

Акоп'яни Н. М.,*кандидат філологічних наук, доцент,
професор кафедри ділової іноземної мови і перекладу
Національного технічного університету «Харківський політехнічний інститут»***Котов М. В.,***кандидат філологічних наук,
доцент кафедри ділової іноземної мови і перекладу
Національного технічного університету «Харківський політехнічний інститут»*

МЕТАФОРИЗАЦІЯ АНГЛОМОВНОЇ ЛЕКСИКИ У СФЕРІ ПРОГРАМУВАННЯ ТА ОСОБЛИВОСТІ ЇЇ ПЕРЕКЛАДУ УКРАЇНСЬКОЮ МОВОЮ

Анотація. Стрімкий розвиток інформаційних технологій та програмування впливає на мову та збагачує її спеціалізованими термінами. Одним із джерел збагачення терміносистеми у сфері програмування є загальноуживана лексика.

Статтю присвячено вивченню функціонально-семантичного потенціалу термінів метафоричної природи у галузі програмування, утворених на основі загальноуживаних слів у англійській мові. Окрім того, у роботі також досліджуються особливості перекладу таких термінів українською мовою.

У результаті аналізу метафори у сфері програмування було виділено шістнадцять головних доменів на основі семантичного потенціалу стилістично нейтральних слів, які потім трансформувались у терміни у сфері програмування. Цими базовими доменами є «Релігія», «Кулінарія», «Предмети побуту», «Тварини», «Транспорт», «Людина», «Смерть», «Архітектура», «Природа», «Казки і міфологія», «Медицина», «Спорт», «Військо» і «Література».

Встановлено, що одними з найпродуктивніших джерел терміноутворення у сфері програмування є природні (20%), антропоморфні (13%), кулінарні (13%) та архітектурні (10%) метафори.

На анімалістичні і побутові метафори припадає по 9% від усієї метафоризованої концептосфери програмування. Відзначено, що зооморфічність термінів особливо яскраво проявляється при описі характеристик поведінки учасників процесів програмування та їх стилю роботи. Танатологічні метафоризовані терміни у дослідженому корпусі становлять 7%. Вони, як правило, описують цикл активності чи припинення функціонування продукту, коду чи проекту. До доменів «Казки і міфологія» та «Транспорт» було віднесено по 5% термінів-метафор в сфері програмування. Найменш продуктивними виявились релігійна (3%), медична та спортивна (по 2% кожна), а також військова та літературна (по 1% кожна) метафори.

Своєю чергою, аналіз перекладу англійських термінів метафоричної природи у сфері програмування українською мовою дозволив визначити основні методи перекладу. До таких відносимо дослівний переклад, транскрибування, транслітерування, змішане транскрибування та переклад за допомогою еквівалента у цільовій мові.

Ключові слова: домен, метафора, загальноуживана лексика, терміни у галузі програмування.

Постановка проблеми. Сучасний світ характеризується глибокою інтеграцією інформаційних технологій (ІТ) у повсякденне життя людей. Забезпечуючи ефективну обробку, зберігання та передачу інформації, ІТ стали основою для динамічного розвитку науки, освіти, бізнесу, медицини та багатьох інших сфер. У свою чергу, невід'ємною частиною й центральним елементом ІТ є програмування. Саме завдяки програмуванню як процесу проєктування, написання, тестування та налагодження програмного забезпечення реалізуються інноваційні ідеї та відбувається адаптація ІТ до потреб суспільства.

Як і ІТ, сфера програмування динамічно розвивається: створюються нові мови програмування, вдосконалюються інструментарій та платформи, які спрощують написання та тестування програмного коду, з'являються нові підходи й архітектури для створення безпечних й масштабованих рішень. Такий динамічний розвиток вимагає залучення дедалі більшої кількості людей до процесів розроблення та підтримки програмного продукту.

