Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

**Комп’ютерний практикум № 1**

з дисципліни «Технології паралельних обчислень»

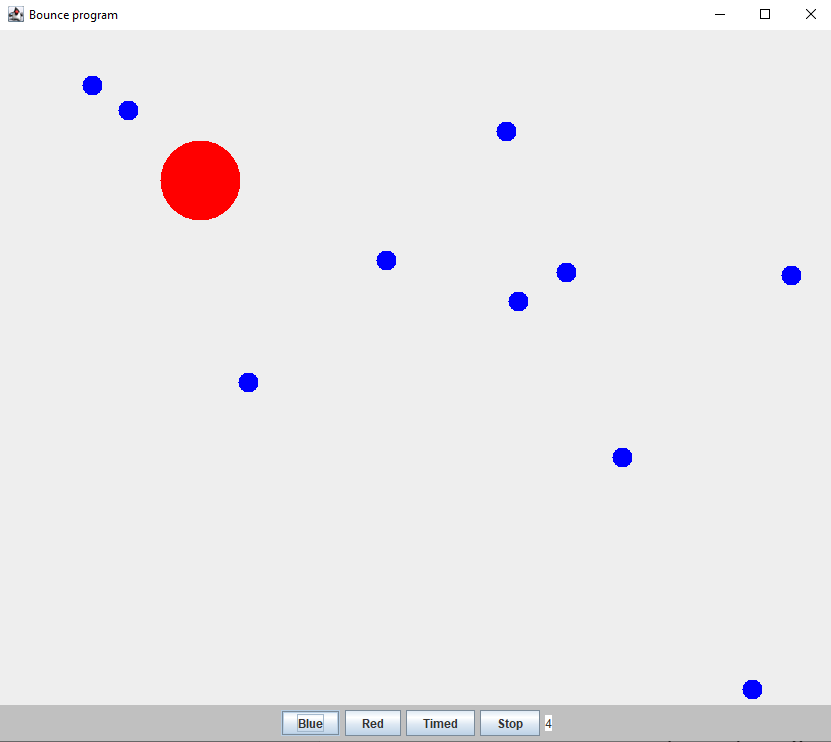
**Тема:** «Розробка потоків та дослідження пріоритету запуску потоків»

|  |  |
| --- | --- |
| **Виконав:**  студент групи ІТ-04  Гавриленко Ян  Дата здачі \_\_\_\_\_\_\_\_  Захищено з балом \_\_\_\_\_\_\_ | **Перевірила:**  ас. кафедри ІПІ  Дифучина О.Ю. |

Київ 2023

**Завдання:**

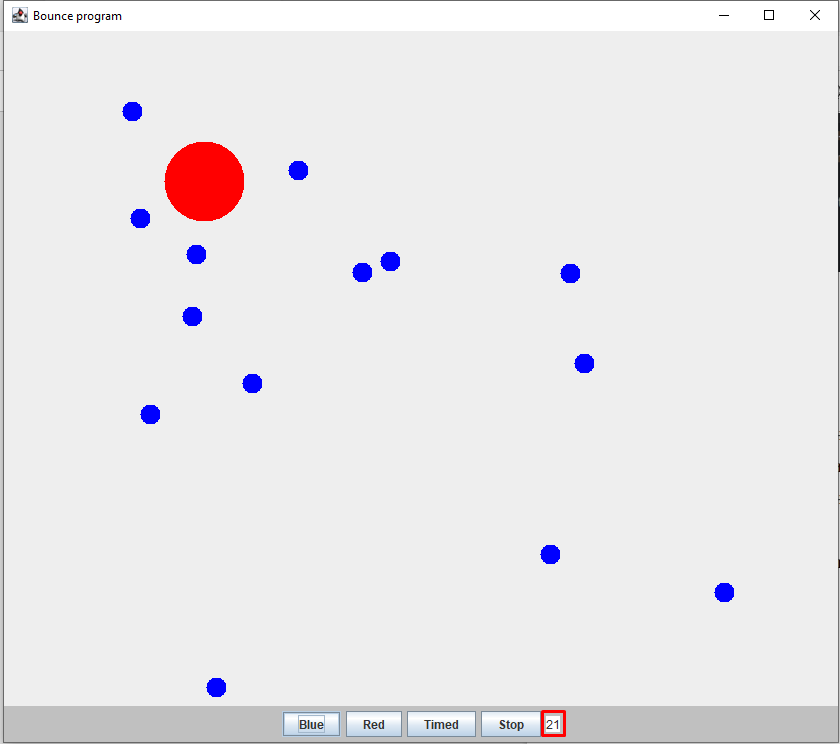
1. Реалізуйте програму імітації руху більярдних кульок, в якій рух кожної кульки відтворюється в окремому потоці (див. презентацію «Створення та запуск потоків в java» та приклад). Спостерігайте роботу програми при збільшенні кількості кульок. Поясніть результати спостереження. Опишіть переваги потокової архітектури програм. **10 балів.**

Скріншот програми: 

Результати спостереження: Створення та початок руху починається по натисканню кнопки, кульки рухаються та зникають одночасно з іншими та незалежно від них.

