДСТУ 2941-94 Системи оброблення інформації. Розроблення систем. Терміни та визначення;

- ДСТУ ISO/IEC 2382-15:2005 Інформаційні технології. Словник термінів. Частина 15. Мови програмування;

- ДСТУ ISO/IEC 2382-5:2005 Інформаційні технології. Словник термінів. Частина 5. Подання даних;
- ДСТУ ISO/IEC 2382-4:2005 Інформаційні технології. Словник термінів. Частина 4. Організація даних;
- ДСТУ ISO/IEC 2382-17:2005 Інформаційні технології. Словник термінів. Частина 17. Бази даних;

Крім загальних положень і вимог ДСТУ, наявність словника специфічних термінів предметної області вимагається в складі комплекту документації на створення конкретних систем.

1.1.3 Термінологія в сфері ІС та ІТ.

Терміносистема інформаційно-технологічної сфери (далі - ІТ) стала об'єктом різних лінгвістичних досліджень порівняно недавно, а самі терміни ІТ, незважаючи на їх затребуваність, відносяться до найменш вивченим проблемам сучасної лінгвістики. Існує безліч визначень поняття «термін». Спроби лінгвістів сформулювати загальне визначення не увінчалися успіхом, очевидно, через багатогранності даного феномена. Р. І. Комарова пише наступне про цю багатогранності: «Ні одиниці більш багатоликої і невизначеною, ніж термін, причому спостерігається кілька підходів до визначення терміна: одні дослідники намагаються дати йому достатню логічне визначення; інші - намагаються описово розкрити зміст терміна, приписавши йому характерні ознаки; треті - виділяють термін шляхом його протиставлення будь-якої негативної одиниці; четверті шукають суперечливі процедури виділення термінів, щоб прийти потім до суворому визначенню цього поняття; п'яті намагаються дати поки хоча б «робоче» визначення »

Термінологічна лексика в області обчислювальної техніки складається зі слів, словосполучень, що вживаються для вираження зв'язків і відносин між

термінованими поняттями. Ці слова і словосполучення або терміносполучень можна розділити на слова:

1) Прості - Web, word, миша, скайп, модем;

2) Складні інтернет, інтерфейс, веб-сервер; Веб-сервер-комп'ютер, призначений для публікацій гіпертекстових документів.

3) абревіатурні, тобто слова, які утворюються від початкових букв знаменних слів словосполучення. При їх проголошенні за назвами літер наголос падає на останню букву: ЕОМ, ІТ, WWW (World Wide Web) і т.д.

WWW- це система публікації ресурсів, представлених у вигляді гіпертекстових документів

Як відомо, останнім часом дуже часто у вигляді нових термінів з'являються назви техніки, установи, імен винахідників або різних сучасних фірмових виробів: Samsung, Apple, IT і інші. Нові терміни в більшості випадків пишуться з великої літери. Деякі назви з цих термінів в текстах пишуться з малої буквою, так вони отримують широке вживання і розповсюдження в мові, проте з часом зв'язок з їх фірмовою назвою втрачається

Таким чином, аналіз історії обчислювальної техніки дозволяє зробити висновок, що термінологічна лексика, розвиваючись протягом тривалого часу, в даний час сформувалася в строгий лексичний термінологічний фонд.

Термінологія ІТ формувалася насамперед в англійській мові, тому що місце зародження комп’ютерних технологій - США. У українській мові багато загальновживаних слів розширили своє значення, потрапляючи в поле вживання в сфері ІТ, за аналогією з відповідними лексичними одиницями в англійській мові. Особливу роль в терміносистемах грають абревіатури. Всі вони мають в мові певний "прототип" - складну одиницю, піддаючи редукції.

Розвиток науки і техніки призводить до появи нових понять та слів, що їх називають. Актуальність дослідження комп’ютерної термінології зумовлена

необхідністю представити її у вигляді строго впорядкованої системи, яка б відповідала сучасному рівню розвитку науки і запитам практики. Метою статті є визначення напрямів, за якими розвивається комп'ютерна термінологія та її місце в українській мові. Об'єктом дослідження є лексика комп'ютерної сфери, яка постійно розвивається та збагачується. Запровадження української комп'ютерної термінології супроводжується низкою проблем, які дещо ускладнюють вироблення єдиної концепції термінотворення.

Комп'ютерна термінологія є частиною спеціальної комп'ютерної лексики, «яка формується в предметній сфері, технологічно пов'язаній з виробництвом персональних комп'ютерів і програмного забезпечення до них»

Іноді для позначення всієї термінології певної галузі використовують поняття «терміносистема» – «знакова модель певної галузі науки, техніки, мистецтва та ін., яка враховує лексико-семантичні й словотвірні зв'язки між номінаціями – термінами»

Поняття «терміносистема» підкреслює таку важливу властивість термінології, як системність. Термінологічна система може бути розділена на окремі елементи. Ядром, навколо якого формується комп'ютерна лексика, є поняття «комп'ютер». Уся комп'ютерна термінологія певним чином пов'язана з цим поняттям. Як відомо, «архітектура» комп'ютера поділяється на апаратне і програмне забезпечення. Отже, комп'ютерну термінологію можна розділити на напрями, у яких ці поняття є базовими. Тобто можна говорити про термінологічну лексику апаратного забезпечення і термінологічну лексику програмного забезпечення.

Із поширенням персонального комп'ютера в нашу мову потрапила величезна кількість англомовної лексики, і багато звичних понять були замінені запозиченими аналогами.

Українська комп'ютерна лексика має яскраво виражене англомовне забарвлення. Переважна кількість сучасних комп'ютерних термінів є запозиченими з англійської мови: процесор, сканер, інтерфейс, монітор, модем

та ін. Багато лексичних одиниць комп'ютерної лексики (WiFi, URL, LAN), назви додатків (MicrosoftWord), адреси електронних ресурсів (http://www.google.com.ua) фіксуються тільки за допомогою англійської мови, що може викликати труднощі в людей, які не володіють відповідними мовними навичками.

Українські електронні термінологічні словники

Полегшують роботу з термінами автоматичні та автоматизовані системи комп'ютерного перекладання, які набули поширення останнім часом. Серед українських термінологічних словників, які передбачають комп'ютеризацію процесу перекладання, розрізняють термінологічні словники, якими можна користуватися в режимі он-лайн та термінологічні глосарії. Зупинюся на українських електронних термінологічних словниках.

Кілька позицій термінологічних словників, що працюють у режимі он-лайн:

- KirzeN – мультимедійний словник (українська та 40 інших мов);
- LJ Ukrainian dictionary LJ – український словник;
- zakon.nau.ua – словник законодавчих термінів (електронне видання НАУ Online);

Забезпечує Вам пошук документів у трьох базах даних (НАУ-Експерт, НАУ-Русскоязычная, NAU-English), а також перегляд безкоштовних документів.

Нагадуємо, що для коректного відображення баз даних он-лайн у Вас повинен бути встановлений браузер Internet Explorer 6, Mozilla Firefox або Opera.

- Словник банківських та економічних термінів (електронне видання Otpbank)
- короткий математичний словник (електронне видання на formula.com.ua – математика для школи);
- архівістика – російсько-український словник термінів;

Кілька позицій термінологічних глосаріїв:

- словник термінів (інформаційне суспільство) – глосарій української Вікіпедії
- термінологічний словник з моделювання систем (глосарій української Вікіпедії);
- словник термінів на тему «Нерухомість» (глосарій української Вікіпедії);
- метеорологічні та гідрологічні терміни (глосарій української Вікіпедії);
- комп’ютерна термінологія (глосарій української Вікіпедії);
- словник економічних термінів (менеджмент, банківська справа, тощо) – глосарій української Вікіпедії;
- українсько-англійський глосарій термінів Європейського Союзу на сайті europa.dovidka.com.;
- глосарій термінів фондового ринку на сайті Словопедія;
- глосарій «Управління якістю» (Вакуленко А.В.) на сайті Словопедія;
- англійсько-українська мовна пара багатомовного глосарію історичних термінів на сайті social studies.

ABBYY Lingvo — система електронних словників без функції повнотекстового перекладу. Містить близько 8,7 млн словникових статей. До складу програми входить навчальний модуль Lingvo Tutor, який допомагає вносити та запам'ятовувати нові слова. У деяких словниках більшість слів озвучена професійними дикторами.

1.2 Задачі пошуку та ідентифікації термінів

1.2.1 Задачі пошуку по ключу

Одна з дій яка найчастіше зустрічається в програмуванні - пошук. Існує кілька основних варіантів пошуку, і для них створено багато різних алгоритмів. При подальшому розгляді робиться принципове допущення: група даних, в якій необхідно знайти заданий елемент, фіксована. Буде вважатися, що багато з N елементів задано в вигляді такого масиву

a: array[0..N1] of Item

Зазвичай тип Item описує запис з деяким полем, що грає роль ключа. Завдання полягає в пошуку елемента, ключ якого дорівнює заданому аргументу пошуку x. Отриманий в результаті індекс **i**, що задовольняє умові a [i] .key = x, забезпечує доступ до інших полів виявленого елемента. Так як тут розглядається, перш за все, сам процес пошуку, то ми будемо вважати, що тип Item включає тільки ключ.

Лінійний пошук

Якщо немає ніякої додаткової інформації про данні які шукають, то очевидний підхід простий послідовний перегляд масиву зі збільшенням крок за кроком тієї його частини, де бажаного елемента не виявлено. Такий метод називається лінійним пошуком. Умови закінчення пошуку такі:

Елемент знайдений, ai = x.

Весь масив переглянутий і збігів не виявлено.

Це дає нам лінійний алгоритм:

Алгоритм 1.

i:=0;

while (i<N) and (a[i]<>x) do i:=i+1

Слід звернути увагу, що якщо елемент знайдений, то він знайдений разом з мінімально можливим індексом. Це перший з таких елементів. Рівність **i** = N свідчить, що збігів не існує.

Очевидно, що закінчення циклу гарантовано, оскільки на кожному кроці значення **i** збільшується, отже, воно досягне кінцевого числа кроків межі N; фактично ж, якщо збігу не було, це станеться після N кроків

На кожному кроці алгоритму здійснюється збільшення індексу і обчислення логічного виразу. Можна спростити крок алгоритму, якщо спростити логічний вираз, який складається з двох членів. Це спрощення здійснюється шляхом формулювання логічного виразу з одного члену, але при цьому необхідно гарантувати, що збіг відбудеться завжди. Для цього достатньо в кінець масиву помістити додатковий елемент зі значенням x. Такий допоміжний елемент називається бар'єром. Тепер масив буде описаний так:

a: array[0..N] of integer

і алгоритм лінійного пошуку з бар'єром виглядає наступним чином:

Алгоритм 1.

a[N]:=x; i:=0;

while a[i]<>x do i:=i+1

Ясно, що рівність **i** = N свідчить про те, що збігів (якщо не брати до уваги збіги з бар'єром) не було.

Пошук в таблиці

Пошук в масиві іноді називають пошуком в таблиці, особливо якщо ключ сам є складовим об'єктом, таким, як масив чисел або символів. Часто зустрічається саме останній випадок, коли масиви символів називають рядками або словами. Рядковий тип визначається так:

String = array[0..M√1] of char

відповідно визначається і відношення порядку для рядків x і y

x = y, якщо xj = yj для 0 face="Symbol" =< j < M

x < y, якщо xi < yi для 0 face="Symbol" =< i < M і xj = yj для 0 face="Symbol"=< j < i

Для того щоб встановити факт збігу, необхідно встановити, що всі символи порівнюваних рядків відповідно рівні один одному. Тому порівняння складових операндів зводиться до пошуку їх незбіжних частин, до пошуку на нерівність. Якщо нерівних частин не існує, то можна говорити про рівність. Припустимо, що розмір слів досить малий, скажімо, менше 30. В цьому випадку можна використовувати лінійний пошук і діяти таким чином.

Для більшості практичних застосувань бажано виходити з того, що рядки мають змінний розмір. Це передбачає, що розмір вказується в кожному окремому рядку. Якщо виходити з раніше описаного типу, то розмір не повинен перевищувати максимального розміру M. Така схема досить гнучка і підходить для багатьох випадків, в той же час вона дозволяє уникнути складнощів динамічного розподілу пам'яті. Найчастіше використовуються два таких уявлення розміру рядків:

Розмір неявно вказується шляхом додавання кінцевого символу, більше цей символ ніде не вживається. Зазвичай для цієї мети використовується недрукований Символ із 00h. (Для подальшого важливо, що це мінімальний символ зі всієї безлічі символів.)

Розмір явно зберігається в якості першого елемента масиву, рядок s має наступний вигляд: s = s0, s1, s2, ..., sM-1. Тут s1, ..., sM-1 фактичні символи рядка, а s0 = Chr (M). Такий прийом має ту перевагу, що розмір явно доступний, недолік ж у тому, що цей розмір обмежений кількістю символів (256).

В подальшому алгоритмі пошуку віддається перевага першій схемі. У цьому випадку порівняння рядків виконується так

i:=0;

while (x[i]=y[i]) and (x[i]<>00h) do i:=i+1

Кінцевий символ працює тут як бар'єр.