Популярність інформаційних технологій та збільшення кількості осіб, зайнятих в сфері розроблення та підтримки програмного забезпечення, не можуть не позначитись на мові як у побутовому, так і професійному спілкуванні. З одного боку, у повсякденному мовленні дедалі частіше можна почути терміни, які раніше використовувались лише фахівцями-програмістами. З іншого боку, значна кількість загальноуживаної лексики також стала джерелом для збагачення терміносистеми у галузі програмування. У контексті останнього особливо актуальним вбачаємо вивчення метафоричних термінів через "їхню подвійну природу, яка об'єднує точність та емоційну нейтральність, що притаманні звичайним термінам, з образністю та асоціативністю, які характерні метафорам" [1].

Водночас, глобалізаційні процеси та міжкультурна взаємодія, які характеризують сьогодення, також актуалізують подальше вивчення особливостей перекладу українською метафоричних термінів у сфері програмування з англійської мови, де вони й з'явилися. Це сприятиме розвитку української ІТ-галузі, а також підвищенню комп'ютерної грамотності українських користувачів через впорядкування української програмістської терміносистеми.

Аналіз останніх досліджень і публікацій. У нашому дослідженні ми спираємось на висновки щодо функціонування

метафори таких лінгвістів як Д. Лакофф, М. Джонсон, З. Ковечеш, С. Жаботинська, О. Воробйова, О. Селіванова та інших. Англomовну термінологію у сфері інформаційних та комп'ютерних технологій, а також особливості її перекладу українською мовою досліджували О. Буйвол [2], І. Кучман [3], Л. Кармазіна [1], Р. Синдега [4], І. Фесенко й О. Сивачук [5] та інші. При цьому дослідження метафоричних термінів у сфері програмування у зіставному аспекті в українській та англійській мовах наразі ще не здобуло належної уваги.

Стаття має за мету дослідження англomовних метафоричних термінів у сфері програмування та особливостей їхнього перекладу українською мовою.

Об'єктом дослідження є терміни метафоричної природи у галузі програмування, утворені на основі загальноуживаних слів в англійській мові, та їх еквіваленти українською, а предметом – функціонально-семантичний потенціал таких термінів.

Виклад основного матеріалу. Матеріалом дослідження слугували 200 стилістично нейтральних слів, які трансформувались у терміни у сфері програмування. Терміни були відібрані методом суцільної вибірки з сучасних англomовних ресурсів: словників, наукових та науково-популярних статей, підручників, технічної документації.

Під терміном у галузі програмування розуміємо лексичну одиницю, яка є інструментальною для ефективної комунікативної взаємодії спеціалістів в галузі проектування, написання, тестування та налагодження програмного забезпечення, а також зберігання знань, отриманих в результаті професійної діяльності.

Одним із найпродуктивніших способів творення нових термінів у галузі програмування є метафоризація (надання нового

значення) загальноуживаної стилістично нейтральної лексики. Метафоричний термін “поеднує у собі ознаки терміну – мовного знаку, що позначає наукове поняття спеціальної галузі знань, – та метафори” [6, с. 695]. Метафору розуміємо як “спосіб концептуалізації дійсності шляхом тлумачення сутностей певної сфери людського досвіду в термінах сутностей іншої сфери досвіду” [7, с. 74]. Дія механізму метафори пояснюється в термінах мапування (mapping) [8, с. 71] – когнітивної операції, що полягає у накладанні концептуальних елементів царини джерела на концептуальні елементи царини цілі [9, с. 245], у процесі чого «висвітлюються» ті ознаки осмислюваного явища, які, з погляду носіїв мови, є суттєвими для його розуміння [10, с. 54]. Своєю чергою, концептуальною структурою, яка слугує фоном (контекстом) для виділення (профілювання) концепту, є домен [7, с. 21].

