

ШЛЯХИ ПРЕЗЕНТАЦІЇ ПРЕДМЕТНОЇ ІНФОРМАЦІЇ ПРИ НАВЧАННІ ГАЛУЗЕВОГО ПЕРЕКЛАДУ

Статтю присвячено важливості формування предметного й термінологічного компонентів фахової компетентності перекладача професійно-орієнтованих текстів. З цією метою було складено поняттєві схеми на основі англійських текстів за темою "Мова комп'ютерного програмування", які були детально проаналізовані. Були зроблені висновки про необхідність розробки експериментальної методики навчання галузевого перекладу на основі вищезгаданих схем.

Ключові слова: *фахова компетентність перекладачів професійно-орієнтованих текстів, предметна компетентність, термінологічна компетентність, поняттєва схема, комп'ютерні науки, мова комп'ютерного програмування.*

Фахова компетентність перекладача професійно-орієнтованих текстів частково є проявом загальної фахової компетентності (ФКП), де остання є метою навчання майбутніх перекладачів. ФКП розуміється дослідниками як володіння людиною здатністю і умінням виконувати певні трудові функції [1: 24]. Робоча модель ФКП включає п'ять компетентностей: екстралінгвістична, білінгвальна, особистісна, перекладацька та стратегічна [2]. Оскільки наше дослідження стосується галузевого перекладу, ми зосередимось на предметній складовій екстралінгвістичної компетентності. Предметна складова – певний діапазон знань, необхідний для перекладу у відповідній сфері, без наявності якого переклад значно ускладнюється, або навіть є неможливим [3]. Згадані знання є сумою понять, на яких ґрунтується відповідна сфера, а самі поняття в кожній мові позначаються відповідними термінами, володіння якими є обов'язковою передумовою фахового перекладу. В свою чергу, термін визначається як одиниця лексичного рівня (слово або словосполучення), яка позначає певне поняття у відповідній галузі людської діяльності [4]. Таким чином, майбутнім перекладачам необхідно засвоїти згадані поняття, а також оволодіти відповідною термінологією.

Для розв'язання такого завдання нами було обрано сферу "Комп'ютерні науки" (теми – "Мови підприємницького спрямування", "Мови програмування, що спрямовані на освіту", "Об'єктно-орієнтовані мови" і "Декларативні мови"). Оригінальні англійські тексти були перекладені українською мовою, були виділені терміни-відповідники в ТО та ТП, складено англо-український та українсько-англійський термінологічні словники, відібрані головні поняття, на яких ґрунтуються вищезгадані тексти, і встановлена їх ієрархія, а також складено поняттєві схеми ТО і ТП, які були проаналізовані.

Отже, **метою** статті є опис досвіду складання поняттєвих схем на основі англійських текстів для засвоєння головних понять згаданої теми. **Актуальність** нашого дослідження зумовлюється тим, що на сьогодні дослідження щодо формування предметного та термінологічного компонентів ФКП в межах згаданої сфери відсутні. Поняттєвий аналіз в межах кожної теми є **об'єктом** дослідження. Розроблені поняттєві схеми для згаданих тем сфери "Комп'ютерні науки" є **предметом** нашого дослідження.

Матеріалом дослідження послужив оригінальний англійський текст обсягом 8 663 друкованих знаків [5] та його переклад українською мовою (обсяг ТП – 10 625 друкованих знаків). Якщо за обсягом українськомовний ТП (в друкованих знаках) перевищує обсяг ТО англійською мовою, то за кількістю слів українськомовний ТП є майже еквівалентним ТО: 1 355 слів проти 1 322 слова в англійськомовному ТО. Кожен текст містив по 109 термінологічних одиниць, які в свою чергу містили по 201 слово термінологічного характеру. Отже, щільність текстів є високою – 15 % в англійськомовному ТО та 15 % в українськомовному ТП.

Далі на рис. 1, рис. 2 та рис. 3 представлені поняттєві схеми ТО й ТП, а також їх детальний аналіз. На рис.1 представлено поняттєву схему теми *Business and Education Oriented Languages*. Мови підприємницького спрямування (*business-oriented languages*). Перша з таких мов, що показана на рис. 1, "Кобол" (*COBOL*), використовує систему записів, схожу на англійську мову. Вона має структуру обліку даних (*record data structure*) для обчислення у галузі підприємництва (*business computations*), що ґрунтуються на оперуванні великими обсягами даних. Різномірні дані (*heterogeneous data*) (напр., ім'я, ідентифікаційний номер, вік, адреса) фіксуються в єдиному блоці за допомогою запису, який є важливим прикладом "об'єднання" даних ("*chunking*" data) в одному об'єкті, що є притаманним майже усім сучасним мовам.

