

ФОРМУВАННЯ ЗМІСТУ ПРОФЕСІЙНОЇ ПІДГОТОВКИ МАЙБУТНІХ ІНЖЕНЕРІВ-ПРОГРАМІСТІВ У ГАЛУЗІ ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ

Сіциліцин Ю. О.

PhD, старший викладач кафедри інформатики і кібернетики

Мелітопольський державний педагогічний університет

імені Богдана Хмельницького

вул. Гетьманська, 20, Мелітополь, Запорізька область, Україна

orcid.org/0000-0002-3888-5575

yurarud@gmail.com

Ключові слова: паралельні обчислення, паралельне програмування, професійна підготовка, навчання паралельного програмування, технології паралельного програмування.

У статті описується формування змісту професійної підготовки майбутніх інженерів-програмістів у галузі паралельних обчислень за матеріалами іноземних фундаментальних робіт. Автор висвітлює проблеми, які виникають у разі формування змісту професійної підготовки майбутніх інженерів-програмістів у галузі паралельних обчислень. Складність формування змісту професійної підготовки майбутніх інженерів-програмістів у галузі паралельних обчислень пояснюється постійним швидким розвитком теоретичної бази використання паралельних обчислень та інструментальних засобів розробки паралельних і розподілених програм. Для формування змісту професійної підготовки майбутніх інженерів-програмістів з паралельних і розподілених обчислень були використані такі основні джерела, як: рекомендації щодо складу предметної галузі паралельних і розподілених обчислень проекту ACM і IEEE-CS, рекомендації щодо складу предметної галузі паралельних і розподілених обчислень за підтримки наукового фонду NSF та комітету IEEE-TCPP, фундаментальні праці з паралельних обчислень. На основі описаних вище джерел виокремлено п'ять основних галузей знань, які охоплюють весь спектр проблематики паралельних обчислень: математичні основи паралельних обчислень, паралельні обчислювальні системи (комп'ютерні основи), технології паралельного програмування, паралельні алгоритми розв'язання задач, паралельні обчислення, великі завдання і конкретні предметні галузі. Був сформований зведений зміст професійної підготовки у вигляді переліку тематичних розділів з паралельних обчислень, у якому вказується наявність або відсутність відповідної теми. В результаті аналізу парадигм глобальної комп'ютерної освіти від ACM за декілька років та матеріалів фундаментальних праць закордонних авторів був сформований зміст професійної підготовки у вигляді списку тем для навчання майбутніх інженерів-програмістів паралельних обчислень, а також рекомендована послідовність їх вивчення у вигляді траєкторії навчання.

FORMATION OF THE CONTENT OF PROFESSIONAL TRAINING OF FUTURE SOFTWARE ENGINEERS IN THE FIELD OF PARALLEL COMPUTING

Sitsylitsyn Yu. O.

*PhD, Senior Lecturer at the Department of Computer Science and Cybernetics
Bogdan Khmelnsky Melitopol State Pedagogical University
Hetmanska str., 20 Melitopol, Zaporizhzhia region, Ukraine
orcid.org/0000-0002-3888-5575
yurarud@gmail.com*

Key words: *parallel calculations, parallel programming, professional training, training in parallel programming, parallel programming technologies.*

The article describes the formation of the content of professional training of future software engineers in the field of parallel computing based on foreign fundamental works. The author highlights the problems that arise in shaping the content of professional training of future software engineers in the field of parallel computing. The difficulty of forming the content of professional training of future software engineers in the field of parallel computing is due to the constant rapid development of the theoretical basis for the use of parallel computing and tools for developing parallel and distributed programs. The following main sources were used to form the content of professional training of future software engineers in parallel and distributed computing: recommendations on the subject area of parallel and distributed computing of the ACM and IEEE-CS project, recommendations on the subject area of parallel and distributed computing with the support of NS. and the IEEE-TCPP Committee, fundamental work on parallel computing. Based on the sources described above, five main areas of knowledge were identified, covering the full range of parallel computing issues: mathematical foundations of parallel computing, parallel computing systems (computer basics), parallel programming technologies, parallel problem-solving algorithms, parallel computing, big tasks and specific subject areas. A consolidated content of professional training was formed in the form of a list of thematic sections on parallel calculations, which indicates the presence or absence of the relevant topic. ACM's analysis of the paradigms of global computer education over the years and the work of foreign authors has led to a list of topics for training future software engineers in parallel computing and a recommended sequence of study paths.

