



**XIII МІЖНАРОДНА НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ**

**МАТЕМАТИКА.  
ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ.  
ОСВІТА**

**ЛУЦЬК-СВІТЯЗЬ**

**31 травня - 2 червня 2024 р.**

**Тези доповідей**  
(друкуються в авторській редакції)

**XIII МІЖНАРОДНА НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ**

**МАТЕМАТИКА.  
ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ.  
ОСВІТА**

**ЛУЦЬК-СВІТЯЗЬ**

**31 травня – 2 червня 2024 р.**

**ТЕЗИ ДОПОВІДЕЙ**

(друкуються в авторській редакції)

**Луцьк – 2024**

## ОРГАНІЗАТОРИ:

**Волинський національний університет імені Лесі Українки**  
*Кафедра загальної математики та методики навчання інформатики*

**Київський національний університет імені Тараса Шевченка**  
*Кафедра інтегральних та диференціальних рівнянь*

**Національний університет «Львівська політехніка»**  
*Кафедра інформаційних систем та мереж*

## ПРОГРАМНИЙ КОМІТЕТ КОНФЕРЕНЦІЇ:

### ГОЛОВА:

1. **Сергієнко Іван Васильович**, доктор фізико-математичних наук, професор, академік НАН України, директор Інституту кібернетики імені В.М.Глушкова НАН України.

### ЧЛЕНИ ПРОГРАМНОГО КОМІТЕТУ:

2. **Биков Валерій Юхимович**, доктор технічних наук, професор, академік НАПН України, Почесний директор ІЦО НАПН України.

3. **Безущак Оксана Омелянівна**, доктор фізико-математичних наук, професор, декан механіко-математичного факультету Київського національного університету імені Тараса Шевченка, заслужений працівник освіти України.

4. **Берко Андрій Юліанович**, доктор технічних наук, професор кафедри інформаційних систем та мереж Інституту комп'ютерних наук та інформаційних технологій Національного університету «Львівська політехніка», заступник завідувача кафедри з повної освіти.

5. **Войтович Ігор Станіславович**, доктор педагогічних наук, професор, завідувач кафедри інформаційно-комунікаційних технологій та методики викладання інформатики Рівненського державного гуманітарного університету.

6. **Голодюк Лариса Степанівна**, доктор педагогічних наук, доцент, заступник директора з науково-методичної діяльності комунального закладу «Кіровоградський обласний інститут післядипломної педагогічної освіти імені Василя Сухомлинського».

7. **Камінський Роман Миколайович**, доктор технічних наук, професор кафедри систем штучного інтелекту Інституту комп'ютерних наук та інформаційних технологій Національного університету «Львівська політехніка».

8. **Капустян Олексій Володимирович**, доктор фізико-математичних наук, професор, завідувач кафедри інтегральних та диференціальних рівнянь Київського національного університету імені Тараса Шевченка.

9. **Король Ігор Іванович**, доктор фізико-математичних наук, професор, проректор з науково-педагогічної роботи ДВНЗ «Ужгородський національний університет».

10. **Кунанець Наталія Едуардівна**, доктор наук із соціальних комунікацій, професор кафедри інформаційних систем та мереж Інституту

**СЕКЦІЯ II**  
**ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ**

<b>Khomyak M., Melnyk Ar.</b> Data visualization in economic research using power BI .....	<b>74</b>
<b>Nyzhnyk O. I.</b> Self-driving car simulation in a virtual world .....	<b>76</b>
<b>Polishchenko R. P.</b> Overview of methods for visual navigation for autonomous vehicles.....	<b>77</b>
<b>Shakhmatov I., Zamrii Ir.</b> Application of graph neural networks for effective automation of Web spam filtering.....	<b>79</b>
<b>Аверічев І. М., Левкуша О. В.</b> Особливості міграції даних з баз даних.....	<b>81</b>
<b>Ананченко О. Є.</b> Методи забезпечення інформаційної безпеки адаптивних корпоративних освітніх інформаційних систем.....	<b>83</b>
<b>Антонюк Б. П.</b> Аналіз та використання патернів при проектуванні програмних систем .....	<b>85</b>
<b>Барабаш О. В., Мусієнко А. П., Макарчук А. В.</b> Знаходження умов виконання ймовірнісного критерію функціональної стійкості інформаційних систем .....	<b>87</b>
<b>Барабаш О. В., Свинчук О. В.</b> Забезпечення функціональної стійкості інтелектуальних систем об'єктів енергетики .....	<b>89</b>
<b>Богачевський В. В., Антонюк Б. П.</b> Основні складові відеоадаптера та їх властивості .....	<b>91</b>
<b>Бондарук А. С. , Жигаревич О. К.</b> Використання штучного інтелекту для написання програмного коду .....	<b>93</b>
<b>Вовк П. Б.</b> Cisco umbrella cloud Web security.....	<b>94</b>
<b>Гангало І. М., Гончаренко Д. О.</b> Використання автоматизованої системи перевірки лабораторних робіт в сучасній системі освіти ..	<b>96</b>
<b>Глинчук Л. Я.</b> Аналіз онлайн-середовищ розробки для вивчення логічної мови програмування Prolog .....	<b>98</b>
<b>Гордієнко К. О., Герцюк М. М.</b> Актуальність автоматизованого тестування програмного забезпечення методами штучного інтелекту.....	<b>100</b>
<b>Гришанович Т. О.</b> Паралельне програмування в Java.....	<b>103</b>
<b>Жебка С. В., Жебка В. В.</b> Вибір алгоритму консенсусу для технології блокчейн .....	<b>104</b>
<b>Жигаревич О. К., Загура Ю. В.</b> Використання цифрових технологій для забезпечення кібербезпеки в Україні .....	<b>106</b>