Наступна мова підприємницького спрямування, "Ес-К'ю-Ел" (*SQL*) (див. рис. 1), була створена для визначення організації бази даних (*database*). Такі бази даних називаються реляційними (*relational*), оскільки "Ес-К'ю-Ел" (*SQL*) дозволяє відправляти їм запит стосовно певної інформації. Комерційні

програми баз даних (*commercial database*) широко використовують подібні мови для своїх запитів.

Мови програмування, що спрямовані на освіту (*education-oriented languages*). Як свідчить рис. 1, мова програмування "Бейсик" (*BASIC*) була розроблена для легкого засвоєння початківцями (*novices*), особливо студентами, що не спеціалізуються у комп'ютерних науках (*non-computer science majors*), а також для забезпечення безперебійного функціонування комп'ютера у випадку почергової роботи на ньому багатьох користувачів (*time-sharing computer*). "Бейсик" мала прості структури даних і систему записів (*simple data structure and notation*), також її було перекладено як підрядник (*translated line-by-line*) і виконано буквально, що дозволило швидше знайти програмні помилки. Крім того, вона стала популярною мовою перших персональних комп'ютерів (*early personal computers*). Її останні формати запозичили чимало структур даних і керуючих структур (*data and control structures*) з інших сучасних мов, що робить її потужнішою, але менш зручною для початківців.

Наступною мовою програмування, що спрямована на освіту (див. рис. 1) є "Паскаль" (*PASCAL*), яка призначена для навчання структурованому програмуванню (*structured programming*). Ця мова надавала особливого значення методичному використанню умовних керуючих структур (*conditional control structures*) та структур, що керують циклом без операторів переходу (*GOTO*) (*loop control structures*). "Паскаль" дозволяла визначати типи даних (типи даних, що визначались користувачем (*user-defined data types*)) для систематизації складної інформації (програміст давав імена складним даним, які перекладач міг перевірити ще до запуску програми для правильного використання).