Переваги потоків: Переваги полягають в тому, що обчислення та оновлення даних про положення кульок може виконуватися на декількох потоках одночасно, паралелізуючи тим самим ці процеси.

1. Модифікуйте програму так, щоб при потраплянні в «лузу» кульки зникали, а відповідний потік завершував свою роботу. Кількість кульок, яка потрапила в «лузу», має динамічно відображатись у текстовому полі інтерфейсу програми. **10 балів.**

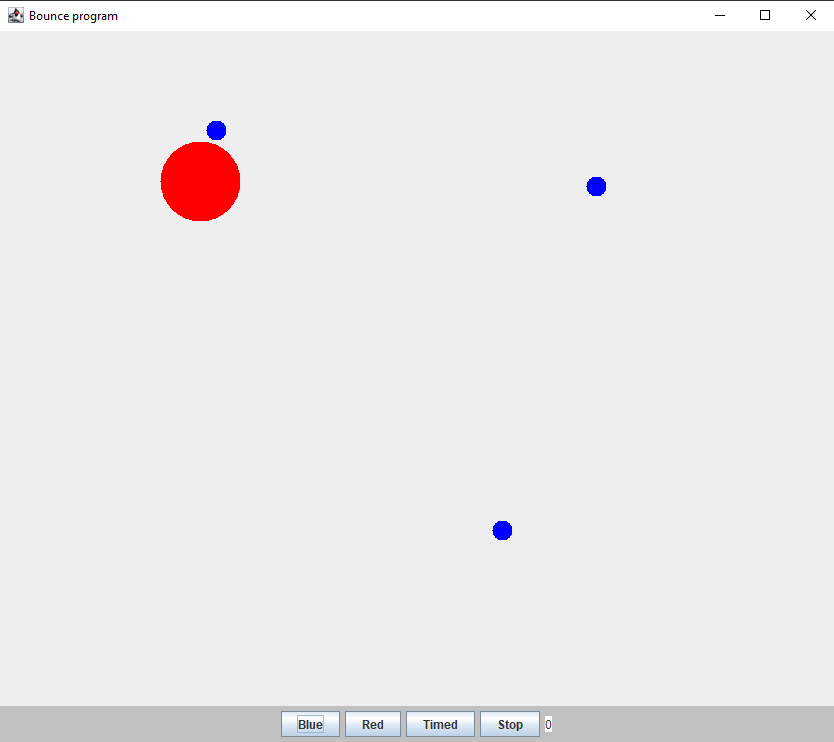


1. Виконайте дослідження параметру priority потоку. Для цього модифікуйте програму «Більярдна кулька» так, щоб кульки червоного кольору створювались з вищим пріоритетом потоку, в якому вони виконують рух, ніж кульки синього кольору. Спостерігайте рух червоних та синіх кульок при збільшенні загальної кількості кульок. Проведіть такий експеримент. Створіть багато кульок синього кольору (з низьким пріоритетом) і одну червоного кольору, які починають рух в одному й тому ж самому місці більярдного стола, в одному й тому ж самому напрямку та з однаковою швидкістю. Спостерігайте рух кульки з більшим пріоритетом. Повторіть експеримент кілька разів, значно збільшуючи кожного разу кількість кульок синього кольору. Зробіть висновки про вплив пріоритету потоку на його роботу в залежності від загальної кількості потоків. **20 балів.**

Висновок*:* при невеликій кількості кульок, їх рук відбувається без різниці, або мінімально відрізняється. При збільшенні ж кількості потоків, рух червоних кульок залишається приблизно незмінним, коли ж сині кульки можуть змінювати характер руху – відставати, пропускати деякі кадри зі змінами тощо. Це доводить те, що пріоритет виконання дій на потоках збільшує частку виконаних дій на них, відносно до інших потоків.

