

МЕТОДИКА ПОНЯТТЄВОГО АНАЛІЗУ ГАЛУЗЕВИХ ТЕКСТІВ ДЛЯ ФОРМУВАННЯ ПРЕДМЕТНОЇ КОМПЕТЕНТНОСТІ МАЙБУТНІХ ПЕРЕКЛАДАЧІВ

Купрієнко Ю. В.

Національний технічний університет "ХПІ"

Черноватий Л. М.

Харківський національний університет імені В. Н. Каразіна

Стаття присвячена питанням аналізу галузевих текстів для сфери "Комп'ютерна наука" (тема – "Мова комп'ютерного програмування"), а також важливості засвоєння відповідної термінології для формування фахової компетентності перекладача.

Ключові слова: мова комп'ютерного програмування, фахова компетентність перекладача, предметна компетентність, термінологічна одиниця, поняттєва структура.

The article is devoted to the professional-oriented texts analysis for the field of computer science (subject – "Computer programming language"). It also deals with the importance of proper terminology acquisition to form translator's professional competence.

Key words: computer programming language, translator's professional competence, subject competence, terminological unit, concept structure.

Мета навчання майбутніх перекладачів – формування фахової компетентності перекладача (ФКП), яку визначають як належне використання специфічних здібностей згідно з вимогами оточення, тобто цілеспрямована поведінка [7], одним із компонентів якої є екстралінгвістична компетентність, розглядана як та, що охоплює всі знання поза межами лінгвістичних та перекладознавчих [5]. У межах цієї компетентності перебуває предметна компетентність, яку тлумачать як певний діапазон знань, необхідних перекладачеві у відповідній сфері [5].

У межах кожної сфери, що входять до складу предметної компетентності (громадсько-політична, економічна, юридична тощо), існує система понять, на якій ґрунтується ця сфера, а також термінологічні одиниці для їх позначення в кожній мові. Було обрано найвдаліше, на нашу думку, визначення поняття "термін" (або "термінологічна одиниця"), на яке будемо спиратися в подальших роботах, а саме: "термін (від грец. *τέρμα* 'кінець, межа') – це одиниця лексичного рівня (слово або словосполучення), яка позначає певне поняття у відповідній галузі людської діяльності, утворює функціонально-тематичний клас галузевої лексики і є органічним (системним чи позасистемним) елементом термінологічного фонду [1, с. 24]. Таким чином, завданням навчання майбутніх перекладачів є, по-перше, засвоєння згаданих понять, а по-друге, оволодіння відповідною термінологією.

Для розв'язання такого завдання потрібно обрати сфери, які ввійдуть до складу предметної компетентності майбутніх перекладачів (залежно від спрямованості навчання у відповідному навчальному закладі: перекладачі загального спрямування, перекладачі технічних текстів тощо); у межах обраних сфер визначити найважливіші теми; в кожній темі виділити головні поняття та встановити їх ієрархію, знайти двомовні терміни для позначення згаданих понять.

Проблема визначення компонентів предметної компетентності майбутніх перекладачів, виділення поняттєвої структури окремих її тем та добору термінологічних одиниць формулювалася в деяких теоретичних працях [5], а також практично реалізувалася (за винятком виділення поняттєвої структури) в деяких сферах, наприклад, економіка [2], юриспруденція [3], патентна справа тощо [4]. Однак на рівні кандидатських дисертацій це питання досі не вивчалася, в чому й полягає **актуальність** нашого дослідження.

Його **об'єктом** є поняттєвий аналіз галузевих текстів з метою відбору головних понять, встановлення їх ієрархії та термінологічних одиниць для їх позначення, а **предметом** – розроблення поняттєвих схем для сфери “Комп’ютерні науки”, яка видається важливою для предметних знань майбутніх перекладачів у світлі зростання вимог до рівня володіння майбутніми перекладачами комп’ютерними технологіями перекладу [9]. Перший досвід дослідження в цій сфері описано в попередній нашій праці [6]. Матеріалом аналізу в цьому разі послужила тема “Мови комп’ютерного програмування”, в межах якої паралельно здійснювався й добір двомовної термінології для позначення відповідних понять.