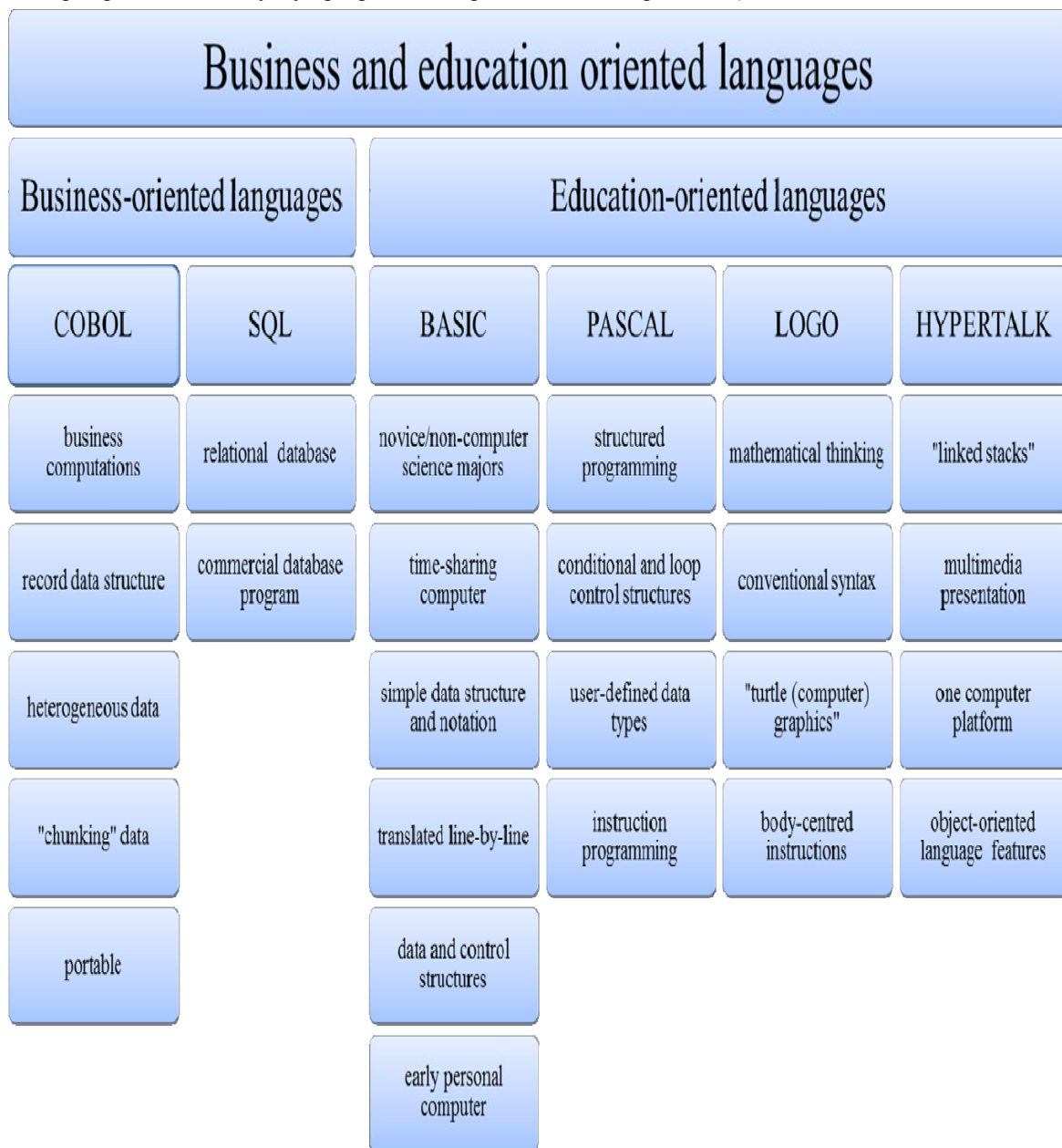


Рис. 1. Поняттєва схема теми *Business and Education Oriented Languages*

Мова програмування "Лого" (*LOGO*) (див. рис. 1) була створена для використання у галузі освіти, а деякі науковці використовували її для навчання школярів математичного мислення (*mathematical thinking*). "Лого" відзначалася традиційнішим синтаксисом (*conventional syntax*) та "черепашачою графікою" (*"turtle graphics"*), простим способом створення комп'ютерної графіки. Згадана графіка використовувала просторово-орієнтовані команди (*body-centered instructions*), що конкретизували дії відносно поточного місцезнаходження об'єкта.

Останньою мовою, що спрямована на освіту на рис. 1 є мова сценаріїв "Гайперток" (*HYPERTALK*), яка дозволила користувачеві швидко поєднувати текст, графіку та аудіозаписи у "зв'язані стеки" (*"linked stacks"*). Ця мова була особливо популярною серед педагогів, що практикували мультимедійні презентації (*multimedia presentations*) на заняттях. Хоча "Гайперток" мала чимало ознак об'єктно-орієнтованих мов програмування (*object-oriented language features*), компанія "Епл" не розробляла її для інших комп'ютерних платформ (*one computer platform*) і дозволила їй занепасти.

Об'єктно-орієнтовані мови (*object-oriented languages*). Як свідчить рис. 2, об'єктно-орієнтовані мови допомагають справлятися зі складністю (*manage complexity*) великих програм. Об'єкти розбивають по папкам дані і відповідні операції (*package data and operations*) так, що тільки операції є публічно доступними (*publicly accessible*), а внутрішні деталі структури даних є прихованими. Таке приховування інформації зробило масштабне програмування (*large-scale programming*) простішим. Крім того, об'єкти можуть бути виведені (*derived objects*) з більш загальних, "успадковуючи" їх властивості. Така ієрархія об'єктів (*object hierarchy*) дозволила визначити спеціалізовані (*specialized objects*) з них, не повторюючи все, що знаходиться в більш загальних.