Постановка проблеми. Одним з методів підвищення конкурентоспроможності ІТ-підприємств є зниження собівартості розробки програмного забезпечення та підвищення його ефективності. Слід відзначити, що розробка програм, які використовують тільки одне ядро процесора, спираючись на класичні послідовні алгоритми, не дає необхідного приросту продуктивності порівняно з тими, що використовують у своєму коді паралельні багатоядерні рішення.

Поточний рівень розвитку суперкомп'ютерних технологій на основі паралельних обчислень, розповсюдження багатоядерних процесорів зумовлює актуальність вивчення паралельного програмування. Обчислювальна паралельність діє в різних специфічних формах залежно від фази програмування, складності паралельних фрагментів і характеру зав'язків між ними. Паралельне

програмування включає всі особливості класичного послідовного.

У процесі вивчення паралельної технології програмування формується паралельний стиль мислення, що передбачає наявність здатностей: до попереднього умоглядного «розпаралелювання» поставленого завдання – його аналізу з метою виділення підзадач, які можуть виконуватися паралельно; до «розпаралелювання» потоків даних – виділення потоків даних, якими будуть обмінюватися підзадачі, що виконуються паралельно; утримування в пам'яті дій усіх підзадач у певний проміжок часу для правильного керування їхньою спільною роботою.

За досить високого рівня розвитку паралельного стилю мислення інженер-програміст здатний передбачити специфічні проблеми, що виникають у роботі паралельних алгоритмів, проявляються

не кожен раз і суттєво ускладнюють налагодження програм. Лише досягнення певного рівня розвитку паралельного стилю мислення дозволить інженеру-програмісту ефективно реалізовувати паралельні обчислення. Отже, освоєння паралельного програмування може бути успішним за умови зміни характеру розумової діяльності, формування паралельного стилю мислення, що своєю чергою допоможе формуванню професійної підготовки майбутніх інженерів-програмістів у галузі паралельних обчислень.

Для досягнення цієї мети необхідно спроектувати зміст професійної підготовки у вигляді списку тем для навчання майбутніх інженерів-програмістів паралельних обчислень, а також рекомендовану послідовність їх вивчення.

Натепер є декілька фундаментальних робіт, у яких розглядається зміст професійної підготовки, але вони мають між собою багато розбіжностей.

Мета статті – сформулювати зміст професійної підготовки майбутніх інженерів-програмістів у галузі паралельних обчислень за матеріалами іноземних фундаментальних робіт.

Аналіз досліджень та публікацій. Фундаментальні дослідження у галузі підготовки майбутніх інженерів-програмістів у закладах вищої освіти здійснюють В. Биков, В. Осадчий, З. Сейдаметова, О. Спирін, Ю. Триус. Також науковці висвітлюють окремі аспекти зазначених проблем, зокрема: питання якості професійної підготовки програмістів (А. Власюк, В. Круглик, К. Осадча, А. Стрюк); вимоги до професійних якостей майбутніх інженерів-програмістів (Н. Длугунович, А.Л. Ко, P.L. Li, J. Zhu); організацію навчання майбутніх інженерів-програмістів у вищих інших країнах світу – В. Круглик. Огляд та вибір засобів паралельного програмування здійснили С. Лупін, М. Посипкін, В. Воеводін, Е. Уїльямс та інші вчені. Н. Скопін досліджує навчання паралельного програмування, науковець А. Шеметова робить огляд методичних прийомів навчання паралельного програмування.

Складність такого аналізу змісту професійної підготовки майбутніх інженерів-програмістів у галузі паралельних обчислень пояснюється постійним швидким розвитком теоретичної бази використання паралельних обчислень та інструментальних засобів розробки паралельних і розподілених програм.

Виклад основного матеріалу. Проблематика паралельних обчислень має відношення перш за все до однієї зі складових частин комп'ютерної науки (Computer Science) [2; 1; 4]. Вперше аспекти паралелізму стали згадуватися в редакції Computing Curricula від 2001 р., але і в 2001 р. [3], і в 2008 р. [2] паралелізм не було виділено в окрему галузь знань: проблематика паралельних обчислень включалася як окремі

розділи в декількох різних частинах рекомендацій з вивчення комп'ютерних наук. І тільки в 2013 р. була розроблена перша редакція рекомендацій з вивчення комп'ютерних наук, де створено окрему галузь знань «Паралельні і розподілені обчислення» (Parallel and Distributed Computing) [1]. У рекомендаціях 2020 р. зміст розділів галузі знань «Паралельні і розподілені обчислення» було суттєво оновлено.