Таким чином, **метою** статті є опис досвіду складання поняттєвих схем на основі англomовних текстів для засвоєння головних понять, на яких ґрунтується згадана тема. Досягнення зазначеної мети передбачало розв’язання таких **завдань**: 1) обрати репрезентативний англomовний текст оригіналу (ТО) для перекладу та аналізу; 2) створити текст перекладу (ТП) українською мовою; 3) виділити в обох текстах терміни-відповідники; 4) скласти англо-український та українсько-англійський термінологічні словники; 5) визначити головні поняття, на яких ґрунтуються ТО і ТП, і встановити їх ієрархію; 6) знайти спосіб представлення поняттєвого складу згаданої сфери за допомогою відповідних схем і проаналізувати їх. Послідовність розв’язання цих завдань і визначила подальший зміст статті.

Матеріалом дослідження слугував оригінальний англomовний текст обсягом 5 766 друкованих знаків, запозичений з оригінального джерела [8], та його переклад українською мовою (обсяг ТП – 6 516 друкованих знаків). За кількістю слів українськомовний ТП є майже еквівалентним ТО: 871 слово проти 876 слів в англomовному ТО. Далі в обох текстах ми виділили терміни та термінологічні словосполучення: 103 термінологічні одиниці було виявлено в кожному тексті, які містили по 172 слова термінологічного характеру, тобто щільність текстів є високою – по 20 % термінологічних одиниць у ТО та ТП. На основі відібраних термінів та термінологічних словосполучень було складено українсько-англійський та англо-український словники.

Далі було визначено головні поняття, на яких ґрунтуються ТО і ТП, встановлено їх ієрархію та складено поняттєві схеми, які наведено на рис. 1 та рис. 2, а їх аналіз разом з англomовними відповідниками українських термінів подано далі. На рис. 1 представлено поняттєву схему теми Computer Programming Language.

Як випливає з рис. 1, центральним поняттям у цьому тексті є “Мови комп’ютерного програмування” (*Computer programming languages*), які поділяються на мови “низького рівня” (“*low-level language*”) та мови “вищого рівня” (“*high-level language*”). Мова комп’ютерного програмування використовується для формулювання точних команд (*instructions, operations*) цифровому комп’ютеру. Для формулювання згаданих команд використовується кілька бітів, ще кілька – для передачі операндів (*operands*) (даних, з якими виконує дії оператор), або ж для повідомлення про розташування (*locations*) наступної команди.

Мови “низького рівня” вимагають від програміста безпосереднього керування (*explicit management*), притаманними певному комп’ютеру характеристиками, а також знання архітектури (структури) комп’ютера (*must know architecture*). Мови “низького рівня” поділяються на машинну мову (*machine language*) і мову асемблера (*assembly language*).

Машинна мова складається з числових кодів (*numeric codes*) операцій, які використовуються комп’ютером. Коди у свою чергу складаються з двійкових цифр (“бітів”) (*binary digits*), які перетворюються як з, так і на шістнадцяткові (*hexadecimal digits*). Машинна мова легко перекладається мовами “вищого рівня” (*translation from higher languages*). Мова асемблера є рівнем вищою за машинну мову. Короткі мнемонічні коди (*mnemonic codes*) використовуються для команд, що дозволяє програмісту давати назви блокам пам’яті (*names for blocks of memory*). Структура мови асемблера дозволяє легко перекладати її на машинну мову.