Тепер повернемося до задачі пошуку в таблиці. Він вимагає вкладених пошуків, а саме: пошуку по рядках таблиці, а для кожного рядка послідовних порівнянь між компонентами. Наприклад, нехай таблиця T і аргумент пошуку x визначаються таким чином:

T: array[0..N-1] of String;

x: String

Прямий пошук рядка

Часто доводиться стикатися зі специфічним пошуком, так званим пошуком рядка. Його можна визначити в такий спосіб. Нехай заданий масив s з N елементів і масив p з M елементів, причому 0 <M face = "Symbol" = <N. Описані вони так:

s: array[0..N1] of Item

p: array[0..M1] of Item

Пошук рядка виявляє перше входження p в s. Зазвичай Item це символи, тобто s можна вважати деяким текстом, а p словом, і необхідно знайти перше входження цього слова в зазначеному тексті. Ця дія типова для будь-яких систем обробки текстів, звідси і очевидна зацікавленість у пошуку ефективного алгоритму для цього завдання. Розберемо алгоритм пошуку, який будемо називати прямим пошуком рядка.

Алгоритм 3.

i:=-1;

repeat

i:=i+1; j:=0;

```
        while (j<M) and (s[i+j]=p[j]) do j:=j+1;
until (j=M) or (i=N-M)
```

Вкладений цикл з передумовою починає виконуватися тоді, коли перший символ слова p збігається з черговим, i-м, символом тексту s. Цей цикл повторюється стільки разів, скільки збігається символів тексту s, починаючи з i-го символу, з символами слова p (максимальна кількість повторень рівне M). Цикл завершується при вичерпанні символів слова p (перестає виконуватися умова j <M) або при розбіжності чергових символів s і p (перестає виконуватися умова s [i + j] = p [j]). Якщо збіг стався з усіма символами слова p (тобто слово p знайдено), то виконується умова j = M, і алгоритм завершується. В іншому випадку пошук триває до тих пір, поки не переглянутою залишиться частина тексту s, яка містить символів, менше, ніж є в слові p (тобто цей залишок вже не може збігтися зі словом p). В цьому випадку виконується умова i = N-M, що теж призводить до завершення алгоритму. Це показує гарантованість закінчення алгоритму.

Цей алгоритм працює досить ефективно, якщо допустити, що розбіжність пари символів відбувається після незначної кількості порівнянь у внутрішньому циклі. При великій потужності типу Item це досить частий випадок. Можна припускати, що для текстів, складених з 128 символів, розбіжність буде виявлятися після однієї або двох перевірок. Проте, в гіршому випадку продуктивність буде викликати побоювання.

1.2.2 Завдання повнотекстового пошуку

Повнотекстовий пошук – пошук по всьому вмісту документа. Приклад повнотекстового пошуку – будь-яка пошукова система Інтернет, наприклад www.google.com, www.yandex.ru. Як правило, повнотекстовий пошук для прискорення пошуку використовує попередньо побудовані індекси. Найбільш поширеною технологією для індексів повнотекстового пошуку є інвертовані індекси.

Алгоритм Кнута, Моріса і Пратта

Приблизно в 1970 р Д. Кнут, Д. Моріс і В. Пратт винайшли алгоритм, який фактично вимагає тільки N порівнянь навіть в найгіршому випадку. Новий алгоритм ґрунтується на тому міркуванні, що після часткового збігу початкової частини слова з відповідними символами тексту фактично відома пройдена частина тексту і можна визначити деякі відомості (на основі самого слова), за допомогою яких потім можна швидко просунутися по тексту. Наведений приклад пошуку слова ABCABD показує принцип роботи такого алгоритму. Символи, які зазнали порівняння, тут підкреслені. Зверніть увагу: при кожній розбіжності пари символів слово зсувається на всю пройдену відстань, оскільки менші зсуви не можуть привести до повного збігу.

```
ABCABCABAABCABD
ABCABD
  ABCABD
   ABCABD
    ABCABD
     ABCABD
```

Основною відмінністю КМП-алгоритму від алгоритму прямого пошуку є здійснення зсуву слова не на один символ на кожному кроці алгоритму, а на деяку змінну кількість символів. Таким чином, перед тим як здійснювати черговий зсув, необхідно визначити величину зсуву. Для підвищення ефективності алгоритму необхідно, щоб зсув на кожному кроці був би якомога більшим.

Якщо j визначає позицію в слові, що містить перший неспівпадаючий символ (як в алгоритмі прямого пошуку), то величина зсуву визначається як j-D. Значення D визначається як розмір найдовшої послідовності символів слова, які безпосередньо передують позиції j, яка повністю збігається з початком слова. D залежить тільки від слова і не залежить від тексту. Для кожного j буде своя величина D, яку позначимо dj.

Так як величини dj залежать тільки від слова, то перед початком фактичного пошуку можна обчислити допоміжну таблицю d; ці обчислення зводяться до деякої предтрансляції слова. Відповідні зусилля будуть виправданими, якщо розмір тексту стане значно перевищувити розмір слова (M << N). Якщо потрібно шукати багато входжень одного і того ж слова, то можна користуватися одними і тими ж d. Наведені приклади пояснюють функцію d.

Останній приклад на рис. 2.2 показує: так як pj рівне A замість F, то відповідний символ тексту не може бути символом A через те, що умову si <> pj закінчує цикл. Отже, зсув на 5 не призведе до подальшого збігу, і тому можна збільшити розмір зсуву до 6. З огляду на це, визначаємо обчислення dj як пошук найдовшого збігання послідовності з додатковим обмеженням pdj <> pj. Якщо ніяких збігів немає

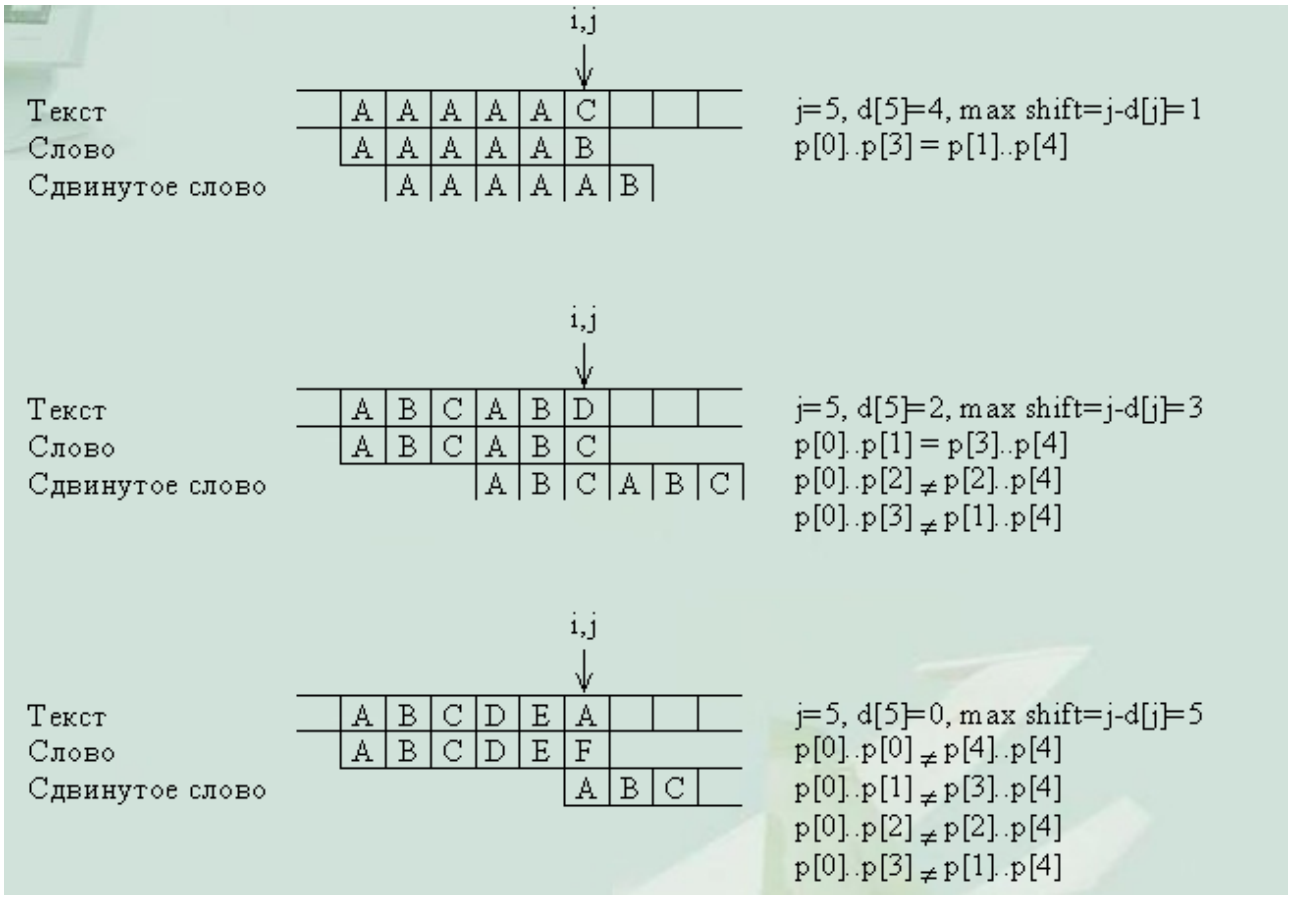


Рис. 2.2. Частковий збіг зі словом і обчислення d j.

то вважається dj = -1, що вказує на зсув на ціле слово щодо його поточної позиції. Наступна програма демонструє КМП-алгоритм.

```
Program KMP;
const
	Mmax = 100; Nmax = 10000;
var
	i, j, k, M, N: integer;
	p: array[0..Mmax-1] of char; {слово}
	s: array[0..Mmax-1] of char; {текст}
	d: array[0..Mmax-1] of integer;
begin
{Введення тексту s і слова p}
Write('N:'); Readln(N);
Write('s:'); Readln(s);
Write('M:'); Readln(M);
Write('p:'); Readln(p);
{ Заповнення масиву d}
j:=0; k:=-1; d[0]:=-1;
while j<(M-1) do begin
	while(k>=0) and (p[j]<>p[k]) do k:=d[k];
	j:=j+1; k:=k+1;
	if p[j]=p[k] then
		d[j]:=d[k]
	else
		d[j]:=k;
end;
{Пошук слова p в тексті s}
i:=0; j:=0;
while (j<M) and (i<N) do begin
	while (j>=0) and (s[i]<>p[j]) do j:=d[j]; {Зсув слова}
```

```
        i:=i+1; j:=j+1;
end;
{ Виведення результату пошуку }
if j=M then Writeln('Yes') { знайдений }
else Writeln('No'); {не знайдений }
Readln;
end.
```

Точний аналіз КМП-пошуку, як і сам його алгоритм, досить складний. Його винахідники доводять, що потрібно близько M + N порівнянь символів, що значно краще, ніж M * N порівнянь з прямого пошуку. Вони так само наголошують на тому позитивну властивість, що покажчик сканування і ніколи не повертається назад, в той час, як при прямому пошуку після розбіжності перегляд завжди починається з першого символу слова і тому може включати символи, які раніше вже були видимими. Це може привести до негативних наслідків, якщо текст читається з вторинної пам'яті, адже в цьому випадку повернення обходиться дорого. Навіть при буферизованному введенні може зустрітися настільки велике слово, що повернення перевищить ємність буфера.

Алгоритм Боуера і Мура

КМП-пошук дає справжній виграш тільки тоді, коли невдачі передувало деяке число збігів. Лише в цьому випадку слово зсувається більш ніж на одиницю. На жаль, це швидше виняток, ніж правило: збіги зустрічаються значно рідше, ніж розбіжності. Тому виграш від використання КМП-стратегії в більшості випадків пошуку в звичайних текстах дуже незначний. Метод же, запропонований Р. Боуером і Д. Муром в 1975 р, не тільки покращує обробку найгіршого випадку, але дає виграш в проміжних ситуаціях.