Для формування змісту професійної підготовки майбутніх інженерів-програмістів з паралельних і розподілених обчислень доцільно використовувати такі основні джерела:

1) Рекомендації щодо складу предметної галузі паралельних і розподілених обчислень (Зміст 1), що підготовлені в межах спільного проєкту ACM і IEEE-CS з розробки навчальних планів галузі комп'ютеринг [4].

2) Рекомендації щодо складу предметної галузі паралельних і розподілених обчислень (Зміст 2), що розроблені в межах проєкту, який виконувався за підтримки наукового фонду NSF (США) та комітету IEEE-TCPP [6].

Сутність підходу до визначення змісту професійної підготовки з паралельних і розподілених обчислень полягає в такому:

1. Набір знань і умінь, який повинен засвоїти фахівець для успішної професійної діяльності, визначається набором окремих галузей знань, що являють собою окремі частини відповідної сфери діяльності.

2. Далі галузі знань діляться на менші структури, звані розділами, які являють собою окремі тематичні модулі всередині галузі.

3. Кожен розділ своєю чергою складається з набору тем, що являють собою нижній рівень цієї ієрархії у відповідній сфері діяльності. Кожна тема супроводжується вказівкою, чи є вона обов'язковою або факультативною, а також рекомендованим обсягом навчального часу, необхідним для її вивчення.

Важливо відзначити, що подібна структура галузей, розділів і тем визначає набір знань, необхідний для освоєння відповідної сфери діяльності, а не перелік навчальних курсів. Навчальні плани й відповідні навчальні курси розробляються далі вже на основі змісту.

Спроектуємо зміст професійної підготовки майбутніх інженерів-програмістів у галузі паралельних обчислень, використовуючи документи, вказані вище як Зміст 1 [4] та Зміст 2 [6] для визначення предметної галузі паралельних обчислень, розглянемо фундаментальну працю з паралельних обчислень «Структурне паралельне програмування: шаблони ефективних обчислень» (автори: М. МакКул, Дж. Рейндерс, А. Робісон) [5]. Обозначимо цю працю як Зміст 3.

На верхньому рівні Змісту виокремлено п'ять основних галузей знань, які охоплюють весь спектр проблематики паралельних обчислень:

- 1) математичні основи паралельних обчислень;
- 2) паралельні обчислювальні системи (комп'ютерні основи);
- 3) технології паралельного програмування;
- 4) паралельні алгоритми розв'язання задач;
- 5) паралельні обчислення, великі завдання і конкретні предметні галузі.

Представимо зміст професійної підготовки у вигляді переліку тематичних розділів (табл. 1). У правих стовпцях таблиці вказується наявність (+) або відсутність (-) відповідних тем Змісту в альтернативних розробках, тобто у Змісті 1 [4] і у Змісті 2 [6] відповідно. Відмітка «+/-» означає, що

тема представлена значною мірою, відмітка «-/+» говорить про те, що тема розкривається частково.

Розглянемо складники змісту професійної підготовки з паралельних обчислень, що запропоновані Змістом 1 (Рекомендації ACM 2020) [4], Змістом 2 (Рекомендації проекту NSF / IEEE-TCPP) [6] та Змістом 3 (М. МакКул, Дж. Рейндерс, А. Робісон) [5].

У Змісті 3 [5] науковці виділяють такі методи та технології розробки паралельних програм, як:

- 1) традиційні мови програмування та компілятори, що можуть розпаралелювати. Векторизація програм;
- 2) програмні бібліотеки для розробки паралельних програм: Shmem, Linda, MPI, PVM, Intel TBB (Thread Building Blocks), Microsoft TPL (Task Parallel Library);