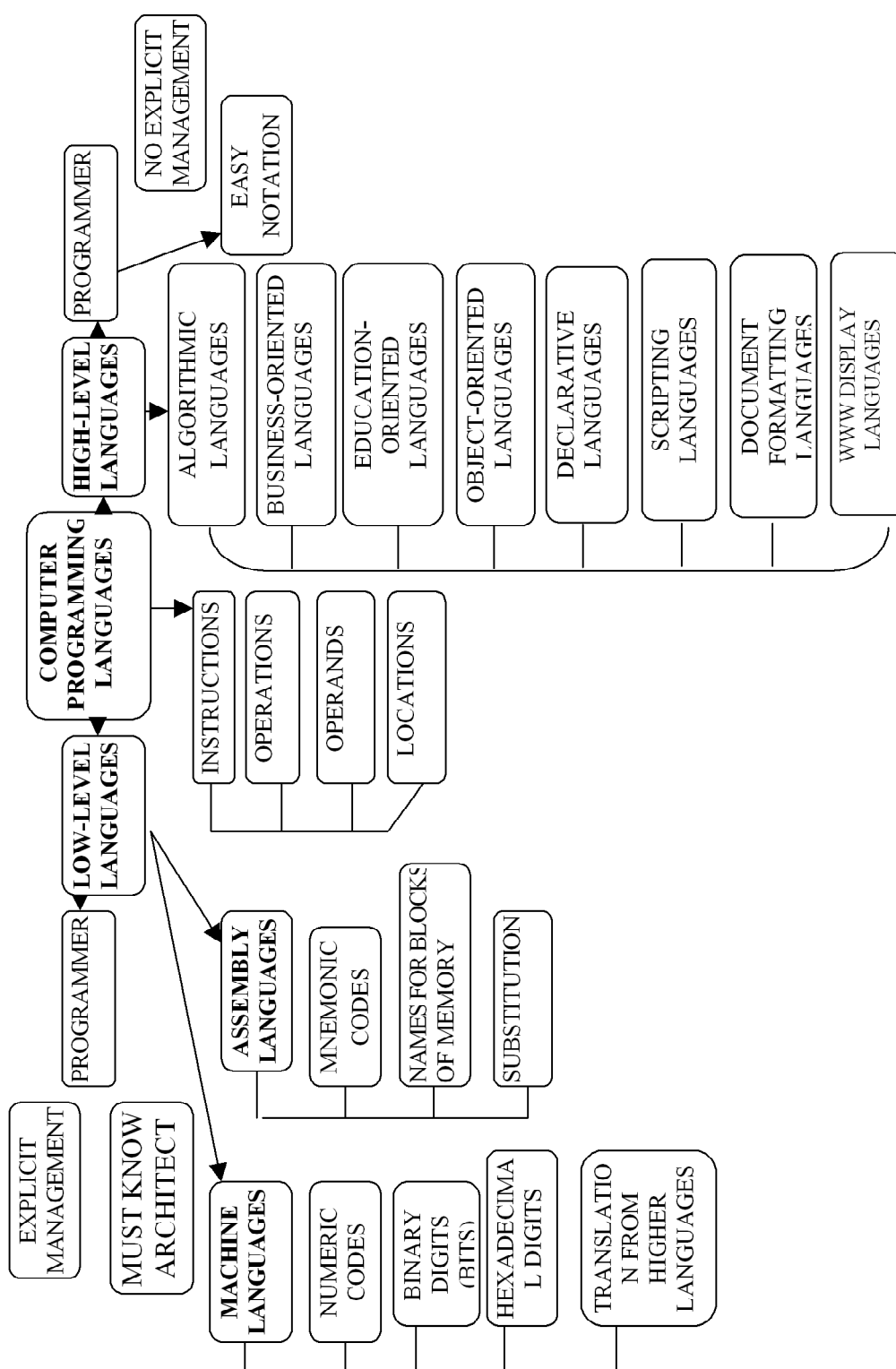


Рис. 1. Поняттєва схема теми Computer Programming Language

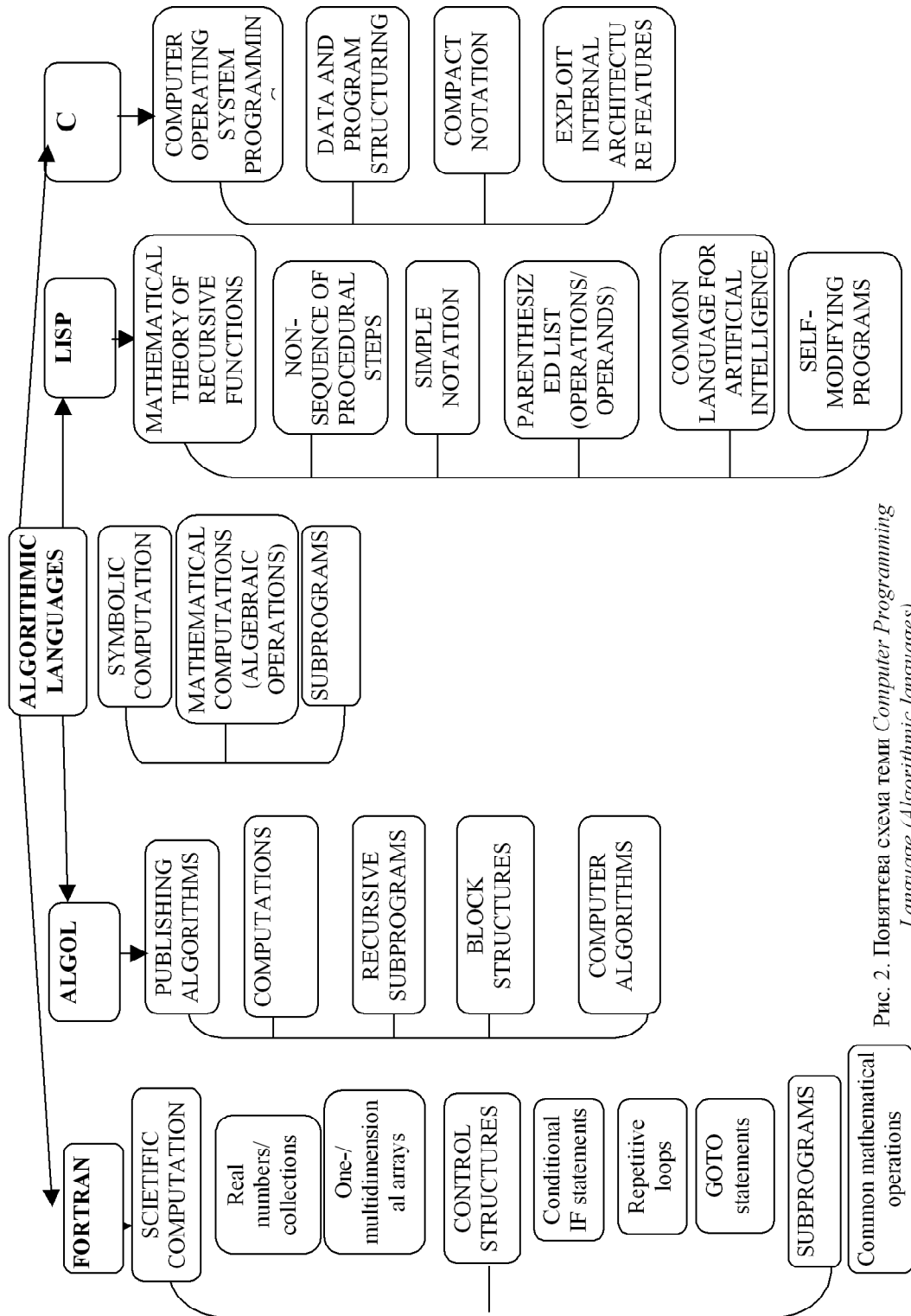


Рис. 2. Поняттєва схема теми Computer Programming Language (Algorithmic languages)

Мови “високого рівня” не потребують прямого керування (*no explicit management*) програмістами і ґрунтуються на зручнішій для використання системі записів (*easy notation*). У свою чергу мови “високого рівня” поділяються на алгоритмічні мови (*algorithmic languages*), мови підприємницького спрямування (*business-oriented languages*), мови програмування, що спрямовані на освіту (*education-oriented languages*), об’єктно орієнтовані мови (*object-oriented languages*), декларативні мови (*declarative languages*), мови сценаріїв (*scripting languages*), мови для форматування документів (*document formatting languages*), мови інтерфейсу всесвітньої павутини (*World Wide Web languages*).

Алгоритмічні мови (*algorithmic languages*), як свідчить рис. 2, призначені для вираження математичних або символічних обчислень (*mathematical/symbolic computations*) (напр., алгебраїчні операції (*algebraic operations*)), а також дозволяють використання певних підпрограм (*subprograms*). Ці мови були першими мовами високого рівня.

Перша з таких мов, що показані на рис. 2, “Фортран” (*FORTRAN*), була розроблена для наукових обчислень (*scientific computations*) з дійсними числами та їх сукупностями (*real numbers/collections*), розподіленими як одно- або багатовимірні масиви (*one-/multidimensional arrays*). Її керуючі структури (*control structures*) включали умовні судження, що починалися з “якщо” (*conditional IF statements*), повторювані цикли (*repetitive loops*) і оператора *GOTO* (*GOTO statements*) (“перейти до певного рядка за номером”). Багато підпрограм (*subprograms*) для стандартних математичних операцій (*common mathematical operations*) вдало використовуються завдяки мові “Фортран”.