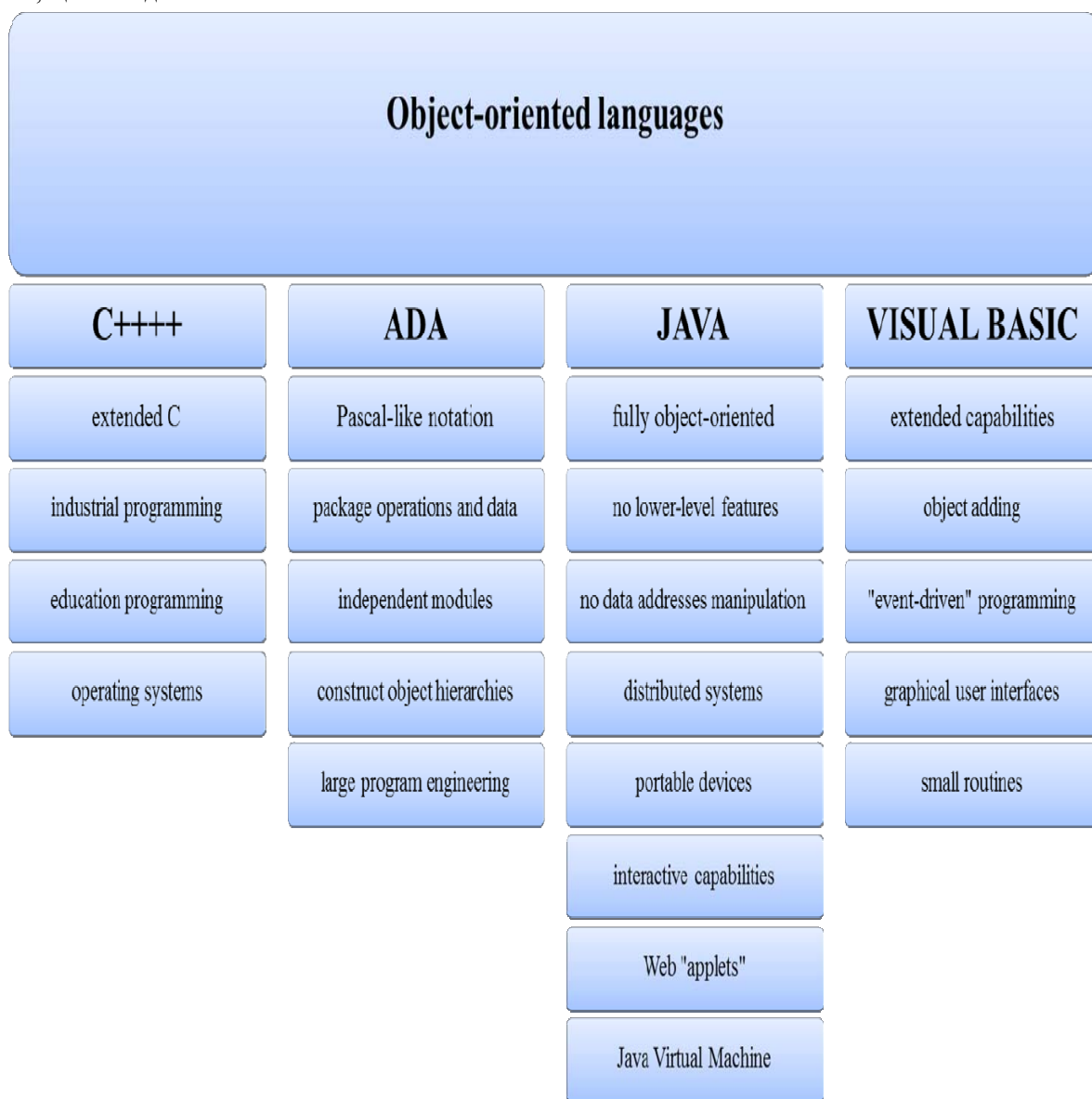


Рис. 2. Поняттєва схема теми *Object-oriented Languages*

Перша з таких мов, що показана на рис. 2 "Cі++++" (C++++), розширила мову "Cі" (C) (*extended C*) шляхом додавання нових об'єктів. Вона є однією з найважливіших мов як для освіти (*education programming*), так і для промислового програмування (*industrial programming*). Значні фрагменти багатьох операційних систем (*operating systems*), таких як *Windows 98* компанії "Майкрософт", написано мовою "Cі++++" (C++++).