БМ-пошук ґрунтується на незвичайному розумінні, порівняння символів починається з кінця слова, а не з початку. Як і в разі КМП-пошуку, слово перед

фактичним пошуком трансформується в деяку таблицю. Нехай для кожного символу x з алфавіту величина dx відстань від самого правого в слові входження x до правого кінця слова. Уявімо собі, що виявлено розбіжність між словом і текстом. У цьому випадку слово відразу ж можна зрушити вправо на dpM-1 позицій, тобто на число позицій, швидше за все більше одиниці. Якщо неспівпадаючий символ тексту в слові взагалі не зустрічається, то зсув стає навіть більшим, а саме зсувати можна на довжину всього слова. Ось приклад, який ілюструє цей процес:

```
ABCABCABFABCABD
ABCABD        {Не совпало с 'C', d['C']=3}
  ABCABD      {Не совпало с F, 'F' нет в слове}
     ABCABD   {Полное совпадение, слово найдено}
```

Нижче наводиться програма зі спрощеною стратегією Боуера-Мура, побудована так само, як і попередня програма з КМП-алгоритмом. Зверніть увагу на таку деталь: у внутрішньому циклі використовується цикл з repeat, де перед порівнянням s і p збільшуються значення k і j. Це дозволяє виключити в індексних виразах складову -1.

```
Program BM;
const
    Mmax = 100; Nmax = 10000;
var
    i, j, k, M, N: integer;
    ch: char;
    p: array[0..Mmax-1] of char; {слово}
    s: array[0..Nmax-1] of char; {текст}
    d: array[' '..'z'] of integer;
begin
```

```
{Введення тексту s і слова p}
Write('N:'); Readln(N);
Write('s:'); Readln(s);
Write('M:'); Readln(M);
Write('p:'); Readln(p);
{Заполнение массива d}
for ch:=' ' to 'z' do d[ch]:=M;
for j:=0 to M-2 do d[p[j]]:=M-j-1;
{Пошук слова p в тексті s}
i:=M;
repeat
        j:=M; k:=i;
        repeat {Цикл порівняння символів}
                k:=k-1; j:=j-1; {слова, починаючи з правого.}
        until (j<0) or (p[j]<>s[k]); {Вихід, якщо порівняли всі }
        {слово або розбіжність. }
        i:=i+d[s[i-1]]; {Зсув слова вправо }
until (j<0) or (i>N);
{Виведення результату пошуку}
if j<0 then Writeln('Yes') {знайдений }
else Writeln('No'); {не знайдений }
Readln;
end.
```

У більшості випадків, крім спеціально побудованих прикладів, даний алгоритм вимагає значно менше N порівнянь. У найсприятливіших обставинах, коли останній символ слова завжди потрапляє на неспівпадаючий символ тексту, число порівнянь рівно N / M.

Автори алгоритму приводять і кілька міркувань з приводу подальших удосконалень алгоритму. Одне з них об'єднати наведену щойно стратегію, що забезпечує великі зсуви в разі розбіжності, зі стратегією Кнута, Морріса і Пратта, що допускає відчутні зсуви при виявленні збігу (часткового). Такий метод вимагає двох таблиць, одержуваних при предтрансляції: d1 тільки що згадана таблиця, а d2 таблиця, відповідна КМП-алгоритму. З двох зсувів вибирається більший, причому і той і інший говорять, що ніякий менший зсув не може привести до збігу.

1.3. Постановка задачі

1.3.1 Призначення та цілі створення системи

Створена мною система буде доцільною, так як вона дає змогу пошуку потрібних термінів у сфері ІТ та ІС, не лише за певною літерою, як було у розглянутих аналогах ЕТС які є в Україні, а також буде можливість пошуку за потрібною абревіатурою, також реалізовано повнотекстовий пошук, по всім полям і записам існуючої бази термінів системи.

Також, є можливість використовувати систему без підключення до Інтернету, так як не завжди є можливість вийти в мережу, це є перевагою.

Користувачами системи будуть адміністратор, який буде наповняти систему термінами та звичайні користувачі(студенти) які будуть користуватись системою з метою навчання, ознайомленням або повторенням термінології у сфері ІТ та ІС.

Система повинна забезпечувати зручний та автоматизований спосіб отримання інформації про терміни у сфері ІТ та ІС та надавати інструменти для фільтрування, пошуку та редагування даних.

1.3.2 Вимоги до створюваної системи

Розробка та використання єдиної мови спілкування користувачів з системою;

– Зручність використання системою (інтерфейсу користувача) на всіх ієрархічних рівнях системи;

– Створення єдиної термінологічної системи, що включає також єдину систему уніфікації термінів, що мають однакове смислове навантаження;

– Просте адміністрування.

– Зручність використання

1.3.2.1 Вимоги до апаратного забезпечення

- Процесор з тактовою частотою 1.5 GHz або краще;
- Оперативна пам'ять(RAM) 2 Гбайт або більше;
- Відеокарта з підтримкою Directx 9.0 з роздільною здатністю не менше 1024x768, з 256 Мбайт відеопамяті.
- Жорсткий диск(HDD): 1 Гбайт вільного простору.

1.3.2.2 Вимоги до програмного забезпечення

- Операційна система: Windows (XP,Vista,7, 8, 10).

1.3.2.3 Вимоги до програмного забезпечення

- Microsoft.NET Framework 4.0 або новіше;

1.3.2.4 Вимоги до СУБД

Microsoft SQL SERVER 2018.

Збереження даних системи необхідно забезпечити через спільну інформаційну базу даних (на сервері) та надати користувачам доступ до даних, згідно визначених прав.

1.3.3 Функції, які має виконувати система.

Система повинна виконувати такі функції:

1. Перегляд даних.
2. Внесення даних
3. Редагування.
4. Пошук.
    - за заданою абревіатурою;
    - повнотекстовий пошук за заданими ключовими словами.

1.3.4 Вхідні та вихідні дані системи

Вхідними даними для інформаційної системи є:

Термінологія у сфері ІТ та ІС. У формі «Наповнення глосарію» адміністратору надається змогу наповнити систему абревіатурами і їх поясненнях на англійській, українській та російській, також є змога на редагування і видалення потрібних записів.

Вихідною інформацією є:

База існуючих у системі термінів у сфері ІТ та ІС. Для звичайних користувачів буде можливість переглянути усі існуючі записи (без змоги редагування), також буде можливість фільтрувати терміни за буквою, та знайти потрібний один або декілька термінів, які будуть відповідати умові пошуку.

РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ

2.1 Обґрунтування вибору засобів розробки системи

Для створення інформаційної системи згідно теми дипломної роботи була обрана платформа Microsoft VisualStudio 2017. В якості мови реалізації програми обраний C #.

В даний час обов'язкової можливістю вважається візуальне проектування, коли програміст будує свої додатки, використовуючи готові модулі. Прикладом можуть служити всі сучасні пакети для розробників - BorlandDelphi, Microsoft VisualStudio 2017 і т.д.

Щоб кошти розробки та технології відповідали вимогам розробників, в корпорації Майкрософт була створена абсолютно нова модель програмування для доступу до даних, заснована на .NET Framework. Побудова на основі .NET Framework гарантує однаковість доступу до даних: компоненти використовують систему загальних типів, загальні шаблони розробки та угоди про просторах імен.

В .NET Framework підтримується пряма і зворотна сумісність. В контексті .NET Framework зворотна сумісність означає, що будь-який додаток, створене в .NET Framework більш ранньої версії, буде виконуватися і в пізніших версіях.

Класи ADO.NET були розроблені для підтримки можливостей нової моделі програмування: інтеграції з XML, єдиного уявлення даних з можливістю комбінування даних з різних джерел, а також засобів оптимізації взаємодії з базою даних, представлених в .NET Framework.

.NET являє собою абсолютно новий спосіб створення розподілених настільних і вбудованих додатків. Для типів .NET не потрібні ні фабрики класів, ні підтримка Unknown, ні реєстрація в системному реєстрі. Ці основні елементи COM не приховані - їх просто більше немає.

Спеціально для нової платформи Microsoft розробила мову програмування - C#, який ввібрав в себе багато чого з того кращого, що є в самих різних мовах програмування, і так само є складовою частиною Microsoft VisualStudio 2017.

Платформа .NET є повністю незалежною від використовуваних мов програмування. Можна використовувати декілька .NET-сумісних мов програмування навіть в рамках одного проекту.

Основні можливості .NET наступні:

- повні можливості взаємодії з існуючими кодом;
- повне і абсолютне міжмовне взаємодія, міжмовна обробка виключень і міжмовне налагодження;
- загальне середовище виконання для будь-яких додатків .NET, незалежно від того, якими мовами вони були створені. Один з важливих моментів при цьому - те, що для всіх мов використовується один і той же набір вбудованих типів даних;
- бібліотека базових класів, яка забезпечує приховування всіх складнощів, пов'язаних з безпосереднім використанням викликів API, пропонує цілісну об'єктну модель для всіх мов програмування, що підтримують .NET;
- дійсне спрощення процесу розгортання програми.

В .NET немає необхідності реєструвати подвійні типи в системному реєстрі. .NET дозволяє різними версіями одного і того ж модуля DLL мирно співіснувати на одному комп'ютері.

Microsoft VisualStudio 2017 підтримує технології Microsoft .NET Framework вже у версії Microsoft .NET Framework v4.7.2, які надають загальномовного середовища виконання і уніфіковані класи програмування.

Для проектування моделі бази даних було використано CASE засіб ErwinDataModeller з подальшою генерацією на сервер Microsoft SQL Server 2018.

SQL — декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними БД, створення схеми бази даних і її модифікації, системи контролю за доступом до бази даних. Сам по собі SQL не є ні системою керування базами даних, ні окремим програмним продуктом. Не

будучи мовою програмування в тому розумінні, як С або Pascal, SQL може формувати інтерактивні запити або, будучи вбудованою в прикладні програми, виступати в якості інструкцій для керування даними. Стандарт SQL, крім того, вміщує функції для визначення зміни, перевірки і захисту даних.

Для керування базою даних було обрано MS SQL Server 2018, вибір було зупинено саме на цій СУБД через ряд наступних переваг MS SQL Server 2018 над іншими СУБД:

1. Добре поєднується з усіма видами продуктів Microsoft.
2. Вартість. MS SQL - це безкоштовний SQL сервер, який має величезні можливості для роботи з високим ступенем завантаження.
3. Сервер не має обмежень за розміром бази даних і може витримати практично необмежене навантаження.
4. SQL Server 2018 в порівнянні з іншими СУБД має більш низькі вимоги до апаратного забезпечення
5. MS SQL це SQL сервер високого класу, який піклується про управління базою даних, її безпеки і стабільності роботи.
6. MS SQL забезпечує найвищий рівень захисту даних.
7. Простота в управлінні.

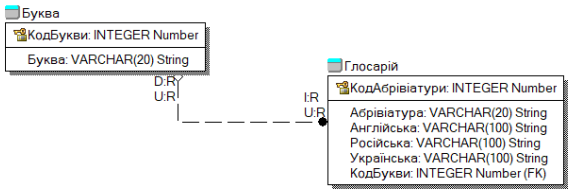
Середовище розробки використовувався VisualStudio 2017.
серія продуктів фірми Майкрософт, які включають інтегроване середовище розробки програмного забезпечення та ряд інших інструментальних засобів. Ці продукти дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, що підтримуються Microsoft Windows, Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NET CompactFramework та Microsoft Silverlight.

2.2 Проектування бази даних

Для побудови бази даних (логічної та фізичної) була застосована CASE - технологія ERwin Data Modeler r7.3 для проектування та документування баз даних, яке дозволяє створювати, документувати і супроводжувати бази даних, сховища і вітрини даних. Моделі даних допомагають візуалізувати структуру даних, забезпечуючи ефективний процес організації, управління і адміністрування таких аспектів діяльності підприємства, як рівень складності даних, технологій баз даних та середовища розгортання..

ERwin підтримує генерацію схеми БД та її опис на мови цільової СУБД таких як ORACLE, Informix, Microsoft SQL Server 2018 та ін. Побудова моделі даних передбачає визначення атрибутів і сутностей, тобто необхідність визначення яка інформація буде зберігатися в конкретній сутності.

Логічна модель (див. Додаток) відображає об’єктно-орієнтовану декомпозицію предметної області, для якої створюється система. На логічній моделі не визначаються типи даних та індекси для таблиць.



Фізична модель (див. Додаток) бази даних описує засоби фізичної реалізації логічного проекту бази даних. Специфіка конкретної СУБД може включати в себе обмеження на іменування об'єктів бази даних, обмеження на підтримувані типи даних, і т.д.