Таблиця 1

Зведений зміст професійної підготовки з паралельних обчислень

№	Галузь знань, розділ	Зміст 1	Зміст 2
1	Математичні основи паралельних обчислень		
1.1	Графіки моделей програм	-	-
1.2	Поняття необмеженого паралелізму	-	-
1.3	Тонка інформаційна структура програм	-	-
1.4	Еквівалентні перетворення програм	-	-
1.5	Моделі обчислень для комп'ютерних систем	+	+
1.6	Математичні моделі паралельних обчислень	-/+	-/+
2	Паралельні обчислювальні системи (комп'ютерні бази)		
2.1	Основи машинного комп'ютера	-	+
2.2	Основи побудови комп'ютерних систем	-/+	+/-
2.3	Паралельні обчислювальні системи	-/+	+/-
2.4	Багатопроцесорні обчислювальні системи	+/-	+/-
2.5	Багатопроцесорні обчислювальні системи зі спільною пам'яттю	-/+	+
2.6	Багатопроцесорні обчислювальні системи з розподіленою пам'яттю	+	+
2.7	Графічні процесори	-/+	-
2.8	Обчислювальні системи транспетафлопу та ексмасштабних характеристик	-	-
2.9	Розподілені обчислювальні системи	+	-/+
2.10	Виклики суперкомп'ютерів і дата-центру	-	-
3	Технології паралельного програмування (основи програмної інженерії)		
3.1	Загальні принципи паралельного програмування	-/+	-/+
3.2	Основи паралельного програмування	-/+	+/-
3.3	Методи і технології розробки паралельних програм	-	+/-
3.4	Паралельні проблемно-орієнтовані бібліотеки та програмні пакети	-	-
3.5	Інструменти для паралельної розробки програм	-	-
3.6	Методи підвищення ефективності паралельних програм	-	-
4	Паралельні алгоритми вирішення задач		
4.1	Загальні принципи розробки паралельного алгоритму	-/+	+/-
4.2	Навчальні алгоритми паралельного програмування	-	+
4.3	Паралельні алгоритми матричних обчислень	-	+/-
4.4	Паралельні алгоритми сортування та пошуку даних	-/+	+
4.5	Алгоритми паралельної обробки графів	-/+	+
4.6	Паралельні алгоритми розв'язку диференціальних рівнянь у часткових похідних	-	-
4.7	Паралельні алгоритми вирішення задач оптимізації	-	-
4.8	Паралельні алгоритми Монте-Карло	-	-
4.9	Паралельні алгоритми для інших класів обчислювально-трудомістких задач	-	-

3) надмовні засоби для організації паралелізму: Cray Fortran, DVM, OpenMP, HPF, Cilk;

4) паралельні розширення традиційних мов програмування: CAF, UPC;

5) паралельні мови програмування: Occam, SISAL, HOPMA;

6) паралельні мови програмування для систем з розподіленою спільною пам'яттю на основі моделі розділеного глобального адресного простору (partitioned global address space, PGAS): Chapel, X10;

7) паралельні мови програмування для графічних процесорів: CUDA, OpenCL;

8) функціональні мови паралельного програмування: Parallel Haskell, Parlog, Erlang, T-Система;

9) засоби і технології для підтримки метакомп'ютингу і розподілених обчислень: Globus, gLite, UNICORE, BOINC, X-Com, Map / Reduce;

10) технології програмування FPGA-комп'ютерів;

11) автоматизація розпаралелювання і оптимізації програм;

12) елементи схемотехніки, мови опису електронних схем, VHDL.

Перелік розділів галузі знань «Паралельні і розподілені обчислення» за матеріалами досліджень [4; 5] наведено в табл. 2.

Виконане порівняння (табл. 2) показує, що рекомендації 2020 р. [4] практично повністю перекриваються варіантами з двох інших розглянутих підходів. Можна відзначити, що в рекомендаціях [4] більш ґрунтовно розкрито розділи, що пов'язані з розподіленими системами, хмарними обчисленнями та формальними моделями і семантикою.

Визначення предметної галузі паралельних і розподілених обчислень було виконано також у межах проекту, підтриманого науковим фондом NSF (США) і комітетом IEEE-TCPP [6]. Перший варіант рекомендацій був підготовлений у 2010 р., остання робоча редакція була опублікована у 2020 р.

Відповідно до цих рекомендацій у галузі паралельних і розподілених обчислень виділено такі чотири галузі знань, як:

- 1) архітектура (Architecture);
- 2) програмування (Programming);
- 3) алгоритми (Algorithms);
- 4) додаткові розділи (Cross Cutting and Advanced Topics).