1. Побудуйте ілюстрацію для методу join() класу Thread з використанням руху більярдних кульок різного кольору. Поясніть результат, який спостерігається. **10 балів.**

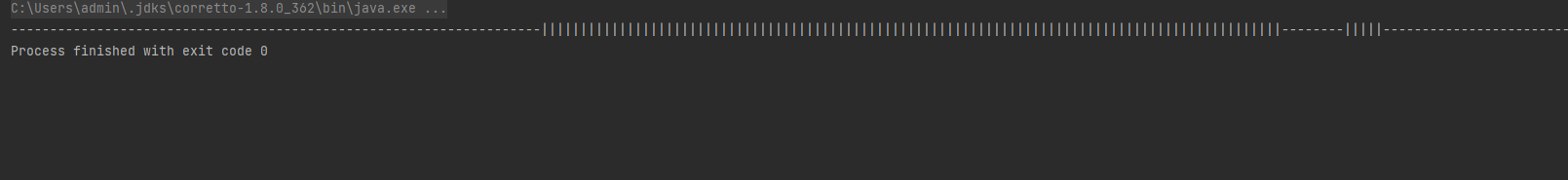
Ілюстрація програми:



Висновок: Головний поток чекає виконання кожного з потоків створених кульок, після чого запускає новий. Основним методом у цьому очікуванні є join.

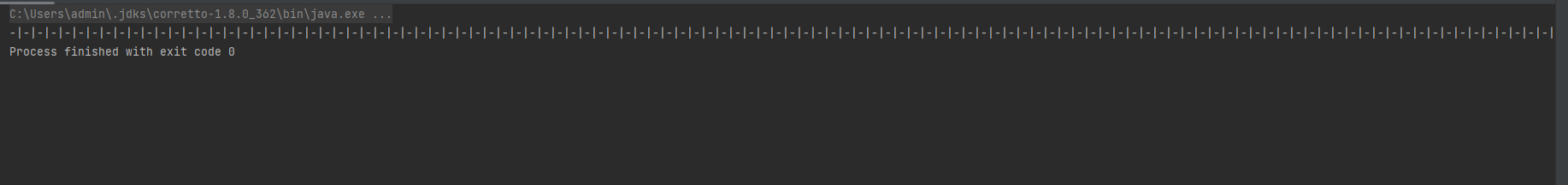
1. Створіть два потоки, один з яких виводить на консоль символ ‘-‘, а інший – символ ‘|’. Запустіть потоки в основній програмі так, щоб вони виводили свої символи в рядок. Виведіть на консоль 100 таких рядків. Поясніть виведений результат. **10 балів.** Використовуючи найпростіші методи управління потоками, добийтесь почергового виведення на консоль символів. **15 балів.**

Ілюстрація роботи:



Висновок: Обидва потоки правильно виконують свою роботу, виводячи в рядок два типи символів. Однак результат не є передбачуваним і може виводити символи в довільному порядку

Ілюстрація роботи після змін:



1. Створіть клас Counter з методами increment() та decrement(), які збільшують та зменшують значення лічильника відповідно. Створіть два потоки, один з яких збільшує 100000 разів значення лічильника, а інший – зменшує 100000 разів значення лічильника. Запустіть потоки на одночасне виконання. Спостерігайте останнє значення лічильника. Поясніть результат. **10 балів.** Використовуючи синхронізований доступ, добийтесь правильної роботи лічильника при одночасній роботі з ним двох і більше потоків. Опрацюйте використання таких способів синхронізації: синхронізований метод, синхронізований блок, блокування об’єкта. Порівняйте способи синхронізації. **15 балів.**

Виконання програми:

Зображення, що містить текст

Автоматично згенерований описЗображення, що містить текст

Автоматично згенерований опис

Висновки: Метод використання лічильнику без виконання синхронізації потоків призводить до непередбачуваного результату через неатомарність дій, які виконує потік. У той час, усі інші варіанти синхронізації в даному випадку показують себе однаково, стабілізуючи виконання.

Програмний код наведений в репозиторії:

<https://github.com/yan14171/parralel_calcullation_labs>

**Висновки:**

Після виконання лабораторної дійшов до висновків, що у багатьох випадках використання низькорівневих потоків для паралелізації обчислень може призвести до прискорення виконання програми. Однак, потрібно в ручну займатися кожним потоком окремо, витрачаючи рядки коду та час. Також, пріоритетність потоків може інколи змінити характер виконання багатопоточних програм, хоча і не дає відтворюваних та передбачуваних змін. Існують численні методи синхронізації, використання яких є обов’язковим для доступу до даних.