

АНГЛІЙСЬКОМОВНІ ТА СЕМІОТИЧНІ КОНСТИТУЕНТИ МОВИ ПРОГРАМУВАННЯ PYTHON

Лазебна Н. В.

*кандидат філологічних наук, доцент,
доцент кафедри теорії та практики перекладу
Національний університет «Запорізька політехніка»
вул. Жуковського, 64, Запоріжжя, Україна
orcid.org/0000-0001-5886-693X
natalialazebnaya@gmail.com*

Ключові слова:

*лінгвосеміотичний континуум,
кодування, вербалізація,
полівалентність, код
програми, англійськомовний
дигітальний дискурс.*

Розвиток обчислювальної техніки супроводжується створенням нових і вдосконаленням наявних способів і засобів спілкування «людина – машина», а саме мов програмування (МП). Під мовами програмування розуміють правила подання даних і запису алгоритмів їх оброблення, які автоматично виконуються «машиною». У більш абстрактному вигляді мови програмування є засобом створення програмних моделей об'єктів і явищ зовнішнього світу. Мова програмування є кодом, який потребує інтенсивного осмислення та формалізації. Розуміння такої мови передбачає знання та переосмислення вже наявної інформації в тій чи іншій галузі. Здебільшого, будучи прописаними кодами за допомогою засобів англійської мови, інтерфейси мов програмування дозволяють проектувати програмні продукти для всього світу. Саме мова програмування та спеціальні коди потребують удосконалення з погляду лінгвістики. Однак вербальне оформлення кодів відбувається за допомогою засобів англійської мови, але їхній лінгвальний базис зазвичай залишається поза увагою. Управління програмами також потребує простішого інтерфейсу. Лінгвістам варто звернути увагу на вербалізацію коду програми засобами англійської мови, її опису для скорішого інтегрування у глобальний дигітальний соціопростір. Мова програмування є кодом, який потребує інтенсивного осмислення та формалізації з погляду інформатики та лінгвістики. Спираючись на знання про інтелектуальні фонди, методи, культуру й універсальність англійської мови, на якій будується мова програмування, інтерфейси комп'ютерних програм дозволяють проектувати програмні продукти для всього світу. Мова програмування як інтегративний елемент лінгвосеміотичного базису англійськомовного дигітального дискурсу потребує динамічного та мультимодального аналізу. Концептосфера й система знаків англійськомовного дигітального дискурсу (АДД) актуалізує смисловий зміст, закладений у комунікації «людина – машина» у вербальній і невербальній формах. Дотепер створені десятки різних мов програмування, від найпримітивніших до близьких до природної мови людини. Мови програмування є інструментом програміста для створення програм, однак людині зручніше описувати модельований об'єкт у термінах предметної області засобами природної англійської мови, виводити вербальний код мови програмування на перше місце.

ENGLISH-LANGUAGE AND SEMIOTIC CONSTITUENTS OF PYTHON PROGRAMMING LANGUAGE

Lazebna N. V.

*Candidate of Philological Sciences, Associate Professor,
Associate Professor at the Department of Theory and Practice of Translation
Zaporizhzhya Polytechnic National University
Zhukovskoho str., 64, Zaporizhzhia, Ukraine
orcid.org/0000-0001-5886-693X
natalialazebnaya@gmail.com*

Key words: *linguistic and semiotic continuum, coding, verbalization, polyvalence, program code, English-language digital discourse.*

The development of computer technology is mediated by a variety of new and existing methods and means of communication “Human – Machine”, namely, programming languages (PLs). The concept programming languages determines the rules of data representation and algorithms for their processing, which are automatically performed by the “machine”. In broader terms, programming languages is a means for creating software models of world’s objects and phenomena. Programming language is the code that requires intensive comprehension and formalization. To comprehend this type of language, there is a need for knowledge and reconsideration about the existing information in a given area. In most cases, programming languages are verbalized by means of the English language, which globalizes their software interfaces. There is a need for programming languages linguistic refinement. Software management also requires a simpler interface. Linguists should pay attention to the verbalization of the program code by means of the English language for faster integration into the global digital social space.