Дослідивши метафору у сфері програмування, ми виділили 16 головних доменів на основі семантичного потенціалу стилістично нейтральних слів (приблизно 200 термінів), які потім трансформувались у терміни у сфері програмування. Такими базовими доменами є: «Релігія», «Кулінарія», «Предмети побуту», «Тварини», «Транспорт», «Людина», «Смерть», «Архітектура», «Природа», «Казки і міфологія», «Медицина», «Спорт», «Військо» і «Література» (Див. рис. 1).

Як показує дослідження, домен «Природа» є одним з найпотужніших джерел терміноутворення у сфері програмування (20%): greenfield project – проєкт, що починається з нуля, без необхідності підтримки або врахування попередніх версій чи обмежень; brownfield project – проєкт, який вимагає роботи з наявними системами або кодом, що потребує оновлення або переписування; low-hanging fruit – легко здійснені завдання або

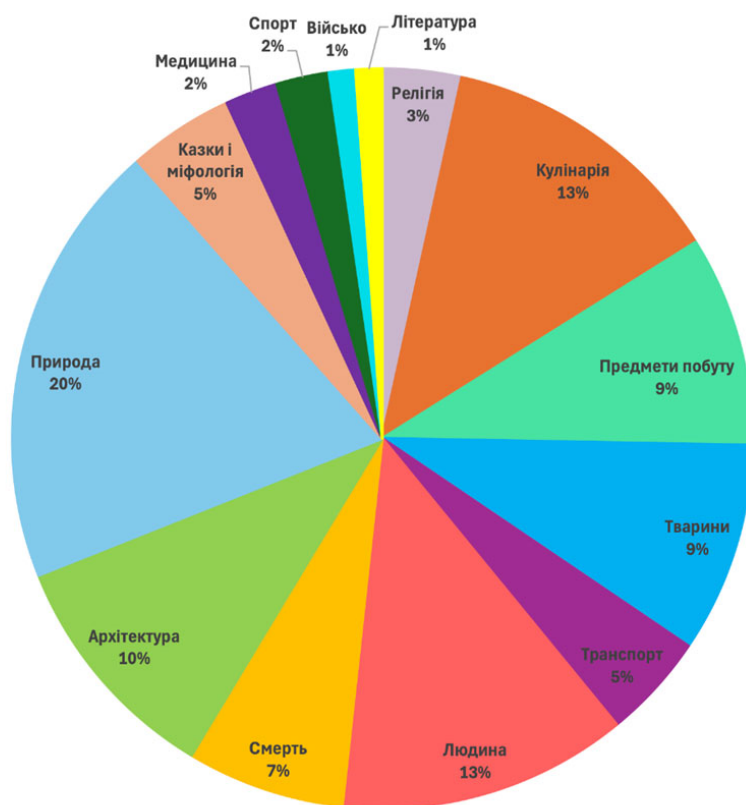


Рис. 1. Домени стилістично нейтральних слів з потенціалом метафоричного терміноутворення в сфері програмування

проблеми, які можна вирішити без особливих зусиль, часто починаючи з них для швидкого результату; *waterfall model* – каскадна модель, модель розробки програмного забезпечення, де процес проєктування розділений на фази, і кожна фаза повинна завершитись перед початком наступної; *tree* (*Tree Data Structure*) – дерево як структура даних, ієрархічна структура даних, яка складається з вузлів (вершин), з'єднаних між собою гілками; *branch* – гілка, нова/окрема версія головного репозиторію; *head branch* – головна гілка, іменоване посилання на коміт, що знаходиться на кінці гілки; *master branch* – основна гілка, гілка за замовчуванням для розробки в системі контролю версій GIT; *root* – корінь, перший вузол дерева даних; *leaves* – листя, кінцеві вузли дерева; *kernel* – ядро, основна частина операційної системи, що керує системними ресурсами та забезпечує взаємодію між апаратним і програмним забезпеченням; *shell* – оболонка, шелл, інтерфейс користувача для доступу до послуг операційної системи, зазвичай через командний рядок; *cloud computing* – хмарні обчислення, наявність обчислювальних ресурсів на вимогу, зокрема зберігання даних і обчислювальної потужності, без необхідності активного управління з боку користувача; *cloud migration* – міграція в хмару, процес перенесення локальної IT-інфраструктури, включаючи бази даних, додатки та інші компоненти, у хмарне середовище; *smoke testing* – димне тестування, попереднє тестування для перевірки роботи основних функцій програмного застосунку.