База даних складається з 2 таблиць (див. Додаток)

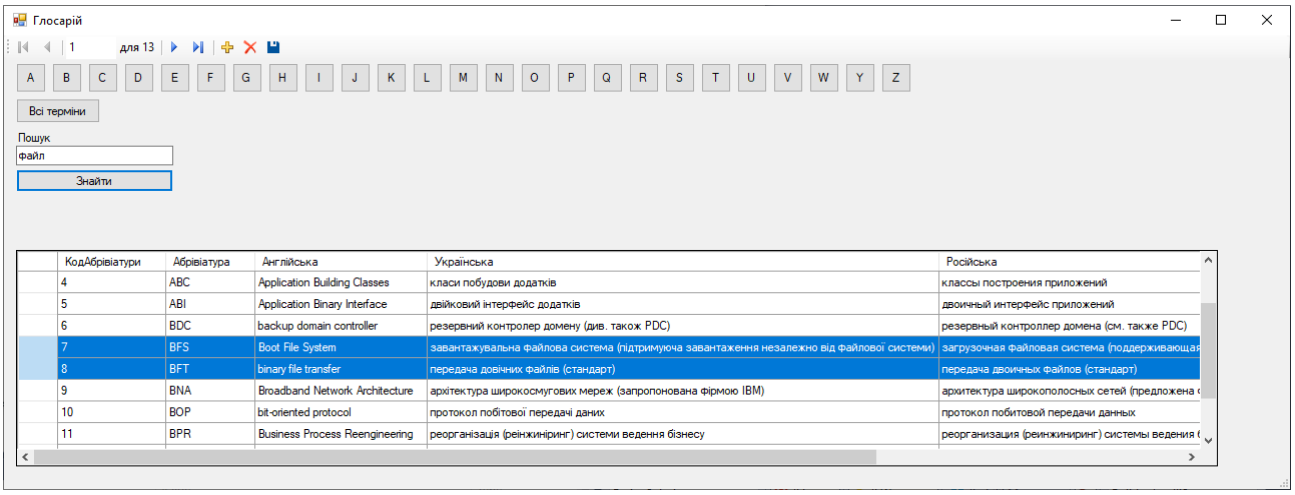
2.3 Реалізація пошуку термінів

Повнотекстовий пошук по всім рядкам і стовбцям БД реалізовано так:

```
for (int i = 0; i < глосарійDataGridView.RowCount; i++)
{
    глосарійDataGridView.Rows[i].Selected = false;
    for (int j = 0; j < глосарійDataGridView.ColumnCount; j++)
        if (глосарійDataGridView.Rows[i].Cells[j].Value != null)
            if (глосарійDataGridView.Rows[i].Cells[j].Value.ToString().Contains(textBox2.Text))
            {
                глосарійDataGridView.Rows[i].Selected = true;
                break;
            }
}
```

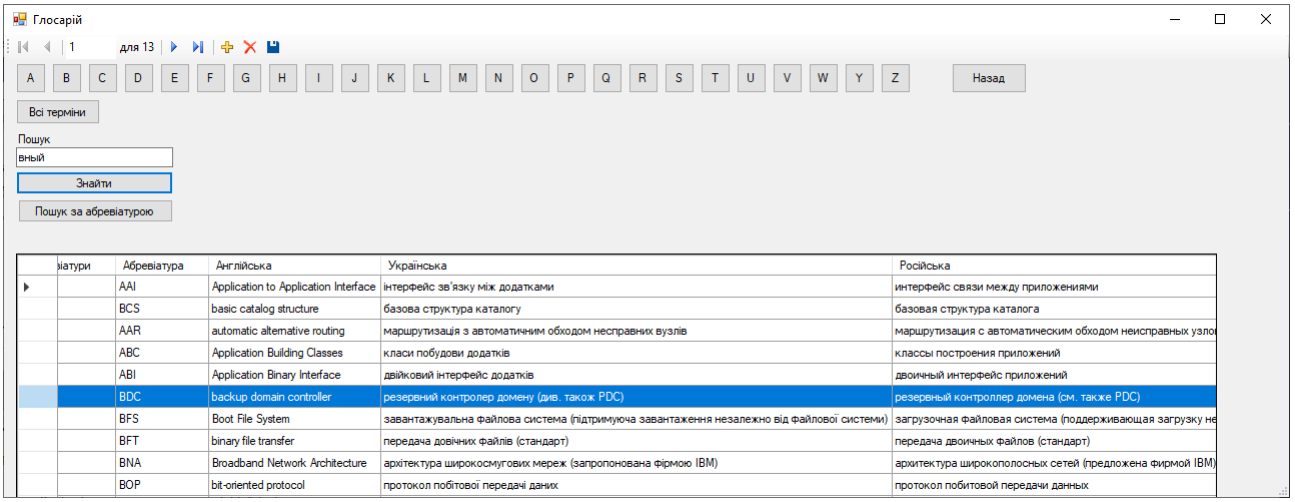
Приклад того, як працює пошук:

1. Приклад того як працює пошук, при пошуку частини слова на українській мові:



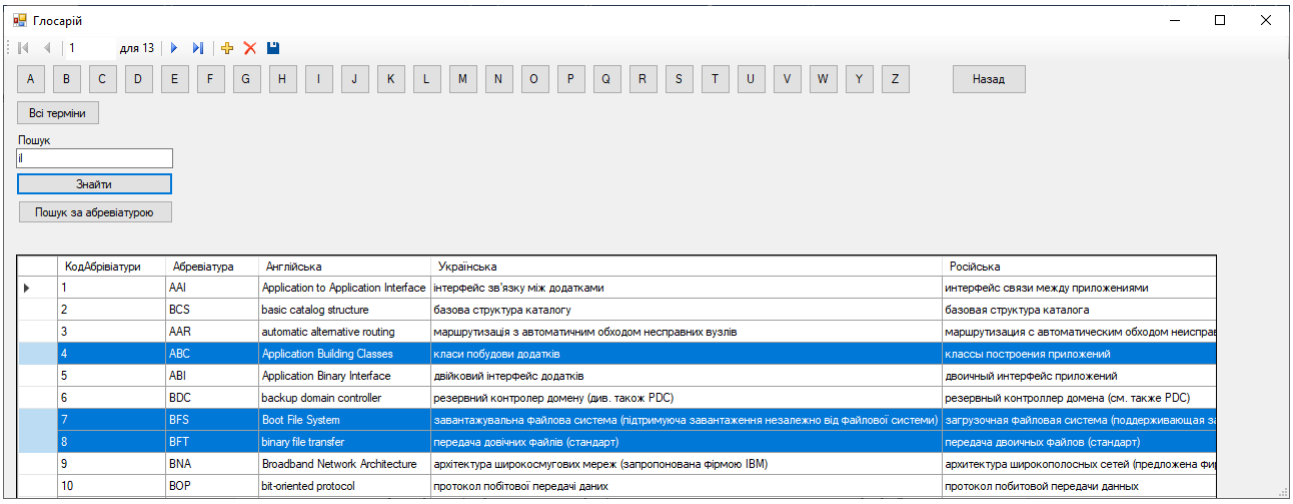
КодАбрівіатури	Абрівіатура	Англійська	Українська	Російська
4	ABC	Application Building Classes	класи побудови додатків	классы построения приложений
5	ABI	Application Binary Interface	двійковий інтерфейс додатків	двоичный интерфейс приложений
6	BDC	backup domain controller	резервний контролер домену (див. також PDC)	резервный контроллер домена (см. также PDC)
7	BFS	Boot File System	завантажувальна файлова система (підтримуюча завантаження незалежно від файлової системи)	загрузочная файловая система (поддерживающая
8	BFT	binary file transfer	передача довічних файлів (стандарт)	передача двоичных файлов (стандарт)
9	BNA	Broadband Network Architecture	архітектура широкосмугових мереж (запропонована фірмою IBM)	архитектура широкополосных сетей (предложена
10	BOP	bit-oriented protocol	протокол побітової передачі даних	протокол побитовой передачи данных
11	BPR	Business Process Reengineering	реорганізація (реінжиніринг) системи ведення бізнесу	реорганизация (реинжиниринг) системы ведения

2. Приклад того як працює пошук, при пошуку частини слова на російській мові:



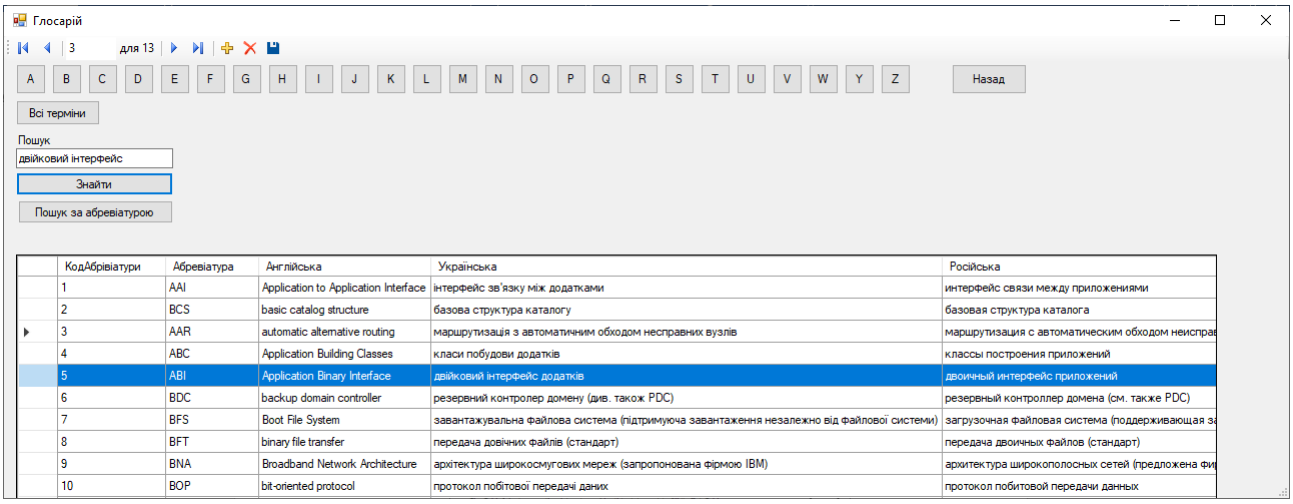
іатури	Абревіатура	Англійська	Українська	Російська
	AAI	Application to Application Interface	інтерфейс зв'язку між додатками	интерфейс связи между приложениями
	BCS	basic catalog structure	базова структура каталогу	базовая структура каталога
	AAR	automatic alternative routing	маршрутизація з автоматичним обходом несправних вузлів	маршрутизация с автоматическим обходом неисправных узло
	ABC	Application Building Classes	класи побудови додатків	классы построения приложений
	ABI	Application Binary Interface	двійковий інтерфейс додатків	двоичный интерфейс приложений
	BDC	backup domain controller	резервний контролер домену (див. також PDC)	резервный контроллер домена (см. также PDC)
	BFS	Boot File System	завантажувальна файлова система (підтримуюча завантаження незалежно від файлової системи)	загрузочная файловая система (поддерживающая загрузку не
	BFT	binary file transfer	передача довічних файлів (стандарт)	передача двоичных файлов (стандарт)
	BNA	Broadband Network Architecture	архітектура широкосмугових мереж (запропонована фірмою IBM)	архитектура широкополосных сетей (предложена фирмой IBM)
	BOP	bit-oriented protocol	протокол побітової передачі даних	протокол побитовой передачи данных

3. Приклад того як працює пошук, при пошуку частини слова на англійській мові:



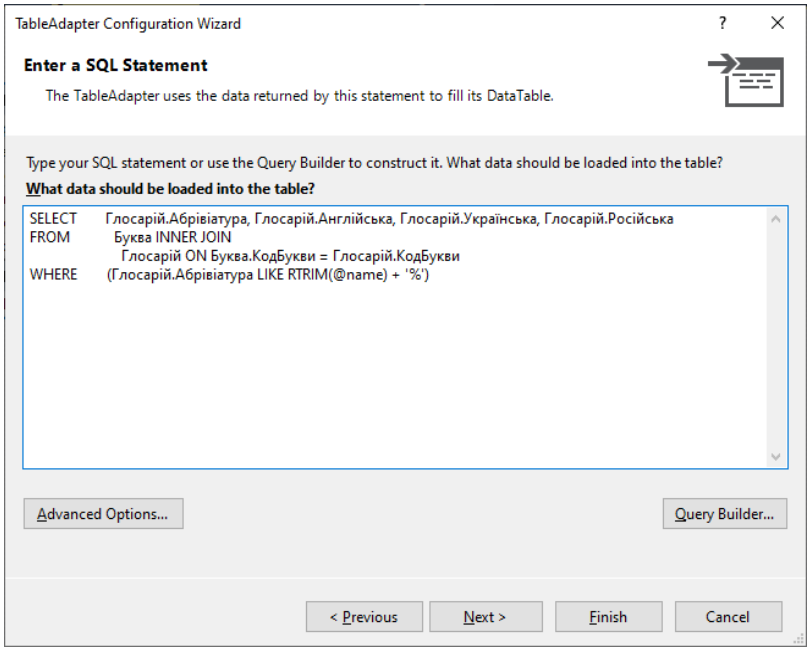
КодАбрівіатури	Абревіатура	Англійська	Українська	Російська
1	AAI	Application to Application Interface	інтерфейс зв'язку між додатками	интерфейс связи между приложениями
2	BCS	basic catalog structure	базова структура каталогу	базовая структура каталога
3	AAR	automatic alternative routing	маршрутизація з автоматичним обходом несправних вузлів	маршрутизация с автоматическим обходом неиспра
4	ABC	Application Building Classes	класи побудови додатків	классы построения приложений
5	ABI	Application Binary Interface	двійковий інтерфейс додатків	двоичный интерфейс приложений
6	BDC	backup domain controller	резервний контролер домену (див. також PDC)	резервный контроллер домена (см. также PDC)
7	BFS	Boot File System	завантажувальна файлова система (підтримуюча завантаження незалежно від файлової системи)	загрузочная файловая система (поддерживающая з
8	BFT	binary file transfer	передача довічних файлів (стандарт)	передача двоичных файлов (стандарт)
9	BNA	Broadband Network Architecture	архітектура широкосмугових мереж (запропонована фірмою IBM)	архитектура широкополосных сетей (предложена Фи
10	BOP	bit-oriented protocol	протокол побітової передачі даних	протокол побитовой передачи данных