The programming language contains special codes, which require detailed clarification and formalization in terms of computer sciences and linguistics. Based on knowledge, methods, culture and specifics of the English language, software interfaces allow global users to design software products for the whole world. programming languages serve as an integrative element of the linguistic and semiotic basis of the English-language digital discourse. This phenomenon requires dynamic and multimodal analysis. The conceptual sphere and sign system of the English-language digital discourse actualizes the semantic content inherent in human-machine communication in both verbal and nonverbal forms. To date, dozens of different programming languages have been created, especially those resembling natural human language. Programming languages is a programmer’s tool for creating programs, but it is more convenient for a user to describe the simulated object in terms of the subject area by means of Natural English language, prioritizing the verbal code of programming languages.

Постановка проблеми. Для обробки природної мови використовується мова програмування (далі – МП) Python спільно з відкритою бібліотекою «Інструментарій природної мови» (NLPK). Для полегшення роботи із цією МП та прописаними англійськомовними командами, залучення лінгвістів до співпраці із програмістами є необхідним, оскільки можливим видається винаходження певного логічного й аргументованого принципу роботи штучних мов, їх більш ефективної роботи у взаємодії «людина – машина – людина» та її варіацій, а також як важливого інструментарію обробки природної мови.

Мета статті. Мови програмування як основний інструмент обробки природної мови (NLP) базуються на мовних засобах англійської мови (вербальна основа) та семіотичному ґрунті, тобто є лінгвоматематичним континуумом. Наукова розвідка присвячена ідентифікації ролі англійської мови як основи штучної мови (мови програмування) для самоаналізу (обробки природної мови, Natural language processing) для медіації взаємодії «людина – машина». Досліджується переродження, або кодування вербальних команд у мову, зрозумілу машині.

Виклад основного матеріалу дослідження. Єдність вербальних та символічних позначень у

сучасному англійськомовному дигітальному дискурсі (далі – АДД) свідчить про впорядкованість та водночас варіативність конститuentів такого типу дискурсу. АДД – це письмовий процес і результат взаємодії учасників комунікації, опосередкованої комп'ютерними технологіями, або тематично з ними співвіднесеними текстами на МП та їхнім лінгвосеміотичним базисом, режимами їх кодування-декодування у процесі комунікації «людина – людина», «людина – машина», «людина – машина – людина» та їхніх варіацій.

«Розгортання» МП Python репрезентує певну побудову лінгвального фундаменту галузі ІТ. Ідентифікатори представлено на Рис. 1. Вони є основними тригерами, які детермінують функціонування досліджуваної МП. Англійськомовні дієслова, сполучники, прикметники тощо модифікують та корелюють лінгвальні та логічні зв'язки.

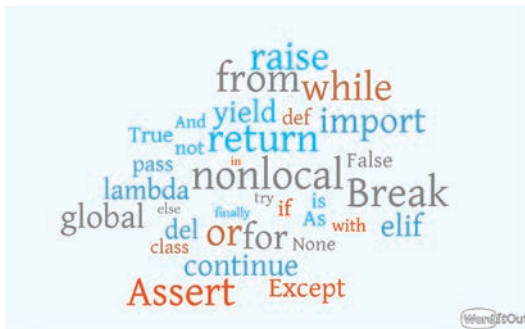


Рис. 1. Ідентифікатори в мові Python

Функційність лінгвосеміотичних елементів продукує мережі відношень, у яких кожен із таких елементів у контексті мов програмування чи природної мови (англійська) може стати емергентним, тобто перероджуватися із слова чи символу в команду, яка, у свою чергу, екстраполюється в англійськомовному дигітальному дискурсі.

Наприклад, декодування інструкції в МП Python відбувається так: функція `print` у поєднанні з методом `str.format()`. `print (<рядок форматування>.format (<перелік форматується виразів>))`.

Рядок форматування містить текст і специфікації формату, взяті у фігурні дужки `{}`. Специфікація формату може бути порожньою, містити номер виразу зі списку форматується виразів (які нумеруються, починаючи з нуля) або номер виразу зі специфікацією формату. Увесь текст, що знаходиться поза цими дужками, виводиться без змін.

Наведемо кілька прикладів з результатами виведення:

Синтаксис: `<ім'я> = input (<s>)`.

Функція `input` виводить на екран рядок `<s>`, потім зчитує введений користувачем рядок і пов'язує `<ім'я>` із цим строковим значенням. Приклади використання:

```
d = 4
s = input ('number:')
print (d * s)
print (d * int (s))
print (d / float (s)).
```

Очевидна циклічність МП Python. Наприклад, крім циклу `while`, існує ще одна конструкція, що дозволяє багаторазово повторювати той самий набір операцій. Цикл `for` можна записати різними способами, зараз нам цікавий той варіант, який дозволяє перебирати елементи впорядкованих послідовностей, зокрема рядків.