Іншими потужними джерелами термінотворення у сфері програмування є антропоморфні та кулінарні метафори, на які припадає 13% від загального складу досліджуваних метафор. До антропоморфних метафор в сфері програмування ми відносимо: *singleton* – одинак, шаблон проєктування, коли існує лише один екземпляр класу, який відповідає за певну функціональність; *cowboy coding* – ковбойське програмування, стиль програмування без належного планування, документації або дотримання правил, коли програміст діє як «ковбой», роблячи все на свій розсуд; *greedy algorithm* – жадібний алгоритм, алгоритм, який на кожному етапі вибирає оптимальне (жадібне) рішення, яке здається найкращим на той момент, хоч і не завжди веде до глобально оптимального результату; *handshake* – рукоштовнання, процес встановлення зв'язку або з'єднання між двома системами чи комп'ютерами, подібно до вітання перед початком спілкування; *head* (*branch*) – головна (гілка), іменоване посилання на коміт, що знаходиться на кінці гілки; *master* (*branch*) – основна (гілка), гілка за замовчуванням для розробки в системі контролю версій GIT; *host* – хост, будь-який апаратний пристрій, що дозволяє доступ до мережі за допомогою інтерфейсу користувача, спеціалізованого програмного забезпечення, мережевої адреси чи протоколів; *listener* – слухач, об'єкт, який очікує і реагує на події у програмуванні, орієнтованому на події; *multi-tenant* – мультиорендність, тип архітектури, що використовується у хмарних обчисленнях, яка дозволяє запускати кілька окремих екземплярів програмного забезпечення з одного фізичного сервера; *parent class* – батьківський клас, клас, від якого успадковуються атрибути та функції дочірніми класами; *child class* – дочірній клас, клас, що успадковує всі атрибути й функції батьківського класу.

Кулінарна група термінів-метафор займає таку ж частку як і антропоморфні в усій метафоризованій концептосфері програмування (13%), зокрема її складають такі метафоричні терміни: *spaghetti code* – спагеті-код, погано структурований, заплута-

ний код, де немає чіткої логіки, що ускладнює його розуміння та супровід; *lasagna code* – лазанья-код, багатошаровий код, де логіка розділена на багато рівнів або «шарів», що робить його важким для розуміння, але дещо краще організованим, ніж «спагеті-код»; *swiss cheese model* – модель швейцарського сиру, модель безпеки чи ризиків, яка показує, як невеликі дірки в різних шарах системи можуть спричинити серйозні проблеми, коли ці дірки виявляються в одному місці; *cookie* – куки, невеликий файл, збережений на комп'ютері користувача для зберігання інформації, такої як налаштування користувача чи сесії; *honeypot* – медовий горщик, спеціально створене вразливе середовище для того, щоб привернути увагу хакерів і відвернути їх від справжніх систем; *breadcrumbs* – крихти, сліди навігації, які допомагають користувачеві зрозуміти, де він знаходиться в ієрархії сторінок чи меню програми; *hot potato* – проблема або задача, якою ніхто не хоче займатися і яку «перекидають» між членами команди; *onion architecture* – цибулинна архітектура, структура програмного забезпечення, де код розподілений на шари, подібно до шарів цибулі, щоб зробити систему гнучкішою та легшою для супроводу; *dog fooding* – користування своїм продуктом, процес, коли компанія сама використовує свій продукт, щоб краще зрозуміти його сильні та слабкі сторони; *salt* – сіль, випадкові дані, додані до паролів перед хешуванням для підвищення їхньої безпеки; *pepper* – перець, додаткове випадкове значення, яке додається до хешів паролів, зазвичай зберігається окремо від самого хешу.