Як можна помітити, виділені галузі знань багато в чому повторюють структуру змісту професійної підготовки, запропоновану у Змісті 3. Разом з цим дві галузі знань («Математичні основи» та «Паралельні обчислення, великі завдання і конкретні предметні галузі»), що наявні у Змісті 3, відсутні в рекомендаціях проекту NSF/IEEE-TCPP. З іншого боку, галузь «Додаткові розділи» розглянутих рекомендацій як така відсутня і розподілена в інших галузях знань Змісту 3.

У рекомендаціях проекту NS /IEEE-TCPP структурні елементи другого рівня – розділи – в явному вигляді не виділено. Фактично склад галузей знань визначається відразу в темах, що ускладнює використання такого визначення предметної галузі. В окремих випадках запропоновані теми об'єднано у групи, з використанням яких структура предметної галузі в межах підходу, який розглядається, має такий вигляд (табл. 3).

Спираючись на матеріали фундаментальних праць, охарактеризованих вище, можна сформулювати список тем для навчання майбутніх інженерів-програмістів паралельних обчислень, а також рекомендовану послідовність їх вивчення (рис. 1):

- поняття паралельних обчислювальних систем;
- види паралелізму в архітектурі EOM;
- параметри архітектури ПОС;
- параметри архітектури РОС;
- парадигми паралельного програмування;
- моделі складності та продуктивності паралельних обчислень;
- методи розробки паралельних алгоритмів;
- програмування паралельних процесів із загальною пам'яттю (паралельні обчислення);
- технологія OpenMP;
- технологія Parallel Virtual Machine;
- класи та бібліотеки мов програмування;

Таблиця 2

Розділи галузі знань «Паралельні і розподілені обчислення» (Зміст 1)

№	Галузь знань, розділ	Зміст 3	Зміст 1
1	Основи паралелізму (Parallelism Fundamentals)	+	+
2	Паралельне розкладання (Parallel Decomposition)	+	+
3	Зв'язок та координація (Communication and Coordination)	+	+
4	Паралельні алгоритми, аналіз і програмування (Parallel Algorithms, Analysis, and Programming)	+	+
5	Паралельна архітектура (Parallel Architecture)	+	+
6	Паралельна продуктивність (Parallel Performance)	+	+
7	Розподілені системи (Distributed Systems)	-/+	-/+
8	Хмарні обчислення (Cloud Computing)	-/+	-/+
9	Формальні моделі та семантика (Formal Models and Semantics)	+/-	-/+

Розділи галузі знань «Паралельні і розподілені обчислення» (Зміст 2)

№	Галузь знань, розділ	Зміст 3	Зміст 2
1	Архітектура		
1.1	Паралелізм даних та керування	+	-/+
1.2	Спільна та розподілена пам'ять	+	-/+
1.3	Ієрархія пам'яті	+	-/+
1.4	Показники ефективності	+	-/+
1.5	Представлення з рухомою комою	+	-
2	Програмування		
2.1	Паралельні парадигми програмного забезпечення	+	-/+
2.2	Проблеми семантики і правильності	+	+
2.3	Проблеми з продуктивністю	+/-	-/+
3	Алгоритми		
3.1	Моделі та оцінки складності	+	+/-
3.2	Алгоритмічні парадигми	+	-/+
3.3	Алгоритмічні задачі	+	-/+
4	Додаткові теми (немає тематичних груп)	-/+	-/+