Синтаксис його такий:

`for <ch> in <str>:`

`<Складник оператора>`,

де `<ch>` – ім'я змінної (яка може не мати жодного значення до початку виконання циклу `for`), `<str>` – вираз строкового типу, `<складник оператора>` – один або більше операторів.

Однак циклічність відбувається в межах визначених умов, які задано англійськомовними сполучниками `while / for`. Окресленість умов також задають символи досліджуваної МП. Номінативний фонд сучасної англійської мови екстраполює динамічні процеси свого розширення, що виражається у використанні знакоутворювальних і знакозмінювальних мовних та немовних кодів. До цих процесів можуть бути віднесені: розширення і / або звуження / трансформація набору знаків (формування нових знаків, розвиток полівалентності знака, поглиблення значення знака); деталізація семіотичних систем і їх розширення; реконфігурація семіотичних систем (трансформація знака, реінтерпретація знака, реконструкція знака).

На прикладі виявлених засобами МП Python помилок можна побачити динамічну природу знака, інтегративний потенціал останнього з англійськомовними лінгвальними засобами.

```
N = int (input ('Input N:'))
```

```
s = 0
```

```
for i in range (N):
```

```
s = s + i ** 2
```

```
NameError: name 'range' is not defined.
```

Як бачимо, інтерпретатор, дійшовши до рядка з помилкою, вказав нам, що ім'я `range` йому невідомо (`NameError: name 'range' is not defined`). Знайти і виправити такі помилки зазвичай досить просто, також завдяки тому, що всі вбудовані функції (`range`, `len`, `sorted`, `sum`, `int` тощо) виділяються кольором. У такому разі помилка виникла на мовному рівні, тобто під час некоректного використання англійськомовних засобів у МП Python.

У межах англійськомовного дигітального дискурсу знаки і символи адаптуються під наявну структуру та закони інформаційних технологій. З одного боку, англійська мова постає як первинна система-донор МП, АДД загалом, а з іншого –

репрезентує «вторинні моделюючі системи» [7, с. 123], що дозволяють створювати штучні мови. Засобами природної англійської мови (далі – П(А)М) відбувається створення техногенного англійськомовного кодованого світу та його фрагментів [6, с. 18].

Мови програмування – нібито власні мови, але вони є вербалізованими засобами англійської мови. Також до таких мов залучено візуальні та графічні елементи, символи для опосередкування комунікації «людина – машина». Різноманітність знаків і символів, залучених в англійськомовний дигітальний простір, свідчить про його лінгвоматематичний симбіоз.

У мовах програмування комунікація є опосередкованим кодом, як машинним, так і програмним. В АДД репрезентовано засоби кодування подій і їхніх характеристик, логіки та закономірностей, порядку речей і відношень у віртуальному просторі, який деякою мірою заміщує простір буття сучасників [1, с. 40; 2, с. 314]. Візуально, тобто у знаково-символічній формі, можна побачити лише частину інформаційної технології – зовнішню форму й інтерфейс користувача [9]. У такому разі система відкритих вихідних кодів через вивчення використаних алгоритмів, структур даних, технологій, методик та інтерфейсів створює ефективну комунікацію між користувачем і машиною.

Як уважає К. де Суза, семіотичне дослідження допомогло б змодельовати процес розуміння того, як люди використовують мови програмування, як візуалізують технологічні процеси, щоб створити програми для людей, які забезпечать ефективну комунікацію «людина – машина» [14, с. 580].

```
#!/usr/bin/env python3
# -*- coding: UTF-8 -*-
"""
test_fileinput.py: Process all the files given in the command-line arguments
Usage: test_fileinput.py file1, file2, file3, ...
"""
import fileinput

def main():
    """Get lines from all the file given in the command-line arguments (sys.argv)"""
    lineNumber = 0
    for line in fileinput.input():
        line = line.rstrip() # strip the trailing spaces/newline
        lineNumber += 1
    print('{}: {}'.format(lineNumber, line))
if __name__ == '__main__':
    main()
```

Рис. 2. Текстова репрезентація МП Python

Однак аналіз тенденцій розвитку семіотичних досліджень у галузі інформаційних технологій показав, що вони здебільшого спрямовані на семіотику соціальних мереж і засобів масової інформації в мережі Інтернет, а семіотичні дослідження області мов програмування поки перебувають у стадії становлення. Крім того, математичний базис мов програмування є домінуючим фокусом у дослідженнях такого типу [5, с. 13]. Лінгвістичний аспект мов програмування все ще залишається поза увагою науковців [3, с. 36; 4, с. 92].