Від домену «Архітектура» (10%) утворилися наступні метафоризовані терміни в сфері програмування: *broken window* – розбите вікно, погана практика або незначна помилка у коді, яка залишена без виправлення і може призвести до зниження якості всього проєкту; *spaghetti architecture* – архітектура-спагеті, погано спроектована архітектура програмного забезпечення, у якій модулі чи компоненти переплетені настільки, що це ускладнює зміни та обслуговування; *laying the groundwork* – закладання основ; *walled garden* – огорожа, замкнута система або середовище, яке обмежує взаємодію з іншими системами або додатками, щоб зберегти контроль; *sandbox* – пісочниця, ізольоване тестове середовище, де код може виконуватися без впливу на основну систему; *gateway* – шлюз, інструмент управління API, що слугує посередником між клієнтом і набором різних бекенд-сервісів; *monolithic architecture* (*monolith*) – монолітна архітектура, єдина, велика обчислювальна система з однією базою коду, яка об'єднує всі бізнес-функції разом; *bridge* – міст, патерн проєктування, що розділяє абстракцію і її реалізацію, дозволяючи їм змінюватися незалежно.

На анімалістичні і побутові метафори припадає 9% на кожну категорію від усієї метафоризованої концептосфери програмування. Зооморфічність термінів особливо яскраво проявляється при описі поведінки учасників процесів програмування, їх стилю роботи: *monkey patching* – мавпяче латання, процес внесення змін або додавання функціоналу до вже існуючого коду або бібліотеки, іноді нестандартними методами; *rubber duck debugging* – налагодження з гумовою качечкою, метод налагодження, коли програміст пояснює свій код або проблему простому об'єкту (наприклад, гумовій качечці), щоб знайти помилку або зрозуміти логіку коду; *canary release* – канарка, метод тестування нової версії програмного забезпечення шляхом поступового її випуску для обмеженої кількості користувачів, щоб перевірити, чи не з'являються критичні помилки;

duck typing – качиний тип, принцип програмування, де об'єкт визначається не через його тип, а через поведінку (якщо виглядає і працює, як качка, то це качка); code monkey – мавпочка, що пише код, жартирливий термін для програміста, який просто пише код, не беручи участі в плануванні чи проектуванні, а лише виконує завдання, як «робот»; one-trick pony – поні з одним трюком, образливий термін, який використовується для опису людини, яка має лише одну обмежену здатність або навичку та не може ефективно виконувати будь-які інші завдання; yak shaving – гоління яка, ситуація, коли для досягнення кінцевої мети потрібно виконати серію непрямих і часто нудних задач, термін підкреслює складність і неочевидність такого процесу; gorilla arm – мавпяча рука, ситуація, коли інтерфейс або технологія змушують користувача чи розробника постійно виконувати фізично важку або неприємну роботу, наприклад, тримати руку в піднятій позиції.

Значне місце також посідає побутова семантична група метафоричного терміноутворення в комп'ютерному глосарії (9%): fork – відгалуження, копія проєкту, яку створюють для внесення змін, часто для розвитку нових функцій або для експериментів, не змінюючи основну версію; forking the code – розгалуження коду, створення копії існуючого коду для початку розробки нового напрямку або нової версії програмного забезпечення; pod – под, найменший і найпростіший об'єкт на платформі Kubernetes, який представляє набір контейнерів, що працюють на кластері користувача; container – контейнер, метод пакування і безпечного запуску застосунків у середовищі віртуалізації додатків; black-box testing – тестування методом «чорної скриньки», метод тестування програмного забезпечення, при якому перевіряється робота програми без знання її внутрішньої побудови та схеми роботи, іншими словами, не маючи доступу до коду програми; keys – ключі, унікальний ідентифікатор, який дозволяє іншим програмам автентифікувати користувача, розробника або програму при зверненні до API, щоб підтвердити їхню автентичність.