Наступною об'єктно-орієнтованою мовою, репрезентованою на рис. 2, є "Ада" (*Ada*), що поєднала систему записів мови "Паскаль" (*combined Pascal-like notation*) з можливістю пакування операцій і даних (*package operations and data*) в незалежні модулі (*independent modules*). Вона є об'єктно-орієнтованою і має можливість побудови ієрархії з об'єктів (*construct object hierarchies*), також "Ада" залишається корисною мовою для розробки великих програм (*large program engineering*).

Мова програмування "Джава" (*Java*) (див. рис. 2) є повністю об'єктно-орієнтованою (*fully object-oriented*). Вона обходиться без властивостей нижчого рівня (*lower-level features*), включаючи маніпулювання адресами даних (*manipulate data addresses*), яке не є ані бажаним, ані корисним у програмах для розподілених систем (*distributed systems*) (що працюють на декількох комп'ютерах). Програми "Джава" перекладаються віртуальним механізмом "Джава" (*Java Virtual Machine*), що забезпечує їх мобільність (можливість використання на різних комп'ютерах). Окрім того, що "Джава" додає інтерактивних можливостей (*interactive capabilities*) мережі "Інтернет" через аплети (*Web applets*) (невеликі прикладні програми мовою "Джава"), вона широко використовується для програмування невеликих та переносних пристроїв (*portable devices*), таких як мобільні телефони.

Останньою об'єктно-орієнтованою мовою на рис. 2 є "Візуал Бейсик" (*Visual Basic*), яка була розроблена для розширення можливостей (*extended capabilities*) мови "Бейсик" (*Basic*) шляхом додавання об'єктів (*object adding*) та керованого подіями програмування (*"event-driven" programming*): клавіші, меню та інші елементи графічного інтерфейсу користувача (*graphical user interfaces*). Її використовують для програмування менших підпрограм (*small routines*).