Інтегративним видається саме лінгвосеміотичний континуум, який створює деякий фрейм, здатний трансформуватися згідно з вимогами відносин «людина – машина». Наступний уривок на МП Python репрезентує використання вербальних / англійськомовних та невербальних засобів, які й створюють той самий лінгвосеміотичний континуум, на якому будуватиметься АДД.

Можлива інтерпретація «технічних кодів» співвідноситься з лінгвальним базисом англійської мови [11; 8, с. 245]. Питання щодо спілкування людини з машиною можна вирішити за допомогою технічних кодів, але необхідно їх розглядати не як стереотипи чи даність, а як модифіковані лінгвальні та візуально-графічні засоби П(А)М [10, с. 177]. У віртуальному світі об'єкти інтерпретації є багатовимірними і суб'єкт інтерпретації, тобто знак, втрачає свою однозначність [12].

Висновки. Мови програмування та їхні коди потребують удосконалення з погляду лінгвістики, адже питання з їх вираженням та кодованістю вирішуються програмістами. Однак вербальне оформлення кодів відбувається за допомогою засобів англійської мови, але їхній лінгвальний базис зазвичай залишається поза увагою. Замість вирішення питання семантичної неоднозначності у спілкуванні «людина – машина» пропонуються вдосконалення технічного коду, знак зберігає лише своє значення, а організація коду як лексико-семіотичної єдності залишається поза увагою. Управління програмами також потребує простішого інтерфейсу. Увагу лінгвістів варто звернути до вербалізації коду програми засобами англійської мови та його опису. Таким способом суспільство матиме змогу швидше інтегруватися у глобальний дигітальний соціопростір.

Мова програмування є кодом, який потребує інтенсивного осмислення та формалізації. Розуміння такої мови передбачає знання попередньої інформації про інтелектуальні фонди, методи, цілі й обмеження мови тієї культури, у межах якої створюється програмний продукт. Здебільшого, будучи прописаними засобами англійської мови, інтерфейси

Таблиця 1

Розмежування параметрів природної мови та мов програмування

РОЗМЕЖУВАННЯ ПАРАМЕТРІВ П(А)М ТА МП	
Мова програмування (МП)	
<ul style="list-style-type: none"> – створена людиною для комп'ютера; – письмова форма; – модульність: функції / процедури (в імперативному або функціональному порядку мови) або об'єкти (в об'єктно орієнтованих мовах) дозволяють порівняно вільне заповнення модулів; – відносно вільний порядок речень (декларативні мови); – інтерактивний об'єктно-залежний порядок мов, які є орієнтованими на «події» програми; – лише «основна» частина програми (коли вона існує) є чітко лінійною. 	
Лексичні одиниці	
П(А)М	МП
– лексичні одиниці можна виокремити в категорії частин мови (N, V, ...);	– МП складаються з дискретних одиниць;
	– еквівалентні класи букв та цифр
– лексичні (N, V, Adj, ...): належать до відкритого списку, є референціями до речей, властивостей, дій у «зовнішньому світі»;	зв'язки (ключові слова, назви функцій);
– граматичні одиниці (prep, conj, займенники) важливі для граматичної структури більше, ніж для референцій до «зовнішнього світу».	– основні інструкції або команди – це набір смислових зв'язків;
	– двозначність.
Синтаксис МП	
<ul style="list-style-type: none"> – формальні граматики моделюють синтаксис; – характеризується за ознакою компіляції й інтерпретації; – два рівні синтаксису перевіряються компілятором та інтерпретантом; – чітко сформовані основні вказівки та їх послідовність; – ті самі змінні можуть посилатися на ті самі дані; – другий рівень синтаксису, більш складний, описує процес виконання програми. 	
Семантика	
П(А)М	МП
– композиційна: значення залежить від семантики і синтаксису;	– на рівні зв'язок та змінних: значення даних;
– дистрибутивність слова визначається його оточенням (Вітгенштейн).	– відмінність від природної мови: немає спільного світу між програмами, окрім ієрархічно організованих класів і об'єктів.
<i>На операційному рівні</i>	
– на лексичному рівні: поєднанням примітивних семантичних одиниць (компонентний аналіз, Шенк) або за позицією в мережі (Ф. де Сосюр, семантичні мережі);	– та сама основна інструкція може призвести до різних результатів (залежно від значення змінних, подій, випадкового вибору тощо);
– у П(А)М семантика може існувати окремо від граматики.	– та сама операція може бути виражена різними основними інструкціями.
<i>На рівні програми:</i>	
– та сама програма призводить до різних результатів (залежно від користувача та введених ним даних), але реалізує унікальну функцію / алгоритм;	
– різні програми можуть реалізувати ту саму функцію / алгоритм.	
Граматичні категорії МП	
<ul style="list-style-type: none"> – знаки та символи ({} , “.”, табуляція в Python) або ключові слова (var, if / then / else, function, begin / end); – лексичні одиниці: імена, що надаються змінним або функціям / процедурам, або предикати; – велика різниця з англійською мовою: одиниці залежать від програми і є індивідуалізованими; – типи даних (булеві, системні, реальні, рядок, масив, функції тощо) є аналогічними категоріям POS для цих одиниць: класи еквівалентності для замінюваності за збереження «компіляції»; – різниця із природною мовою: деякі мови програмування приймають потенційно нескінченну кількість функційних ознак. 	
Функції природної мови щодо мови програмування	
П(А)М	МП
– референція;	– формальна мова на рівні інструкції і програмування може сприйматися як суворо композиційна;
– декларативні мови є референційними;	– фатична частина комунікації протокольних програм;
– мова наказів: основа імперативних мов;	– МП штучно розроблені та не мають жодного усного каналу, ніж природна мова;
– фатична: частина програм протоколів спілкування;	– поняття недвозначності не настільки чітке в МП, але існує глобальний композиційний детермінізм на рівні алгоритму (та сама причина, той самий ефект);
– металінгвальна: компілятор містить опис мови.	– відсутність спільного домену інтерпретації між програмами, написаними тією самою МП;
	– МП характеризується змінними, систематичною індексованістю (змінні) та рекурсивністю;
	– відсутність «семантики» без синтаксису;
	– спільний домен посилання.