Танатологічні метафоризовані терміни у програмуванні становлять 7%. Вони описують цикл активності чи припинення функціонування продукту, коду чи проєкту, наприклад: blue screen of death (BSOD) – синій екран смерті, критичний системний збій в операційній системі Windows, коли комп'ютер показує синій екран з повідомленням про помилку і вимагає перезавантаження; dead code – мертвий код, частини коду, які більше не виконуються або не мають впливу на програму, але все ще присутні у вихідному коді; kill switch – вимикач, механізм, що дозволяє швидко відключити програму чи систему в разі серйозних помилок або загроз; kill chain – ланцюг знищення, покроковий процес або стратегія, що використовується для досягнення цілі, наприклад, в атаці на систему, де кожен етап є важливим для успіху; killer feature – вбивча функція, унікальна або дуже потужна функція, яка робить продукт або програмне забезпечення особливо привабливим для користувачів; EOL (End of Life) – закінчення терміну експлуатації, програма досягла кінця свого терміну служби або припинення підтримки (EOS), коли виробник більше не надає оновлень.

До домена «Казки і міфологія» нами було віднесено такі терміни-метафори в сфері програмування (5%): unicorn – надзвичайно рідкісне або ідеальне програмне забезпечення чи технологія, що здається надто хорошим, щоб бути правдою, або стартап, який зможе досягти дуже високої вартості; ghost

variable – примарна змінна, змінна, яка має залишковий ефект у коді або може бути незрозуміло звідки взята, але впливає на роботу програми; zombie process – процес-зомбі, процес, який завершився, але все ще залишається в таблиці процесів, не будучи повністю видаленим; Goldilocks principle – принцип Златовласки, концепція, яка описує ситуацію, коли щось має бути «саме таким, як треба», а не надмірним чи недостатнім, застосовується при налаштуванні систем та продуктів таким чином, щоб вони не були занадто складними або занадто спрощеними, а саме такими, що відповідають потребам користувача. Останній термін походить від казки про Златовласку (Goldilocks), яка шукала середній варіант: не надто гарячу і не надто холодну кашу, не надто м'яке і не надто тверде ліжко, а саме такий варіант, що їй підходив.

На семантичну групу «Транспорт» також припадає 5% термінів-метафор. Зокрема метафоричну природу має вираз bus factor / truck factor, що позначає міру зосередження інформації серед окремих членів проєкту. Bus factor показує кількість розробників команди програмістів, після втрати яких проєкт не може бути продовжений далі. Проєкт міститиме таку інформацію, з якою розробники, що залишилися, не зможуть розібратися. Високий bus factor проєкту означає, що проєкт стійко розвиватиметься, якщо його покине навіть велика кількість програмістів. Іншими словами, низький bus factor – це наявність специфічних знань, якими володіє обмежена кількість розробників команди, заплутаний або малозрозумілий код, використання технології, знаннями про яку володіють лише кілька людей з команди, відсутність документації, конфіденційність тощо. Іншими транспортними термінами-метафорами є: parking lot – паркування, метод залишення ідей або задач на потім, зазвичай до завершення більш важливих справ; helicopter view – погляд з гелікоптера, високий рівень огляду проблеми чи проєкту, коли ви маєте загальне уявлення про всі аспекти, не занурюючись в деталі; pipeline – конвеєр, серія етапів обробки даних, де вихід одного етапу є вхідними даними для наступного.