Наступна алгоритмічна мова, “Алгол” (*ALGOL*) (див. рис. 2), була розроблена з метою побудови алгоритмів для використання у видавничій галузі (*publishing algorithms*), а також для обчислень (*computations*). Вона містила рекурсивні підпрограми (*recursive subprograms*) (здатність алгоритмів до самоактивації й розв’язання проблеми шляхом її перетворення на однотипну, але простішу), мала блочну організацію (*block structures*), а також протягом багатьох років нею публікувалися комп’ютерні алгоритми (*computer algorithms*).

Ще одна алгоритмічна мова, яку зазначено на рис. 2, “Лісп” (*LISP*), ґрунтувалася на математичній теорії рекурсивних функцій (*mathematical theory of recursive functions*), вона не є послідовністю процедурних кроків (*non-sequence of procedural steps*), а використовує просту систему запису, де операції (*operations*) та операнди (*operands*) (дані, з якими виконує дії оператор) подаються списком у дужках (*parenthesized list*). “Лісп” стала загальноприйнятною мовою (*common language*) для програмування штучного інтелекту (*artificial intelligence*), завдяки якій можна було написати здатні до навчання програми (*programs capable of “learning”*) штучного інтелекту як програми, що самовдосконалюються (*self-modifying programs*).

Останньою алгоритмічною мовою програмування, представленою на рис. 2, є Сі (C), яка була розроблена для програмування комп’ютерних операційних систем (*program computer operating systems*) і мала здатність структурувати дані й програми (*structure data and programs*) шляхом об’єднання менших одиниць. Вона використовує компактну систему запису (*compact notation*), а також усі функції внутрішньої архітектури (*features of internal architecture*) комп’ютера.

Отже, нами були розроблені поняттєві схеми, які визначають зміст навчання і можуть сприяти формуванню фахової компетентності майбутніх перекладачів, а також засвоєнню ними відповідної термінології, необхідної для перекладу текстів, що належать до категорії “Мови комп’ютерного програмування”. Однак методика використання таких схем у процесі навчання потребує додаткового вивчення, у чому і вбачаємо перспективу подальшого дослідження.

Література

1. Вакуленко М. О. Українська термінологія: комплексний лінгвістичний аналіз : [монографія] / М. О. Вакуленко. – Івано-Франківськ : Фоліант, 2015. – 361 с.
2. Переклад англomовної економічної літератури. Економіка США. Загальні принципи : навч. посібник / Л. М. Черноватий, В. І. Карабан, І. П. Ярошук. – Вінниця : Нова книга, 2007. – 496 с.
3. Переклад англomовної юридичної літератури: навч. посібник / Л. М. Черноватий, В. І. Карабан, Ю. П. Іванко, І. П. Ліпко. – Вінниця : Нова книга, 2006. – 656 с.
4. Черноватий Л. М. Переклад текстів англomовних засобів захисту інтелектуальної власності: патенти, знаки для товарів та послуг : навчальний посібник / Л. М. Черноватий, С. О. Царьова. – Вінниця : Нова книга, 2011. – 304 с.
5. Черноватий Л.М. Методика викладання перекладу як спеціальності : підручник / Л. М. Черноватий. – Вінниця : Нова книга, 2013. – 376 с.
6. Черноватий Л. М. Поняттєва схема як основа формування предметної та термінологічної складових фахової компетентності перекладачів галузевих текстів / Л. М. Черноватий, Ю. В. Купрієнко // Вісник Дніпропетровського університету імені Альфреда Нобеля. Серія “Педагогіка і психологія”. – 2016. – № 1 (11). – С. 373–378.
7. Alves F. Modelling translator’s competence: Relevance and expertise under scrutiny / F. Alves, J. L. Gonçalves // Doubts and Directions in Translation Studies. – Amsterdam ; Philadelphia : Benjamins, 2007. – P. 41–55.
8. Computer fundamentals [Electronic resource] / R. Tamassia // Algorithm Design. – New York ; Chichester ; Weinheim ; Brisbane ; Singapore ; Toronto : John Wiley & Sons Inc., 2002. – P. 57, 61, 63, 70, 77, 99, 114, 116. – Режим доступу : <http://cs-fundamentals.com/data-structures / introduction -to-data-structures.php>.
9. Gambier Y. Competences for professional translators, experts in multilingual and multimedia communication / Y. Gambier. – Brussel : EMT, 2009. – 104 p. – Режим доступу : http://ec.europa.eu/dgs/translation/external_relations/universities/documents/emt_competences_translators_en.pdf [accessed 6.01.2014].