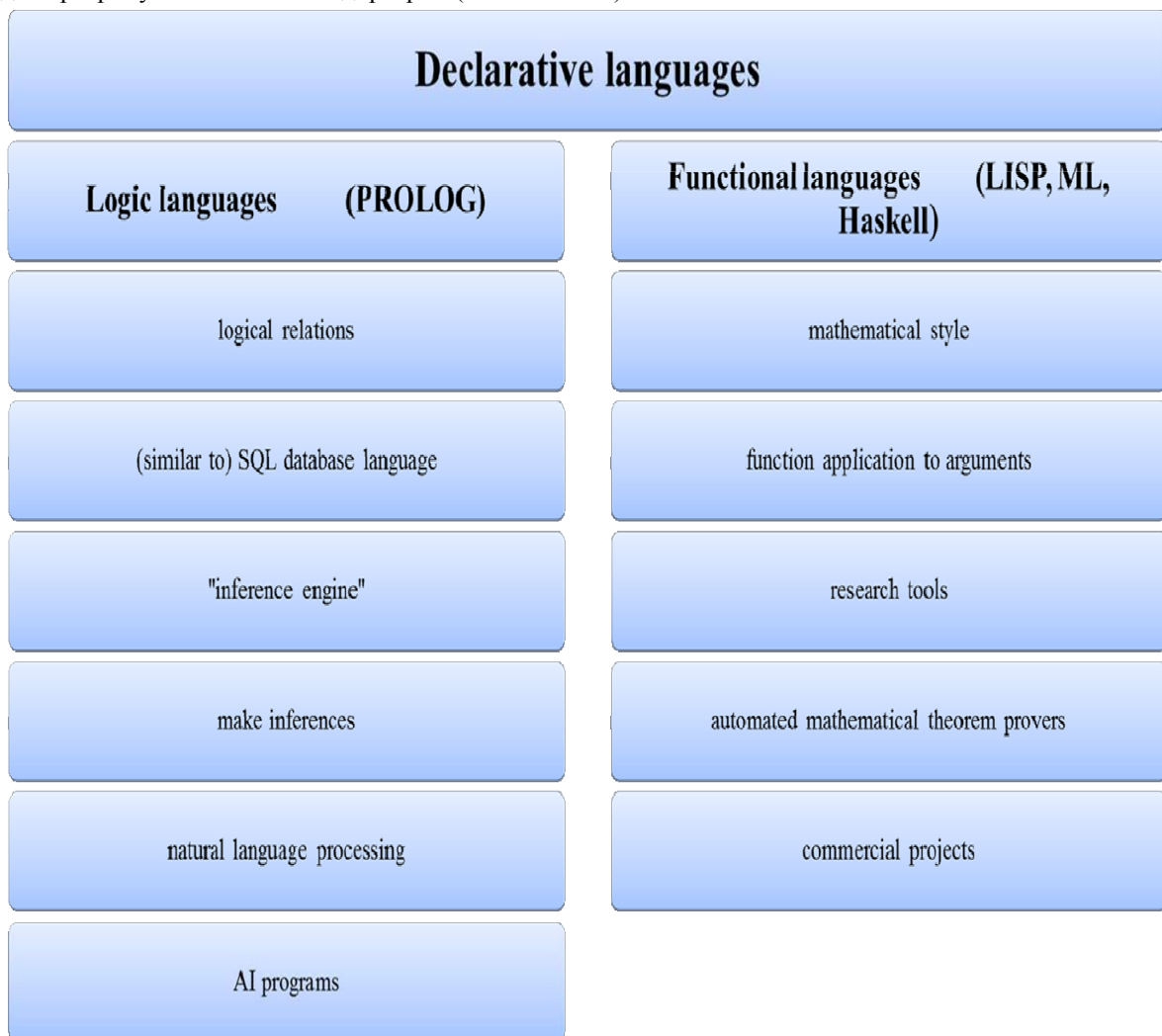


Рис. 3. Поняттєва схема теми *Declarative Languages*

Декларативні мови (*declarative languages*) (див. рис. 3), які також називають не процедурними (*nonprocedural*) або мовами дуже високого рівня (*high level*), включають наступні типи мов: логічні (*logic*) та функціональні (*functional*).

Як свідчить рис. 3, мови логічного програмування (*logic languages*), серед яких найбільш відомою є "Пролог" (*PROLOG*), викладають програму як набір логічних зв'язків (*logic relations*). Такі мови є схожими на мову бази даних (*similar to database language*) "Ес-К'ю-Ел" (*SQL*). Програма виконується за допомогою механізму умовиводу ("*inference engine*"), що відповідає на запити шляхом систематичного пошуку логічних зв'язків для формулювання умовиводу (*make inferences*), який буде відповіддю на запит. Мова "Пролог" широко використовується при обробці текстів, написаних природними мовами (*natural language processing*), а також у програмах для досліджень штучного інтелекту (*AI programs*).

Наступними декларативними мовами є функціональні мови (*functional languages*) (див. рис. 3), які мають математичний характер (*mathematical style*) і ґрунтуються на застосуванні функцій до аргументів (*function application to arguments*). Такі функціональні мови, як "Лісп" (*LISP*), "Ем-Ел" (*ML*) та "Гескел" (*Haskell*) використовують як інструменти дослідження (*research tools*) при розробці мов, в автоматизованих програмах для доведення теорем (*automated mathematical theorem provers*) та в деяких комерційних проєктах (*commercial projects*).

Висновки. Отже, нами були відібрані головні поняття, на яких ґрунтуються тексти теми "Мова комп'ютерного програмування" (а саме: "Мови підприємницького спрямування", "Мови програмування, що спрямовані на освіту", "Об'єктно-орієнтовані мови" і "Декларативні мови"), що належать до комп'ютерної галузі. На основі згаданих понять були розроблені поняттєві схеми, на яких представлені предметні знання, які можуть сприяти формуванню ФКП професійно-орієнтованих текстів. У рамках **подальшого наукового пошуку** ми вважаємо за доцільне додаткове вивчення методики використання таких схем у процесі навчання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ

1. Гавриленко Н. Н. Обучение переводу в сфере профессиональной коммуникации : [монография] / Н. Н. Гавриленко. – М. : РУДН, 2008. – 175 с.
2. Черноватий Л. М. Методика викладання перекладу як спеціальності : [підручник для студентів вищих закладів освіти освітньо-кваліфікаційного рівня "магістр" за спеціальністю "Переклад"] / Л. М. Черноватий. – Вінниця : Нова книга, 2013. – 376 с.
3. 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). – 435 с.
4. Вакуленко М. О. Українська термінологія : комплексний лінгвістичний аналіз : [монографія] / М. О. Вакуленко. – Івано-Франківськ : Фоліант, 2015. – 361 с.
5. Goodrich M. T. Computer fundamentals [Електронний ресурс] / M. T. Goodrich, R. Tamassia // Algorithm Design. – New York ; Chichester ; Weinheim ; Brisbane ; Singapore ; Toronto : John Wiley & Sons Inc., 2002. – P. 57–116. – Режим доступу до сайту : <http://cs-fundamentals.com/data-structures/introduction-to-data-structures.php>.