Особливе місце в метафоричній термінології сфери програмування займає релігійна семантична група (3%). Розглянемо метафоричну природу даних метафор і особливості їх перекладу українською мовою. God object – об'єкт-бог, божественний об'єкт. Це термін із програмування, який означає об'єкт, що має занадто багато відповідальностей або занадто великий вплив на систему. Такий об'єкт порушує принципи вдалого дизайну, зокрема принцип єдиної відповідальності, що робить код важким для розуміння, тестування та підтримки. Такий код досить складно підтримувати, якщо врахувати, що практично вся система залежить лише від нього. Термін «Exodus», що перекладається як вихід, має біблійне коріння і первинно означає «вихід» або «мігрантський шлях». У Біблії «Exodus» позначає вихід єврейського народу з Єгипту. У програмуванні цей термін застосовується метафорично, описуючи великі зміни або важливі етапи, коли система, організація чи команда «виходить» з одного стану в інший. Загалом, це метафора для великого, часто складного переходу або змін у проєкті чи технології. IT evangelist – айті-євангеліст, IT-проповідник/пропагандист, людина, яка займається професійною пропагандою в сфері IT.

Медичні терміни-метафори займають таку ж частку як і спортивні в усій концептосфері програмування (2%). Серед

таких термінів можемо виділити, наприклад, “triage” – триаж, процес оцінки та визначення пріоритетності помилок, багів або завдань у проєкті, процес фільтрації запитів. У деяких випадках, коли є багато запитів або завдань, триаж може бути процесом відбору і визначення, які з них повинні бути оброблені першими, а які можуть чекати. Цей термін походить з медицини, де триаж позначає процес сортування пацієнтів за рівнем терміновості їхнього лікування, але в програмуванні він застосовується до оцінки і вирішення проблем із програмним забезпеченням. Іншим медичним метафоричним терміном в сфері програмування є “code injection” – ін’єкції коду, вид вразливості, з якою з’являється ймовірність запустити на виконання код з метою отримання доступу до ресурсів системи, злову особистих даних або пошкодження програмного забезпечення.

Серед спортивних метафор у сфері програмування можемо виділити: to hit the ground running – почати працювати над проєктом або завданням швидко й ефективно, без втрат часу, взяти старт одразу; race condition – стан гонитви, проблема в багатопотокових програмах, коли кілька потоків намагаються одночасно змінити один ресурс, що може спричинити непередбачувані результати.

Військові і літературні терміни-метафори у сфері програмування є найменш вживаними (1%). До військової метафори можемо віднести, наприклад, “chain of command” – ланцюг командування, командна вертикаль, субординація, ієрархічна структура управління, де кожен має чітке місце та роль у прийнятті рішень. На основі літературного домену було утворено термін “Big Brother” – “Старший Брат”, система спостереження або моніторингу, яка може порушувати конфіденційність або контроль за діяльністю користувачів. Даний образ було створено Джорджем Орвеллом в романі-антиутопії “1984”.

Також аналіз перекладу англійської лексики у сфері програмування українською мовою дозволив визначити основні методи перекладу. Зокрема, до таких відносимо: дослівний переклад (наприклад, god object – божественний об’єкт, spaghetti code – спагетті-код, yak-shaving – гоління яка); транскрибування (cookie – куки); транслітерування (pod – под); змішане транскодування (container – контейнер, triage – триаж); переклад за допомогою еквівалента у цільовій мові (salt – сіль, pepper – перець, keys – ключі тощо).

Висновки. Отже, в результаті нашого дослідження ми прийшли до висновку, що стилістичний прийом метафоризації є потужним і ефективним механізмом терміноутворення англійської лексики в сфері програмування. Серед способів поповнення даної терміносистеми було виділено продуктивний метафоричний потенціал природних, антропоморфних, кулінарних та архітектурних доменів англійської лексики. До найбільш широко використовуваних способів перекладу термінів-метафор у програмуванні нами віднесено: дослівний переклад, транскрибування, транслітерацію, змішане транскодування та переклад за допомогою еквівалента на цільовій мові. Подальшу перспективу досліджень вбачаємо у вивченні метафоризації англійської лексики в інших сферах людської діяльності, зокрема політичній, і аналізі способів її перекладу українською мовою.