4. Приклад того як працює пошук, при пошуку за декількома словами:

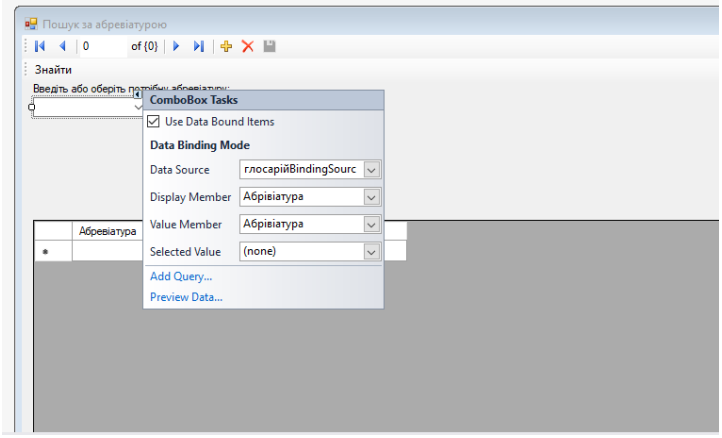


КодАбрівіатури	Абревіатура	Англійська	Українська	Російська
1	AAI	Application to Application Interface	інтерфейс зв'язку між додатками	интерфейс связи между приложениями
2	BCS	basic catalog structure	базова структура каталогу	базовая структура каталога
3	AAR	automatic alternative routing	маршрутизація з автоматичним обходом несправних вузлів	маршрутизация с автоматическим обходом неиспра
4	ABC	Application Building Classes	класи побудови додатків	классы построения приложений
5	ABI	Application Binary Interface	двійковий інтерфейс додатків	двоичный интерфейс приложений
6	BDC	backup domain controller	резервний контролер домену (див. також PDC)	резервный контроллер домена (см. также PDC)
7	BFS	Boot File System	завантажувальна файлова система (підтримуюча завантаження незалежно від файлової системи)	загрузочная файловая система (поддерживающая з
8	BFT	binary file transfer	передача довічних файлів (стандарт)	передача двоичных файлов (стандарт)
9	BNA	Broadband Network Architecture	архітектура широкосмугових мереж (запропонована фірмою IBM)	архитектура широкополосных сетей (предложена Фи
10	BOP	bit-oriented protocol	протокол побітової передачі даних	протокол побитовой передачи данных

Для реалізації пошуку за певною абревіатурою, створюю потрібний запит:

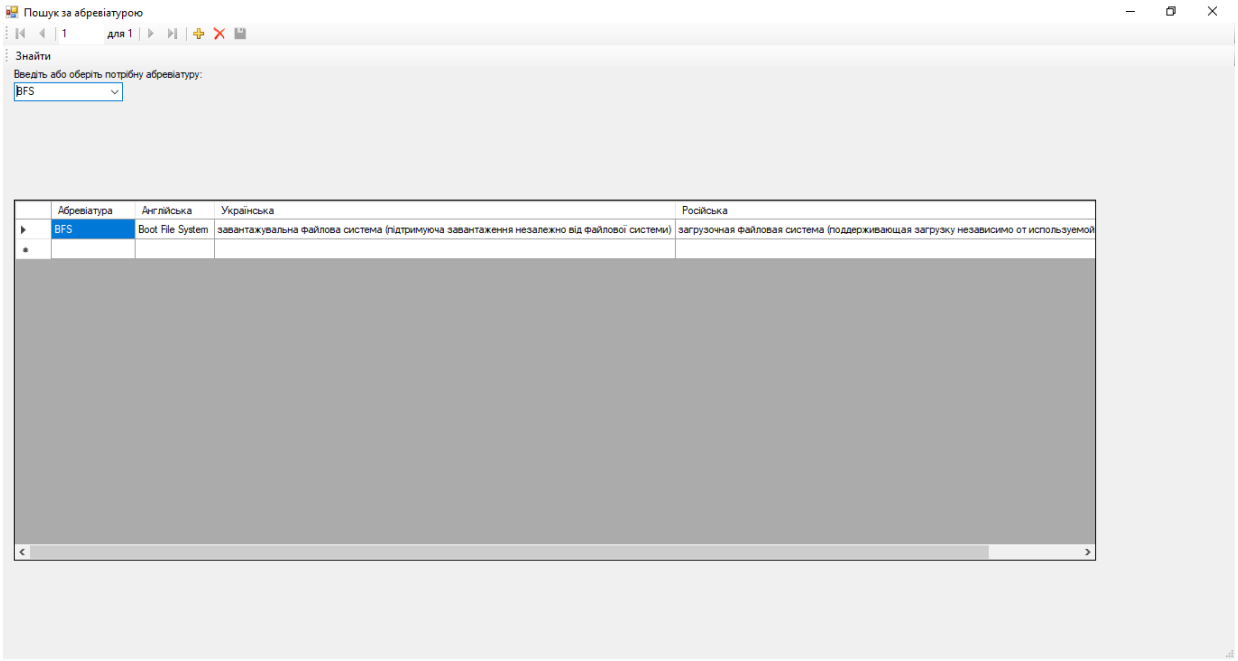


Далі створюю та налаштовую форму де користувач зможе здійснювати пошук за потрібною йому абревіатурою. Для цього на форму розміщуємо табличний вигляд нашого запиту та поля для вводу даних:



На елемент Combox вказуємо шлях до абревіатур, щоб користувач мав можливість обрати самому інформацію про абревіатуру, яку він хоче переглянути. Можливість самому ввести абревіатуру також є:

Результат запиту на пошук за потрібною абревіатурою

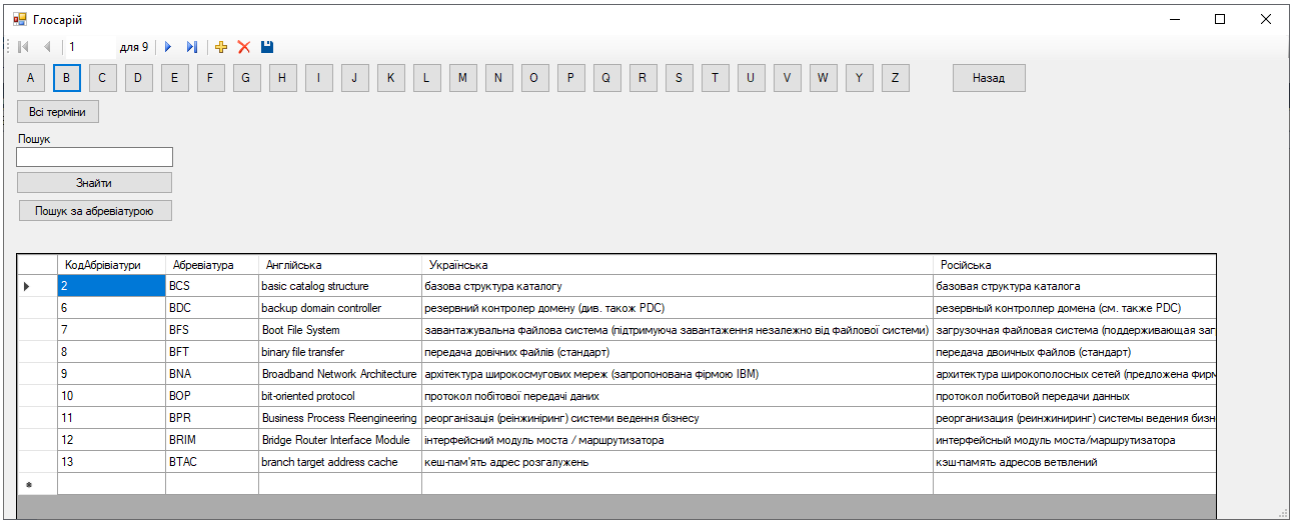


Для пошуку всіх термінів на потрібну літеру, на формі розміщені відповідні літери, при натисканні на потрібну літеру виконується фільтр і виводяться терміни на цю літеру.

Щоб фільтр працював, до налаштувань кнопки додаю такий код, фільтрація буде відбуватися по коду літери:

```
private void button1_Click(object sender, EventArgs e)
{
    this.глосарійBindingSource.Filter = "(КодБукви='1')";
}
```

Приклад того, як працює фільтр:



2.4 Створення інтерфейсу користувача

Спочатку необхідно побудувати логічно–фізичну модель даних за допомогою CASE–засобу ERWin. Фізична модель наведена в додатку А. У MS SQL Server створюємо порожню базу даних. На основі створеної моделі генеруємо базу даних з ERWin у MS SQL Server (Tools – ForwardEngineer/SchemaGeneration). Натискаємо кнопку Generate і відбувається генерація структури БД на основі створеного SQL коду. Діалог зв'язку з БД і виконання SQL коду відбувається в результаті натиснення кнопки Connect. У середовищі MS SQL Server отримуємо згенеровану базу даних.

– Створили новий проект у середовищі Microsoft VisualStudio.

– У діалоговому вікні Створення проекту обрали Windows ApplicationForm, натиснули кнопку ОК.

– Для забезпечення підключення БД, як джерела даних, до проекту у меню «Дані» обрати команду «Додати нове джерело даних».

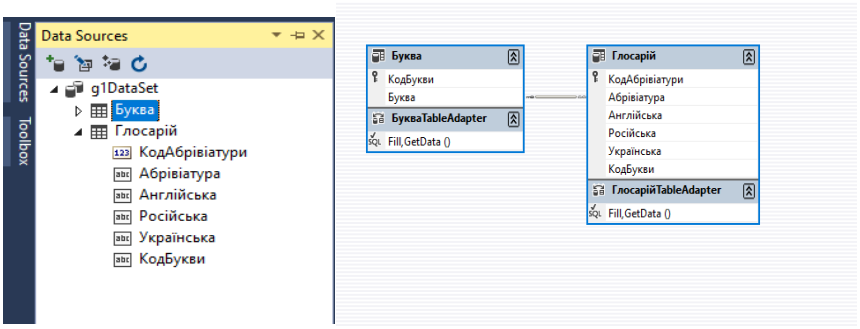
– У майстрі настройки джерела даних обрали пункт «База даних» і натиснути кнопку Далі та обрали «Нове підключення»

– Ввели ім'я серверу, натиснути кнопку Огляд і обрали потрібну базу даних.

– На сторінці «Вибір об'єктів бази даних» розгорнули вузол «Таблиці».

– Встановили прапорці для потрібних таблиць і натиснути кнопку «Готово».

Від тепер у середовищі Microsoft VisualStudio 2017 ми маємо можливість переглянути структуру нашої БД за допомогою внутрішніх засобів, а саме через конструктор набору даних. (див. Додаток А).



Для створення нових форм, додали до проекту «Пуста форма Windows» на першій формі створили кнопки, за допомогою процедур обробки натискання на відповідні кнопки ми маємо можливість переходити на інші форми.

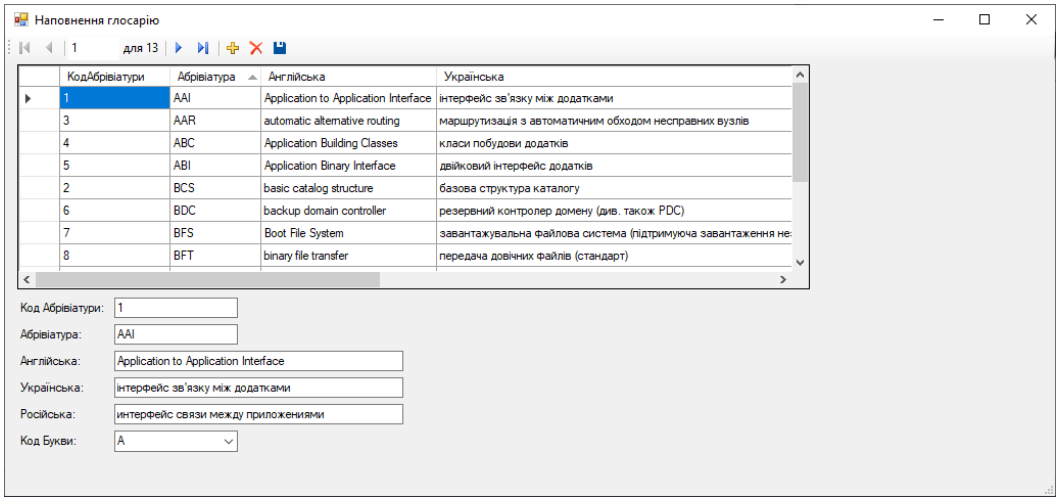
```
private void button4_Click(object sender, EventArgs e)
{

    Hide();
    Form4 N = new Form4();
    N.ShowDialog();

}
```

Аналогічний код прописано для кнопок, що мають відкривати форми.