комп'ютерних програм дозволяють проектувати програмні продукти для всього світу. Формування лінгвосоціотичного базису англійськомовного дигітального дискурсу є тривалим динамічним проце-

сом, під час якого концептосфера і система знаків актуалізує смисловий зміст, закладений у комунікації «людина – машина» у вербальній і невербальній формах, зазнає значних трансформаційних змін.

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

МП – мова програмування.
 КП – комп'ютерна програма.
 АДД – англійськомовний дигітальний дискурс.
 П(А)М – природна англійська мова.
 NLP – обробка природної мови.

ЛІТЕРАТУРА

1. Буднік А.О. Особливості вивчення дискурсу в писемній та усній комунікації. *Науковий вісник Дрогобицького державного педагогічного університету імені Івана Франка. Серія «Філологічні науки (мовознавство)»*. 2016. № 5 (1). С. 39–42.
2. Бузаров В.В. К вопросу о термине «дискурс». *Горизонты гуманитарного знания* : сборник научных трудов Пятигорского государственного университета. 1999. 314 с.
3. Иванов Л.Ю. Язык интернета: заметки лингвиста. *Словарь и культура русской речи*. Москва : Азбуковник, 2000. С. 35–45.
4. Иванов В.М. Интеллектуальные системы. Екатеринбург : Амфора, 2015. 92 с.
5. Клейнер Г.Б. Междисциплинарность, системность, гармония – ориентиры развития социально-экономических исследований. *Перспективы развития междисциплинарных социально-экономических и гуманитарных исследований* : доклады и выступления участников Круглого стола / отв. ред. Г.Б. Клейнер. Ростов-на-Дону : Изд-во ЮФУ, 2015. С. 12–32.
6. Лотман Ю.М. Структура художественного текста. *Об искусстве*. Санкт-Петербург, 1998. С. 14–28.
7. Успенский Б.А. Избранные труды. Москва : Гнозис, 1994. Т. 2 : Язык и культура. URL: <https://biblioclub.ru/index.php?page=book&id=219940>.
8. Cantor N.E., Thomas E.A. Control of attention in the processing of temporal and spatial information in complex visual patterns. *Journal of Experimental Psychology*, 2007. № 3. P. 243–250.
9. Cappelle B. Particle Patterns in English. A Comprehensive Coverage. 2005. URL: <https://kulak.kuleuven.be/nl/onderzoek/Onderzoeksdomeinen/language-literature-translation/letteren/onderzoektaalkunde/fest/descriptiveenglishgrammar/bert-cappelle/bert-cappelle-proefschrift.pdf>.
10. Dubose N. Models of the Visual Cortex. New York : Wiley, 1985. 374 p.
11. Feenberg A. Transforming Technology : A Critical Theory Revised. 2002. URL: [http://lst-iiiep.iiiep-unesco.org/cgi-bin/wwwi32.exe/\[in=epidoc1.in\]/?t2000=0 14688/\(100\)](http://lst-iiiep.iiiep-unesco.org/cgi-bin/wwwi32.exe/[in=epidoc1.in]/?t2000=0 14688/(100)).
12. Sabrina A. Huang and Jeremy Bailenson. 2019. Ch. 4 : Close Relationships and Virtual Reality. URL: <https://vhil.stanford.edu/mm/2019/01/huang-close-relationships-and-vr.pdf>.
13. Souza C. A framework for analyzing and understanding online communities. *J Preece*. 2004. P. 579–610.