Література:

1. Кармазіна Л.Л. Когнітивний підхід до вивчення англійських метафоричних іт-термінів. *Закарпатські філологічні студії*. 2023. № 29(1). С. 133–137.
2. Буйвол О.В. Щодо структурної організації комп’ютерної лексики. *Вісник Національного технічного університету «ХПІ»*. Серія: *Актуальні проблеми розвитку українського суспільства*. 2021. № 1. С. 104–108.
3. Кучман І.М. Переклад англійських термінів у галузі комп’ютерних технологій. *Вісник Житомирського державного університету імені Івана Франка*. 2005. № 23. С. 164–166.
4. Синдега Р.С. Роль терміна у фаховому дискурсі з проблем комп’ютерних наук та інформаційних технологій. *Проблеми семантики слова, речення та тексту*. 2012. № 28. С. 379–384.
5. Фесенко І.М., Сивачук О.М. Англійська комп’ютерна термінологія: структурні особливості та способи творення. *Збірник наукових праць «Нова філологія»*. 2021. № 84. С. 248–254.
6. Бондаренко О. Метафоричні терміни сфери інформаційних технологій: до питання перекладу українською мовою. *Наукові записки [Центральноукраїнського державного педагогічного університету імені Володимира Винниченка]*. Серія : *Філологічні науки*. 2019. Вип. 175. С. 694–698.
7. Матринюк А.П. Словник основних термінів когнітивно-дискурсивної парадигми. Харків : ХНУ імені В.Н. Каразіна, 2011. 196 с.
8. Lakoff G., Johnson M. *Philosophy in the Flesh: The Embodied Mind and its Challenge to Western Thought*. N.Y. : Basic Books, 1999. 602 p.
9. Lakoff G., Johnson M. *Metaphors We Live by*. Chicago : Chicago Univ. Press, 1980. 242 p.
10. Lakoff G. What is a conceptual system? *The nature and ontogenesis of meaning* / Ed. Willis F. Overton, David S. Palermo. New Jersey Hove : Lawrence Erlbaum Associates, 1994. P. 41–86.

Akopiants N., Kotov M. Metaphorization of English programming terminology and its translation into Ukrainian

Summary. The rapid advancement of information technology (IT) and its integral component, programming, significantly impacts language by enriching it with specialized terminology. One key source of this enrichment in programming terminology is general-use vocabulary.

This article explores the functional-semantic potential of metaphorical terms in the sphere of programming, derived from commonly used English words. It also examines the nuances of translating these terms into Ukrainian.

An analysis of metaphors in programming revealed sixteen primary domains, each rooted in the semantic potential of stylistically neutral words that later evolved into programming terms. These key domains include “Religion”, “Cooking”, “Household Items”, “Animals”, “Transport”, “Humans”, “Death”, “Architecture”, “Nature”, “Fairy Tales” and “Mythology”, “Medicine”, “Sports”, “Military”, and “Literature”.

The analysis identified natural (20%), anthropomorphic (13%), culinary (13%), and architectural (10%) metaphors as the most productive sources of terminology formation in programming.

Animalistic and household metaphors each constitute 9% of the total metaphorized conceptual sphere of programming. Notably, zoomorphic terms vividly capture and illustrate the behavioral characteristics, tendencies, and working

styles of individuals engaged in programming processes. Thanatological (death-related) metaphors account for 7%, typically depicting activity cycles or signaling the cessation of a product, code, or project. The domains of “Fairy Tales”, “Mythology” and “Transport” each encompasses 5% of programming metaphors.

The least productive metaphorical categories include religious (3%), medical and sports (2% each), as well as military and literary metaphors (1% each).

Furthermore, an in-depth analysis of how English metaphorical terms in programming are translated into Ukrainian has enabled the identification of key translation strategies. These methods include literal translation, transcription, transliteration, mixed transcoding, and translation using an equivalent in the target language.

Key words: domain, metaphor, commonly used words, programming terms.