Для таблиці «Глосарій» створена форма «Наповнення глосарію» для введення, редагування та видалення записів.

```
namespace GLO2
{
    public partial class Form3 : Form
    {
        public Form3()
        {
            InitializeComponent();
        }

        private void глосарійBindingNavigatorSaveItem_Click(object sender, EventArgs e)
        {
            this.Validate();
            this.глосарійBindingSource.EndEdit();
            this.tableAdapterManager.UpdateAll(this.g1DataSet);

        }

        private void Form3_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 'g1DataSet.Буква' table. You can move, or remove it, as needed.
            this.буквaTableAdapter.Fill(this.g1DataSet.Буква);
            // TODO: This line of code loads data into the 'g1DataSet.Глосарій' table. You can move, or remove it, as needed.
            this.глосарійTableAdapter.Fill(this.g1DataSet.Глосарій);

        }
    }
}
```

Вигляд форми «Наповнення глосарію»



Також для взаємодії з джерелом даних використовуються збережені процедури, написані на Transact-SQL - це діалект мови SQL, розроблений компанією Microsoft для використання в СУБД Microsoft SQL Server.

Використання збережених процедур дає кілька очевидних переваг:

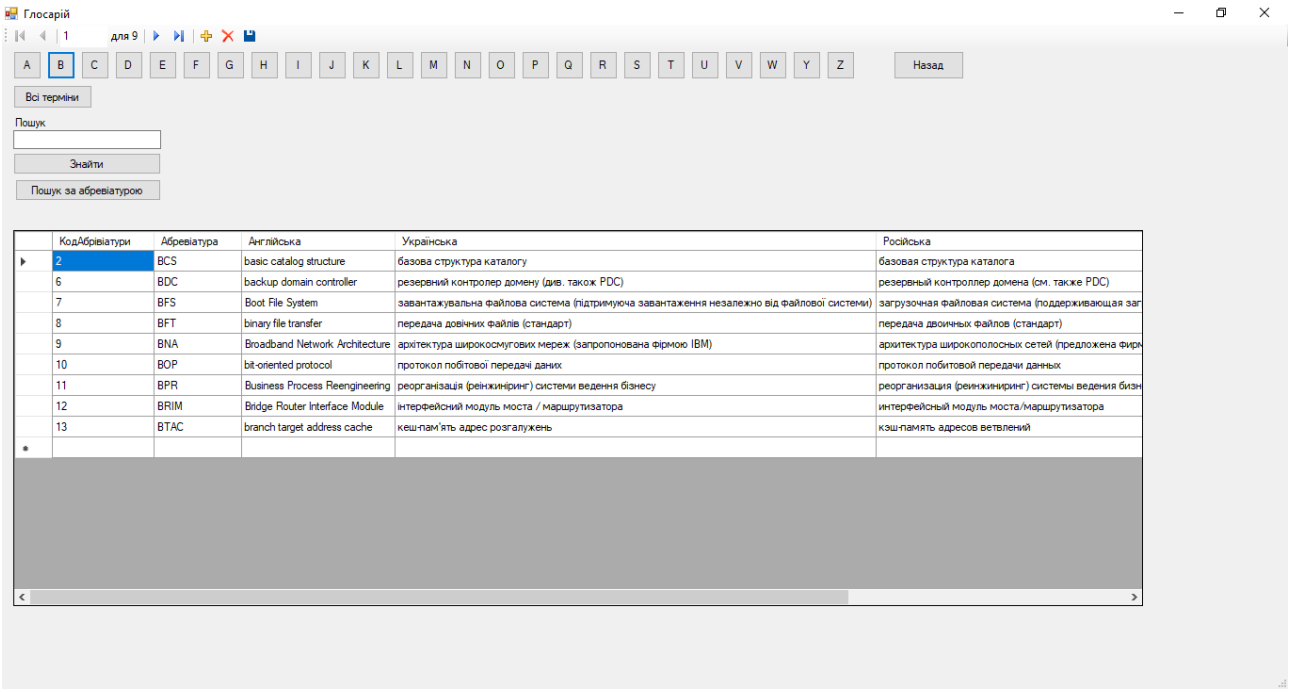
- сценарії виконання збережених процедур кешуються на сервері, що дає відчутний приріст в швидкості при повторному виклику процедур;

- додатковий рівень абстракції дає можливість змінити логіку роботи збереженої процедури без необхідності вносити зміни в додаток (але при цьому не можна змінювати сигнатуру збереженої процедури);

- частина бізнес-логіки додатка виконується на сервері;

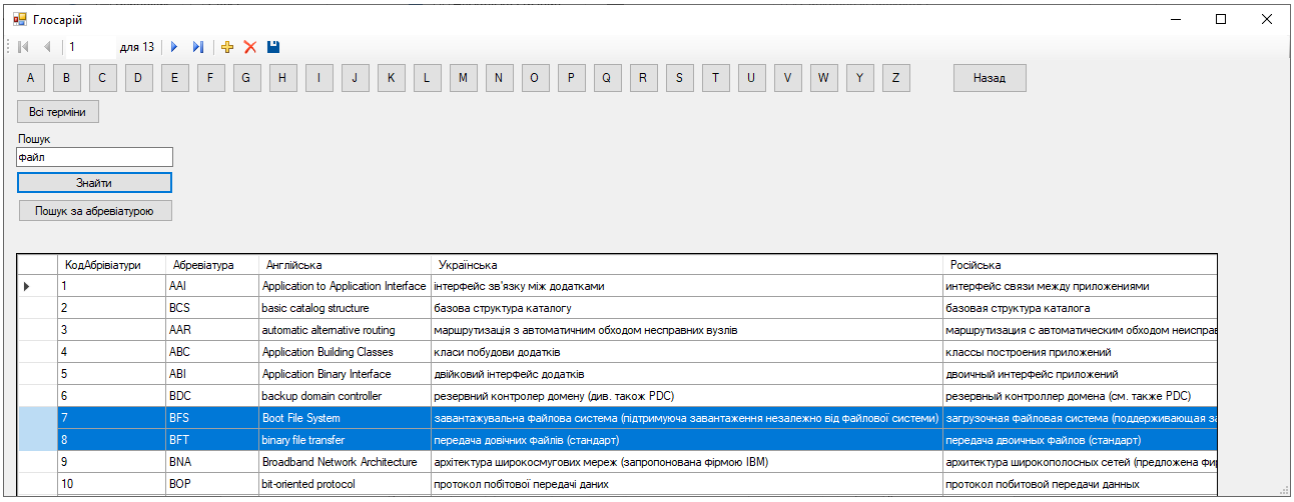
- процедури, що можуть повертати не тільки один набір результатів або, простіше кажучи, таблицю, але і значення вихідних параметрів і навіть кілька наборів результатів (кілька таблиць) за один виклик.

Для перегляду термінів на певну букву достатньо натиснути на кнопку з відповідною буквою і отримаєте всі терміни на цю букву.



КодАбрівіатури	Абревіатура	Англійська	Українська	Російська
2	BCS	basic catalog structure	базова структура каталогу	базовая структура каталога
6	BDC	backup domain controller	резервний контролер домену (див. також PDC)	резервный контроллер домена (см. также PDC)
7	BFS	Boot File System	завантажувальна файлова система (підтримуюча завантаження незалежно від файлової системи)	загрузочная файловая система (поддерживающая заг
8	BFT	binary file transfer	передача довічних файлів (стандарт)	передача двоичных файлов (стандарт)
9	BNA	Broadband Network Architecture	архітектура широкосмугових мереж (запропонована фірмою IBM)	архитектура широкополосных сетей (предложена фирм
10	BOP	bit-oriented protocol	протокол побітової передачі даних	протокол побитовой передачи данных
11	BPR	Business Process Reengineering	реорганізація (реінжиніринг) системи ведення бізнесу	реорганизация (реинжиниринг) системы ведения бизн
12	BRIM	Bridge Router Interface Module	інтерфейсний модуль моста / маршрутизатора	интерфейсный модуль моста/маршрутизатора
13	BTAC	branch target address cache	кеш-пам'ять адрес розгалужень	кэш-память адресов ветвлений

Для пошуку потрібного терміну, потрібно прописати у TexBox, що шукаємо, натиснути на кнопку «Знайти» і отримати результат



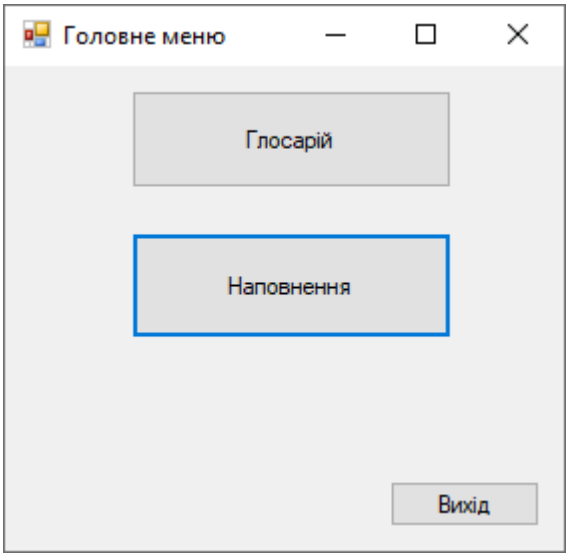
КодАбрівіатури	Абревіатура	Англійська	Українська	Російська
1	AAI	Application to Application Interface	інтерфейс зв'язку між додатками	интерфейс связи между приложениями
2	BCS	basic catalog structure	базова структура каталогу	базовая структура каталога
3	AAR	automatic alternative routing	маршрутизація з автоматичним обходом несправних вузлів	маршрутизация с автоматическим обходом неиспра
4	ABC	Application Building Classes	класи побудови додатків	классы построения приложений
5	ABI	Application Binary Interface	двійковий інтерфейс додатків	двоичный интерфейс приложений
6	BDC	backup domain controller	резервний контролер домену (див. також PDC)	резервный контроллер домена (см. также PDC)
7	BFS	Boot File System	завантажувальна файлова система (підтримуюча завантаження незалежно від файлової системи)	загрузочная файловая система (поддерживающая з
8	BFT	binary file transfer	передача довічних файлів (стандарт)	передача двоичных файлов (стандарт)
9	BNA	Broadband Network Architecture	архітектура широкосмугових мереж (запропонована фірмою IBM)	архитектура широкополосных сетей (предложена фи
10	BOP	bit-oriented protocol	протокол побітової передачі даних	протокол побитовой передачи данных

Код, який реалізує цей пошук:

```
private void button27_Click(object sender, EventArgs e)
{

    for (int i = 0; i < глосарійDataGridView.RowCount; i++)
    {
        глосарійDataGridView.Rows[i].Selected = false;
        for (int j = 0; j < глосарійDataGridView.ColumnCount; j++)
            if (глосарійDataGridView.Rows[i].Cells[j].Value != null)
                if (глосарійDataGridView.Rows[i].Cells[j].Value.ToString().Contains(textBox2.Text))
                {
                    глосарійDataGridView.Rows[i].Selected = true;
                    break;
                }
    }


}
```

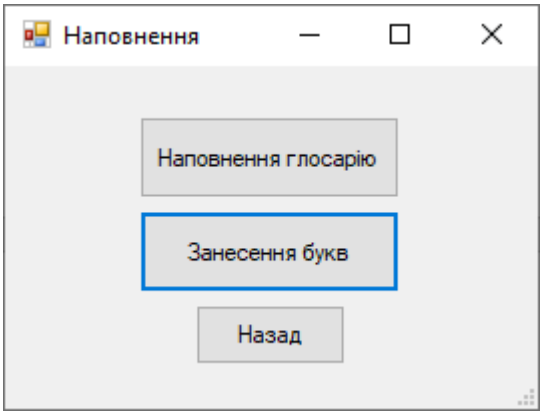
2.5 Інструкція користувача

Для початку роботи програми потрібно запустити додаток. Одразу після запуску користувач побачить інтуїтивно зрозумілий інтерфейс де він може знайти усі посилання та усі доступні функції додатка.



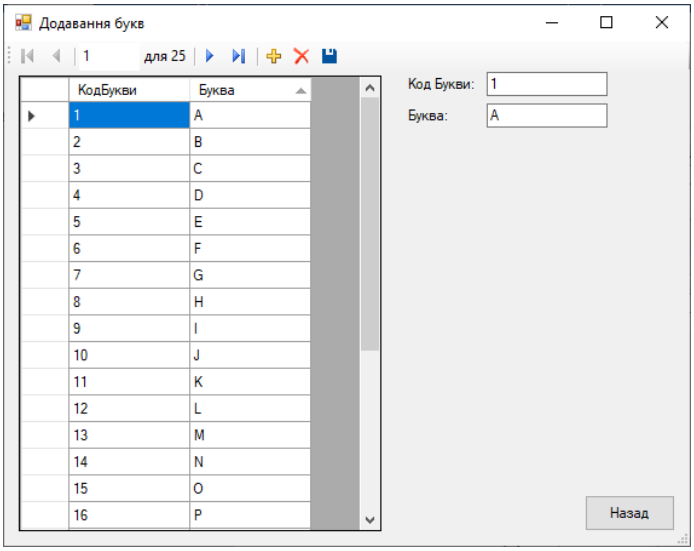
Пункт меню містить такі форми «Глосарій» та «Наповнення». Форма «Наповнення» призначена для адміністраторів, які будуть наповнювати систему, редагувати та видаляти дані у ній.