REFERENCES

1. Budnik, A.A. (2016) Osoblivosti vivchennya diskursu v pisemnij ta usnij komunikaciji [Features of the study of discourse in written and oral communication]. *Scientific Bulletin of Drohobych State Pedagogical University Ivan Franko*. Ser. : Philological Sciences (Linguistics), № 5 (1). P. 39–42.
2. Buzarov, V.V. (1999) K voprosu o terminye “diskurs” [On the question of the term “discourse”]. *Horizons of humanitarian knowledge: Sat. scientific works of PGLU*. Pyatigorsk. 314 p.
3. Ivanov, L.Yu. (2000) Yazyk interneta: zamyetki lingvista [The language of the Internet: notes of a linguist]. *Dictionary and culture of Russian speech*. Moskov: Azbukovnik, P. 35–45.
4. Ivanov, V.M. (2015) Intellektual'nye sistemy [Intelligent systems]. Yekaterinburg: Amphora. 92 p.
5. Kleiner, G.B. (2015) Mezhdisciplinarnost', sistemnost', garmoniya – orientiry razvitiya social'no-ekonomicheskikh issledovanij [Interdisciplinarity, systematics, harmony – guidelines for the development of socio-economic research]. *Prospects for the development of interdisciplinary socio-economic and humanitarian research: Reports and speeches of roundtable participants* / resp. ed. G.B. Kleiner. Rostov-on-Don: SFU Publishing House. P. 12–32.
6. Lotman, Y.M. (1998) Struktura hudozhestvennogo teksta [The structure of the literary text]. *About art*. Saint-Petersburg: Iskustvo-SPB. P. 14–28.
7. Uspensky, B.A. (1994) Izbrannye trudy [Selected works]. Moscow: Gnosis. T. 2. *Language and culture*. URL: <https://biblioclub.ru/index.php?Page = book & id =219940>.

8. Cantor, N.E., Thomas, E.A. (2007) Control of attention in the processing of temporal and spatial information in complex visual patterns. NJ : *Journal of Experimental Psychology: Human Perception and Performance*. № 3. P. 243–250.
9. Cappelle, B. (2005) Particle Patterns in English. A Comprehensive Coverage. URL: <https://kulak.kuleuven.be/nl/onderzoek/Onderzoeksdomeinen/ language-literature-translation/letteren/onderzoektaalkunde/fest/descriptiveenglish grammar/bert-cappelle/bert-cappelle-proefschrift.pdf>.
10. Dubose, N. (1985) *Models of the Visual Cortex*. New York : Wiley. 374 p.
11. Feenberg, A. (2002) *Transforming Technology: A Critical Theory Revised*. URL: [http://lst-iiiep.iiiep-unesco.org/cgi-bin/wwwi32.exe/\[in=epidoc1.in\]/?t2000=0 14688/\(100\)](http://lst-iiiep.iiiep-unesco.org/cgi-bin/wwwi32.exe/[in=epidoc1.in]/?t2000=0 14688/(100)).
12. Sabrina, A. (2019) Huang and Jeremy Bailenson. Ch. 4: Close Relationships and Virtual Reality. URL: <https://vhil.stanford.edu/mm/ 2019/01/huang-close-relationships-and-vr.pdf>.
13. Souza, C. (2004) A framework for analyzing and understanding online communities. *J Preece*. P. 579–610.