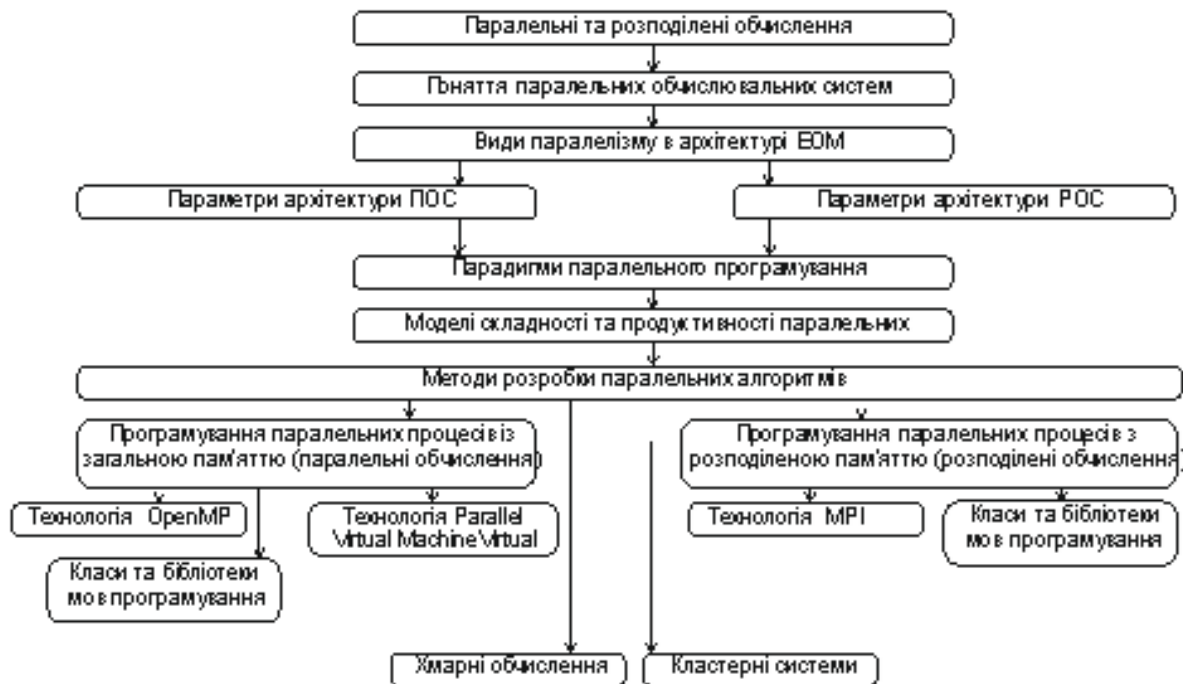


Рис. 1. Траєкторія навчання паралельних обчислень

- програмування паралельних процесів з розподіленою пам'яттю (розподілені обчислення);
- технологія MPI;
- класи та бібліотеки мов програмування;
- хмарні обчислення;
- кластерні системи.

Висновки та перспективи подальших розробок у цьому напрямку. В результаті аналізу пара-

дигм глобальної комп'ютерної освіти від ACM за декілька років та матеріалів фундаментальних праць закордонних авторів був сформований зміст професійної підготовки у вигляді списку тем для навчання майбутніх інженерів-програмістів паралельних обчислень, а також рекомендована послідовність їх вивчення у вигляді траєкторії навчання.

JIITEPATYPA

1. Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. ACM; IEEE; IEEE Computer Society. 2013. 515 p. DOI: 10.1145/2534860.
2. Computer Science Curriculum 2008: An Interim Revision of CS 2001. ACM; IEEE; IEEE Computer Society. 2008. 108 p. URL: <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/computerscience2008.pdf>.
3. Computing Curricula 2001. Computer Science ACM; IEEE Computer Society Association for Computing Machinery. 2001. 240 p. URL: <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2001.pdf>.
4. Computing Curricula 2020: Paradigms for Global Computing Education. New York : Association for Computing Machinery, 2020. 205 p. URL: <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2020.pdf>.
5. McCool M., Reinders J., Robison A. Structured Parallel Programming: Patterns for Efficient Computation. Waltham : Elsevier, 2012. 400 p.
6. NSF/IEEE-TCPP Curriculum Initiative on Parallel and Distributed Computing – Core Topics for Undergraduates (Version 2.0-beta). URL: <https://tcpp.cs.gsu.edu/curriculum/>.

REFERENCES

1. Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. ACM; IEEE; IEEE Computer Society. 2013. 515 p. DOI: 10.1145/2534860.
2. Computer Science Curriculum 2008: An Interim Revision of CS 2001. ACM; IEEE; IEEE Computer Society. 2008. 108 p. Retrieved from: <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/computerscience2008.pdf>.
3. Computing Curricula 2001. Computer Science ACM; IEEE Computer Society Association for Computing Machinery. 2001. 240 p. Retrieved from: <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2001.pdf>.
4. Computing Curricula 2020: Paradigms for Global Computing Education. New York: Association for Computing Machinery, 2020. 205 p. Retrieved from: <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2020.pdf>.
5. McCool M., Reinders J., Robison A. Structured Parallel Programming: Patterns for Efficient Computation. Waltham: Elsevier, 2012. 400 p.
6. NSF/IEEE-TCPP Curriculum Initiative on Parallel and Distributed Computing – Core Topics for Undergraduates (Version 2.0-beta). Retrieved from: <https://tcpp.cs.gsu.edu/curriculum/>.