# ПАРАЛЕЛЬНЕ ПРОГРАМУВАННЯ В JAVA

*Гришанович Т. О.*

*Волинський національний університет імені Лесі Українки*

Паралельне програмування – це процес розбиття складної задачі на менші та простіші завдання, які можуть бути виконані одночасно за допомогою кількох ресурсів комп'ютера. Ці завдання незалежні одне від одного, виконуються паралельно за допомогою різних комп'ютерів або кількох ядер, що присутні в центральному процесорі. Завдяки такому підходу збільшується швидкість розробки проєкту, а ймовірність виникнення помилок зменшується.

Розглянемо засоби мови програмування Java, що дозволяють організовувати паралельні обчислення. Стандартне середовище Java (Java SE) надає розробнику Fork/Join Framework. Цей фреймворк допомагає розробнику легко реалізувати паралельні обчислення в додатках. Складність організації такої архітектури полягає в тому, що розробнику слід вказати, яким саме чином програма розбивається на окремі завдання. Java runtime здійснює це розбиття, використовуючи агрегатні операції (aggregate operations).

Java.util.concurrent – це пакет, що відповідає за визначення Fork/Join Framework в Java. Пакет java.util.concurrent містить різні класи та інтерфейси, які допомагають розробнику у розробці паралельних програм за допомогою мови програмування Java. У паралельній обробці частина цих обчислень оптимізована для використання кількох процесорів одночасно. Щодо багатопотоковості, бездіяльний час одного процесора оптимізований на основі спільного часу. Розділені завдання великої програми об'єднуються в підкласі ForkJoinTask за допомогою їх абстрактних оголошень. Два абстрактні завдання підкласу Fork/Join – «Рекурсивне завдання» (Recursive Task) та «Рекурсивна дія» (Recursive Action).

Класи Fork/Join Framework:

1. Recursive Task: клас корисний тоді, коли виникає необхідність отримати результат виконання завдання. Наприклад, при сортуванні масиву великої розмірності результати кожного із підзавдань слід порівнювати одне з одним. Це абстрактне завдання є комплексним завданням, яке складно запрограмувати.

2. RecursiveAction: клас корисний тоді, коли розробник вирішує, що не слід повертати жодного результату після виконання програми. Наприклад, виникла необхідність ініціалізувати масив великої розмірності спеціальними значеннями. У такому випадку кожне із підзавдань буде працювати на своєму відрізку масиву. Для використання RecursiveAction створюється новий клас. Цей клас діє як дочірній клас, що походить від java.util.concurrent.RecursiveAction. Для виклику RecursiveAction необхідно реалізувати новий екземпляр RecursiveAction. Щойно створений екземпляр RecursiveAction буде викликано за допомогою ForkJoinPool.

3. ForkJoinTask: це абстрактний клас, який використовується для визначення завдання. По суті, завдання можна створити за допомогою методу

fork(), що належить до цього абстрактного класу. Це завдання легше, ніж потік, створений за допомогою класу Thread.

4. ForkJoinPool: клас надає розробнику загальний пул, який допомагає у виконанні завдання ForkJoinTask.

Методи Fork/Join Framework:

1. Compute(): Рекурсивний виклик виконується, коли викликається метод compute() у правильному завданні.

2. Fork(): коли використовується даний метод, шойно створена PrimeRecursiveAction додається до списку завдань для активного потоку.

3. Join(): виклик методу join() у розгалуженому завданні повинен бути одним із останніх кроків після використання інших методів.

Завдяки Fork/Join Framework розробники можуть ефективно використовувати потужності багатоядерних процесорів для паралельної обробки завдань, що дозволяє підвищити продуктивність програм та зменшити час виконання. Цей засіб спрощує розробку паралельних програм, надаючи потужні інструменти для розділення складних завдань на менші частини та їх паралельного виконання. Однак важливо враховувати, що ефективне використання цього фреймворку вимагає глибокого розуміння принципів паралельного програмування та оптимального розподілу завдань між потоками. Загалом, Fork/Join Framework є потужним інструментом, який допомагає розробникам створювати швидкі та ефективні програми, особливо у сферах, де важлива продуктивність та обробка великих обсягів даних.

#### **Список використаних джерел:**

1. Parallel Programming in Java - Javatpoint. [www.javatpoint.com](http://www.javatpoint.com). URL: <https://www.javatpoint.com/parallel-programming-in-java> (date of access: 3.05.2024).

## **ВИБІР АЛГОРИТМУ КОНСЕНСУСУ ДЛЯ ТЕХНОЛОГІЇ БЛОКЧЕЙН**

***Жебка С. В., Жебка В. В.***

*Державний університет інформаційно-комунікаційних технологій*

Через те, що всі вузли в блокчейні децентралізовані та працюють незалежно, підтримка консенсусу створює складну проблему, яку різні типи блокчейнів вирішують по-різному. Різні рішення підтримки консенсусу між вузлами надають нам технічні інновації для подальшого оновлення алгоритмів технології блокчейн.

Алгоритм консенсусу вважається більш безпечним, якщо він може забезпечити захист від різних типів загроз безпеці.

Є ще один критерій – енергоємність алгоритму. Якщо алгоритм споживає занадто багато енергії, його слід змінити та розглянути більш екологічні варіанти.