Форма «Наповнення» включає в себе такі пукти: Наповнення глосарію, Занесення букв та відповідно Назад, для повернення до Головної форми.

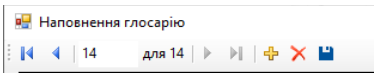


При виборі пункту Занесення букв відкриється форма «Додавання букв», у якій достатньо буде один раз ввести усі букви, в даному випадку латинського алфавіту, для подальшого використання і зручного пошуку термінів за буквами.

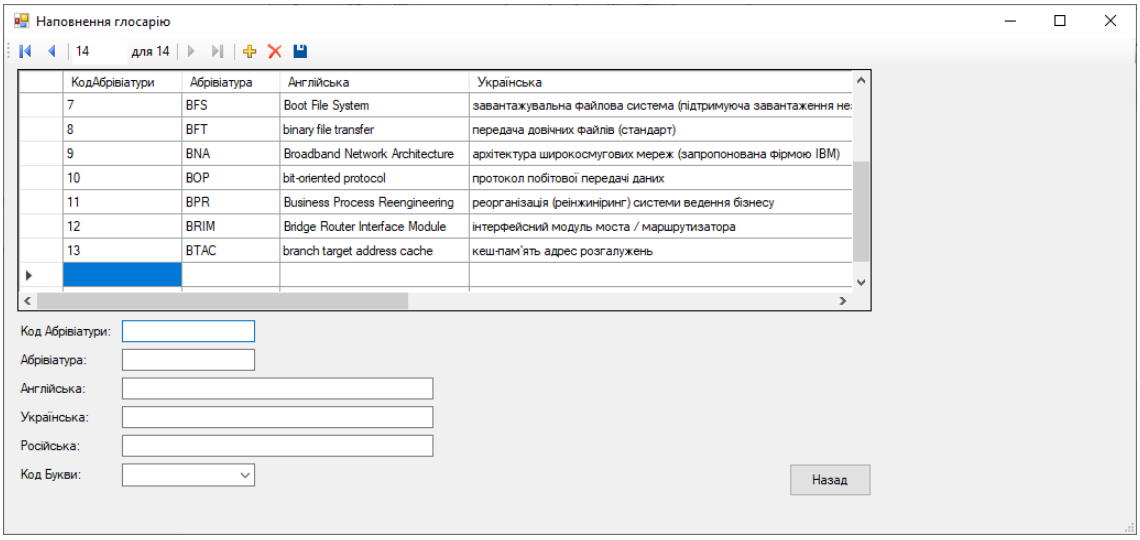
Форма має такий вигляд:



При виборі пункту Наповнення глосарію відповідно відкриється форма «Наповнення глосарію». У цій формі є можливість додавати редагувати та видаляти записи. Для додавання нового запису потрібно натиснути на Плюс у верхньому меню форми, заповнити усі поля та зберегти запис. Для редагуваня обрти потрібний запис, провести у ньому зміни та зберегти його, для видалення обрати потрібний запис та натиснути Хрестик у верхньому меню форми.



Загальний вигляд форма «Наповнення глосарію» має такий вигляд:

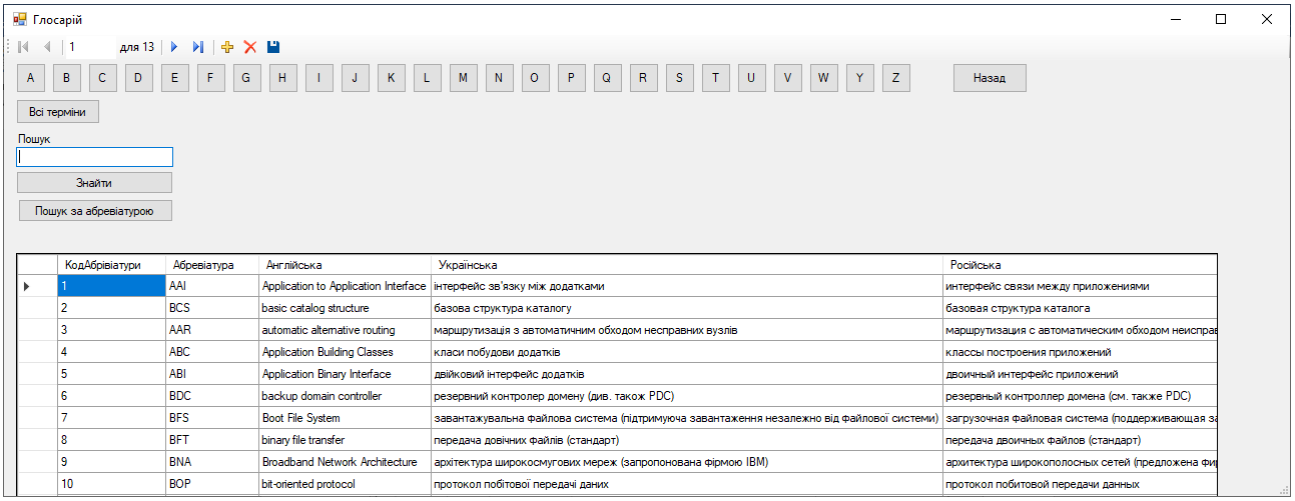


Для користувачів з метою навчання, ознайомлення на Головній формі є пункт Глосарій, який відповідно відкриває форму «Глосарій» . На цій формі розміщена база усіх термінів, є можливість перегляду термінів на потрібну літеру, та пошук потрібного терміну.

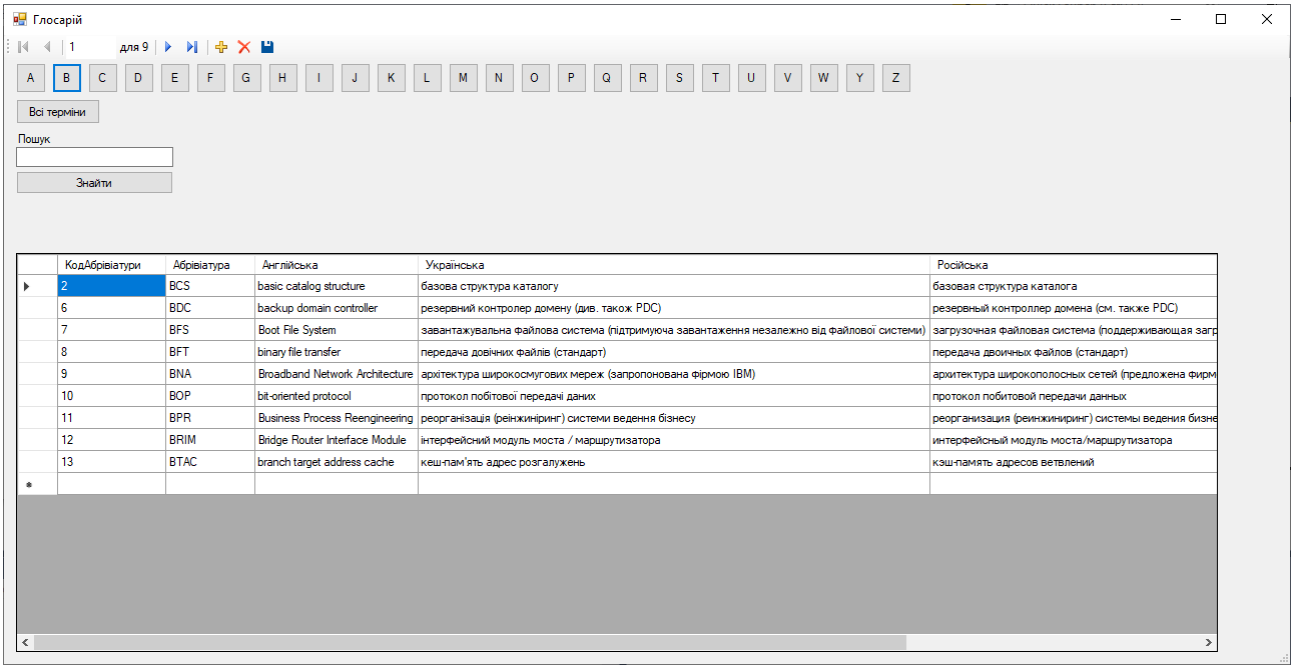
Для перегляду термінів на певну букву достатньо натиснути на кнопку з відповідною буквою і отримаєте всі терміни на цю букву.

Для пошуку потрібного терміну, потрібно прописати у TexBox, що шукаємо, натиснути на кнопку «Знайти» і отримати результат

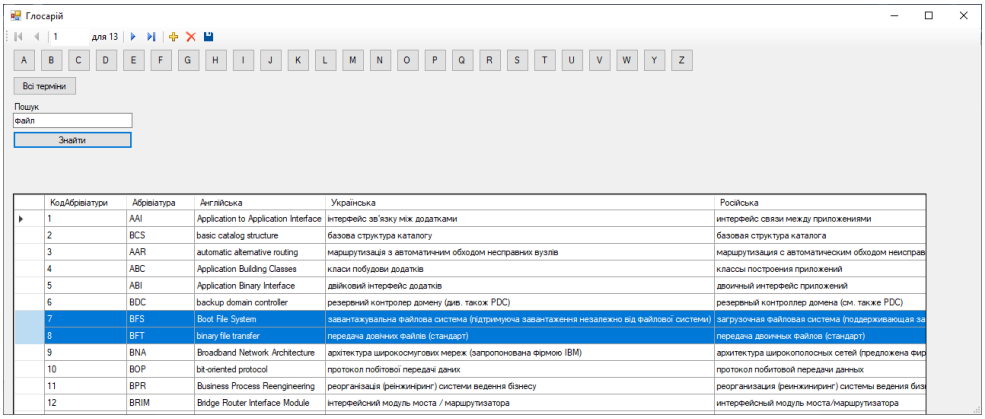
Форма Глосарій має такий вигляд:



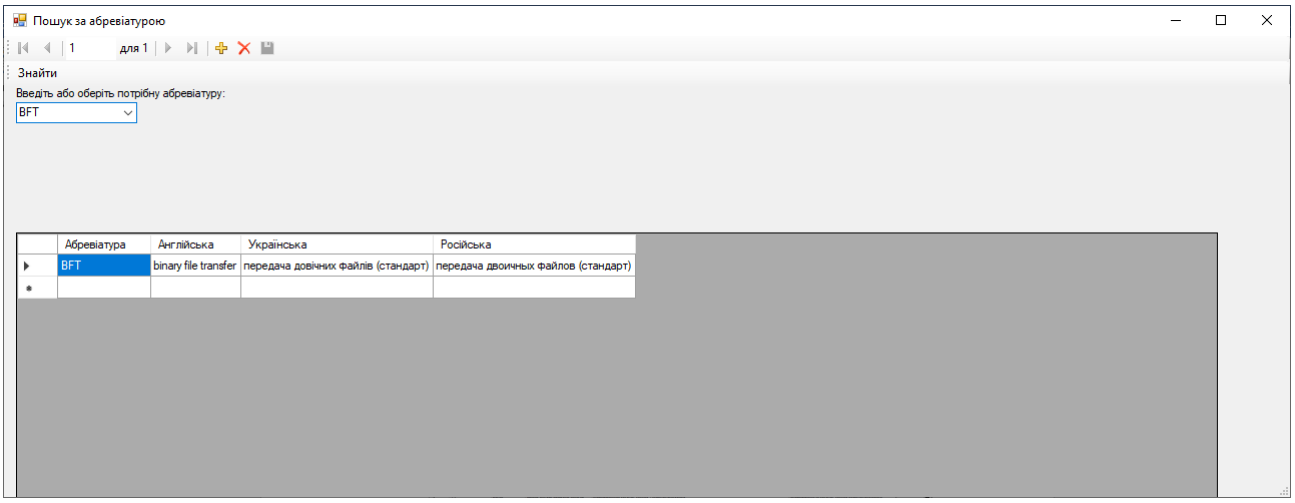
Терміни на потрібну букву:



Пошук потрібних термінів:



Для пошуку за певною абревіатурою потрібно натиснути на кнопку «Пошук за абревіатурою», відкриється відповідна форма, далі потрібно у відповідному полі ввести або обрати потрібну абревіатуру та натиснути кнопку «Знайти»:



РОЗДІЛ 3. ОХОРОНА ПРАЦІ І ТЕХНІКА БЕЗПЕКИ

3.1 Загальні вимоги

Робоче місце - це частина простору, в якому працівник здійснює трудову діяльність, і проводить велику частину робочого часу. Робоче місце, добре пристосоване до трудової діяльності інженера, правильно і доцільно організоване, у відношенні простору, форми, розміру забезпечує йому зручне положення при роботі і високу продуктивність праці при найменшому фізичному і психічному напрузі.

При правильній організації робочого місця продуктивність праці робітника зростає до 20 відсотків.