REFERENCES (TRANSLATED & TRANSLITERATED)

1. Gavrylenko N. N. Obuchenii perevodu v sfere professional'noy kommunikatsii [Translation Teaching in the Sphere of Professional Communication] : [monografiia] / N. N. Gavrylenko. – Moscow : RUDN, 2008. – 175 p. (In Russian).
2. Chernovatii L. M. Metodyka vykladannia perekladu yak spetsial'nosti [Methodology of Teaching Translation as a Profession] : [pidruchnyk dlia studentiv vyshchikh zakladiv osvity osvitno-kvalifikatsiinogo rivnia "magistr" za spetsialnistiu "Pereklad"]. – Vinnytsya : Nova knyga, 2013. – 376 p. (In Ukrainian).
3. Chernovatyi L. M. Ponyattieva skhema yak osnova formuvannia predmetnoi ta terminolohichnoi skladovykh fakhovoyi kompetentnosti perekladachiv haluzevykh tekstiv [Notional Scheme as the Foundation to Form TC Subject-Matter and Terminological Components] / L. M. Chernovatyi, Yu. V., Kupriyenko // Visnyk Dnipropetrovskoho universytetu imeni Alfreda Nobelya. Seriya "Pedahohika ta psykholohiya" [Journal of Alfred Nobel university. Pedagogy and Psychology]. – № 1(11), Dnipropetrovs'k, 2016. – 435 p. (In Ukrainian).
4. Vakulenko M. O. Ukrayins'ka terminolohiia : kompleksnyi lnhvistychnyi analiz [Ukrainian Terminology : intergrated linguistic analysis] : [monografiia]. – Ivano-Frankivsk : Foliant, 2015. – 361 p. (In Ukrainian).
5. Goodrich M. T. (2002). Computer fundamentals [Electronic resource] / M. T. Goodrich, R. Tamassia // Algorithm Design. – New York -Chichester-Weinheim-Brisbane-Singapore-Toronto : John Wiley & Sons Inc., 2002. – P. 57–116. – Rezhym dostupu do сайtu : <http://cs-fundamentals.com/data-structures/introduction-to-data-structures.php>.

Куприенко Ю. В. Методы презентации предметной информации при обучении отраслевого перевода.

Статья посвящена важности формирования предметного и терминологического компонентов профессиональной компетентности переводчика профессионально-ориентированных текстов. С этой целью были составлены понятийные схемы на основе англоязычных текстов по теме "Язык компьютерного программирования", которые были детально проанализированы. Были сделаны выводы о необходимости разработки экспериментальной методики обучения отраслевому переводу на основе вышеупомянутых схем.

Ключевые слова: профессиональная компетентность переводчиков профессионально-ориентированных текстов, предметная компетентность, терминологическая компетентность, понятийная схема, компьютерные науки, язык компьютерного программирования.

Kupriyenko Yu. V. Presentation Methods of Subject-Matter Information while Non-Fiction Translator Teaching.

The article is devoted to the profession-oriented text analysis for the field of computer science (topic – Computer Programming Language). It describes the process of drawing up a notional scheme and the selection of English and Ukrainian terminological units for the acquisition of TC subject-matter and terminological components. The following steps were taken to solve the abovementioned task: 1) The original English texts (topics – Business and education oriented languages, Object-oriented languages and Declarative languages) were chosen to translate and analyze; 2) The texts were translated into Ukrainian; 3) The appropriate terms were selected in both English and Ukrainian texts; 4) English-Ukrainian and Ukrainian-English terminological vocabularies were made based on the selected terms; 5) The main subject-matter notions of both English and Ukrainian texts were selected and their hierarchy was established; 6) The notional schemes based on the abovementioned notions and their hierarchy were made. The notional schemes represent subject-matter knowledge that can help to form professional translator's competence.

The methodology of using such schemes while non-fiction translator teaching needs to be studied within the framework of our further research.

Key words: professional translator's competence, subject-matter competence, terminological competence, notional scheme, computer sciences, computer programming language.