Згідно ГОСТ 12.2.032-78 конструкція робочого місця і взаємне розташування всіх його елементів повинне відповідати антропометричним, фізичним і психологічним вимогам. Велике значення має також характер роботи. Зокрема, при організації робочого місця програміста повинні бути дотримані наступні основні умови:

- оптимальне розміщення устаткування, що входить до складу робочого місця;

- достатній робочий простір, що дозволяє здійснювати всі необхідні рухи і переміщення;

- необхідно природне і штучне освітлення для виконання поставлених завдань;

- рівень акустичного шуму не повинен перевищувати допустимого значення.

Головними елементами робочого місця програміста є письмовий стіл і крісло. Основним робочим положенням є положення сидячи. Робоче місце для виконання робіт в положенні сидячи організується відповідно до ГОСТ 12.2.032-78.

Робоче сидіння (стілець офісний) оснащене такими основними елементами - це сидіння, спинка та стаціонарні підлокітники, також крісло є поворотним, тобто таке, що регулюються за висотою, кутом нахилу сидіння та спинки, за відстанню спинки до переднього краю сидіння, висотою підлокітників.

Робоча поза сидячи викликає мінімальне стомлення програміста. Раціональне планування робочого місця передбачає чіткий порядок і сталість розміщення предметів, засобів праці і документації. Те, що потрібно для виконання робіт частіше, розташоване в зоні легкої досяжності робочого простору.

3.2 Формування системи стандартів безпеки праці

На основі вимог нормативно-технічної та експлуатаційної документації заводу-виробника ПК, чинних санітарних норм, санітарних норм і правил, правил у сфері охорони праці, директив Ради Європейського союзу 90/270/ЄЕС, 89/391/ЄЕС, 89/654/ЄЕС, 89/655/ЄЕС, стандартів ISO, MPRII власник розробляє систему стандартів безпеки праці (ССБП) на виробництві (установі) головною складовою якої є атестація робочих місць.

Відповідно вимог діючого законодавства в галузі охорони праці обов'язковими нормативними документами, що розробляє власник є:

- правила внутрішнього трудового розпорядку підприємства;
- колективний договір та Угода з охорони праці підприємства;
- визначення професійних груп згідно з діючим класифікатором професій (ДК 003-10);
- положення про проведення оперативного контролю у виробничих підрозділах;

- положення про навчання, інструктаж і перевірку знань працівників з питань охорони праці;
- інструкції з охорони праці для працюючих за професіями і видами робіт;
- загально об’єктові інструкції та заходи з пожежної безпеки;
- наказ про порядок атестації робочих місць щодо їх відповідності нормативним актам з охорони праці;
- список працівників, які підлягають періодичним медичним оглядам;
- наказ про організацію безкоштовної видачі молока або інших рівноцінних харчових продуктів працівникам підприємства, які працюють у шкідливих умовах.

3.3 Характеристика умов праці

Вимоги охорони праці перед початком роботи

Перед включенням на робочому місці обладнання працівник повинен:

- Оглянути і привести в порядок робоче місце, прибрати всі сторонні предмети, які можуть відволікати увагу і ускладнювати роботу.
- Закріпити стіл, стільця, підставки під ноги, кут нахилу екрану монітора, положення клавіатури з метою виключення незручних поз і тривалих напруг тіла. Особливо звернути увагу на те, що дисплей повинен знаходитися на відстані не менше 50 см від очей (оптимально 60-70 см).
- Перевірити правильність розташування обладнання.
- Кабелі електроживлення, подовжувачі, мережеві фільтри повинні знаходитися з тильної сторони робочого місця, мережеві фільтри не повинні лежати на підлозі.

• Переконатися у відсутності засвічень, віддзеркалень і відблисків на екрані монітора.

• Переконатися в тому, що на пристроях ПК (системний блок, монітор, клавіатура) і на столі не розташовуються ємності з рідиною, сипучими матеріалами (чай, кава, сік, вода та ін.).

• Включити електроживлення та переконатися в правильному виконанні процедури завантаження устаткування, правильних налаштуваннях.

• При виявленні несправностей повідомити про це адміністратору і до їх усунення до роботи не приступати.

Вимоги охорони праці під час роботи

Протягом всього часу роботи із засобами комп'ютерної та оргтехніки робітник повинен:

- тримати в порядку і чистоті робоче місце;

- стежити за тим, щоб вентиляційні отвори пристроїв нічим не були закриті;

- виконувати вимоги інструкції з експлуатації обладнання;

- дотримуватися, встановлені розкладом, трудовим розпорядком регламентовані перерви в роботі, навчальному процесі, виконувати рекомендовані фізичні вправи.

Працівнику забороняється під час роботи:

- відключати і підключати кабелі інтерфейсу периферійних пристроїв;

- класти на пристрої засобів комп'ютерної та оргтехніки паперу, папки та інші сторонні предмети;

- торкатися до задньої панелі системного блоку (процесора) при включеному живленні;

- відключати електроживлення під час виконання програми, процесу;

- допускати потрапляння вологи, бруду, сипучих речовин на пристрої засобів комп'ютерної та оргтехніки;

- проводити самостійно розкриття і ремонт обладнання;

- проводити самостійно розкриття і заправку картриджів принтерів або копірів;

- працювати зі знятими кожухами пристроїв комп'ютерної та оргтехніки;

- розташовуватися при роботі на відстані менше 50 см від екрану монітора.

При роботі з текстами на папері, листи треба розташовувати якомога ближче до екрану, щоб уникнути частих рухів головою і очима при перекладі погляду.

Робочі столи слід розміщувати таким чином, щоб монітор були орієнтованим бічною стороною, щоб природне світло падало переважно ліворуч також, щоб освітлення не повинно створювати відблисків на поверхні екрану.

Тривалість роботи на ПК без регламентованих перерв не повинна перевищувати 1 години. Під час регламентованого перерви з метою зниження нервово-емоційного напруження, стомлення зорового апарату, необхідно виконувати комплекси фізичних вправ.

Вимоги охорони праці в аварійних ситуаціях

• Про всі несправності в роботі устаткування і аварійних ситуаціях повідомляти безпосередньо експерту.

• При виявленні обриву проводів живлення або порушення цілісності їх ізоляції, несправності заземлення та інших пошкоджень електрообладнання,

появи запаху гару, сторонніх звуків в роботі обладнання і тестових сигналів, негайно припинити роботу і відключити електронні прилади.

- При ураженні користувача електричним струмом вжити заходів щодо його звільнення від дії струму шляхом відключення електроживлення і до прибуття лікаря надати потерпілому першу медичну допомогу.

- У разі загоряння обладнання відключити, повідомити адміністратору, зателефонувати в пожежну охорону, після чого приступити до гасіння пожежі наявними засобами.

ВИСНОВКИ

В даній дипломній роботі було створено електронний глосарій абревіатур у сфері інформаційних технологій. Для цього було використано засоби CASE-технології: CA ERwin Data Modeler. За допомогою ERwin Data Modeler – логічна і фізична моделі бази даних. Відповідно до них було згенеровано базу даних у середовищі СУБД Microsoft SQL Server 2018, а за допомогою IDE Microsoft Visual Studio 2017 – розроблено інтерфейсний додаток користувача.

Створена система забезпечує швидке та просте наповнення бази абревіатурами, та зручний перегляд та пошук потрібних абревіатур.

Впровадження цієї системи дозволить швидко та зручно знаходити потрібні абревіатури у сфері IT та IC. Система може працювати оф-лайн, тому не потрібно буде підключення до мережі Інтернет. Є можливість знайти терміни на відповідну літеру, пошук за обраною абревіатурою, та реалізовано прости повнотекстовий пошук, який дає змогу, шукати абревіатури по всіх рядках та стовбцях бази даних системи, по цілому слові, по частині слова або за декількома словами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. НПАОП 0.00-1.28-10 " Правила охорони праці під час експлуатації електронно-обчислювальних машин" – 8 ст.
2. [4]. В. Г. Грибан, О. В. Негодченко «Охорона праці» – 162 ст.
3. [5]. К.Н. Ткачук, Л.Д. Третякова «Охорона праці та промислова безпека» - 559 ст.
4. Маклаков С.В. BPwin и ERwin. CASE-средства разработки информационных систем. - М.: ДИАЛОГ-МИФИ, 1999 — 256 с.
5. М’якшило О. М. Моделювання баз даних засобами CASE-технології ERWin: Конспект лекцій з дисципліни «Структурне моделювання систем» для студ. спец. 6.080400 напряму «Комп’ютерні науки» всіх форм навчання. – К.: НУХТ, 2008. – 60с.
6. Бідюк П.І., Коршевнюк Л.О. Проектування комп’ютерних інформаційних систем підтримки прийняття рішень: Навчальний посібник. — Київ: ННК «ІПСУ» НТУУ «КПИ», 2010. —340с.
7. Іцік Б. Г. Microsoft SQL Server 2008. Основы T-SQL / Б. Г. Іцік., 2009. – 430 с. – (БХВ-Петербург).
8. Маккі А. Введение в NET 4.0 и Visual Studio 2010 для профессионалов / Алекс Макки., 2012. – 416 с.
9. Ріхтер Д. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Д. Рихтер., 2017. – 896 с. – (4-е изд.).
10. Хейлсберг А. Язык программирования C#. Классика Computers Science. / А. Хейлсберг, М. Торгерсен, С. Вилтамут., 2011. – 784 с. – ("Питер").
11. Методичні вказівки до виконання кваліфікаційної бакалаврської роботи для студентів за напрямом підготовки 6.050101 “Комп‘ютерні науки ” денної та заочної форм навчання / Уклад.: В.В. Самсонов, Л.Ю. Маноха, Т.М. Горлова, Л.Г. Загоровська, О.М. М’якшило, О.А Хлобистова. – К.: НУХТ, 2011. — 15с.

ДОДАТОК А. Схеми бази даних

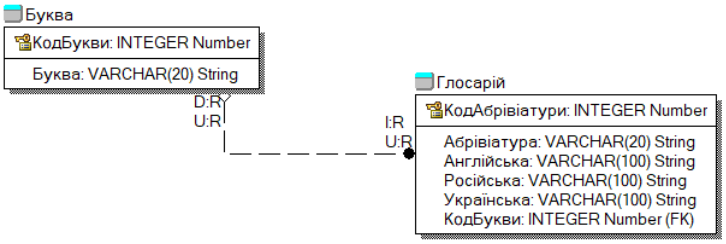


Рисунок 1. Логічна модель бази даних

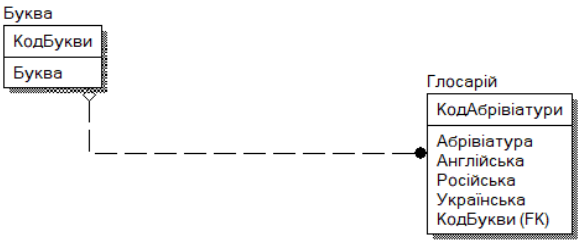


Рисунок 2. Фізична модель бази даних

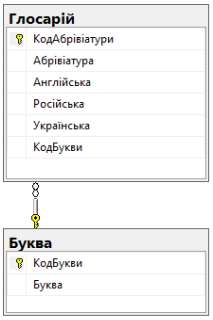


Рисунок 3. Структура бази даних у середовищі MS SQL Server

ДОДАТОК Б. Програмний код

```
for (int i = 0; i < глосарійDataGridView.RowCount; i++)
{
    глосарійDataGridView.Rows[i].Selected = false;
    for (int j = 0; j < глосарійDataGridView.ColumnCount; j++)
        if (глосарійDataGridView.Rows[i].Cells[j].Value != null)
            if (глосарійDataGridView.Rows[i].Cells[j].Value.ToString().Contains(textBox2.Text))
            {
                глосарійDataGridView.Rows[i].Selected = true;
                break;
            }
}
```

Рисунок 1. Код реалізації повнотекстового пошуку.

```
SELECT    Глосарій.Абрівіатура, Глосарій.Англійська, Глосарій.Українська, Глосарій.Російська
FROM      Буква INNER JOIN
            Глосарій ON Буква.КодБукви = Глосарій.КодБукви
WHERE     (Глосарій.Абрівіатура LIKE RTRIM(@name) + '%')
```

Рисунок 2. Код SQL запиту, для пошуку за абревіатурою.

```
private void button1_Click(object sender, EventArgs e)
{
    this.глосарійBindingSource.Filter = "(КодБукви='1')";
}
```

Рисунок 3. Код для відображення абревіатур на певну літеру.

```
namespace GLO2
{
    public partial class Form3 : Form
    {
        public Form3()
        {
            InitializeComponent();
        }

        private void глосарійBindingNavigatorSaveItem_Click(object sender, EventArgs e)
        {
            this.Validate();
            this.глосарійBindingSource.EndEdit();
            this.tableAdapterManager.UpdateAll(this.g1DataSet);

        }

        private void Form3_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 'g1DataSet.Буква' table. You can move, or remove it, as needed.
            this.букваTableAdapter.Fill(this.g1DataSet.Буква);
            // TODO: This line of code loads data into the 'g1DataSet.Глосарій' table. You can move, or remove it, as needed.
            this.глосарійTableAdapter.Fill(this.g1DataSet.Глосарій);

        }
    }
}
```

Рисунок 4. Приклад коду для формування одної з форм системи.