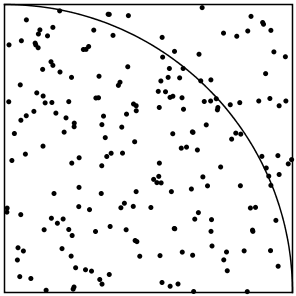
**Simple Tasks**

# Обчислення наближеного значення числа Пі методом Монте-Карло

У цьому завданні вам належить написати паралельну програму, яка обчислює значення числа Пі. Метод обчислення дуже простий:

* Площа квадрата одиничної довжини дорівнює 1
* Площа сектора 90 ° для одиничного кола: π/4
* «Кидаємо» величезну кількість випадкових точок в одиничний квадрат
* Рахуємо кількість точок, що потрапили в межі кола, тобто відстань від яких до (0,0) менше або дорівнює 1
* Частка точок, які потрапили в коло дорівнює наближеному значенню π/4

**Деталі реалізації**

Ваше завдання написати паралельну реалізацію (ParallelMonteCarloPi.java). При написанні програми дотримуйтесь інструкцій:

• Першим і єдиним вхідним аргументом програми є кількість потоків

• В результаті програма виводить наступні дані:

PI is 3.14221

THREADS 8

ITERATIONS 1,000,000,000

TIME 12.83ms

Крім написання програми і виведення результату подумайте над наступними питаннями:

• Як впливає кількість ітерацій (кинутих точок) на кінцевий результат?

• При однаковій кількості точок і випадковому зерні, як впливає на результат різну кількість потоків?

• Як кількість потоків впливає на продуктивність вашої програми? (Для явних результатів вам ймовірно знадобиться набагато більша кількість семплів (ітерацій).

**Middle Task**

# Виконати одне завдання на вибір

1. Напишіть консольний додаток для обчислення суми всіх елементів масиву (з 1\_000\_000 цілих елементів, значення яких генеруються випадковим чином в діапазоні від 0 до 100), використовуючи фреймворк ForkJoin. Для цього:

* створіть та ініціалізуйте масив розмірністю 1\_000\_000 елементів;
* опишіть рекурсивну задачу для поділу масиву на дві частини і виклик для кожної частини рекурсивного завдання для подальшого розподілу;
* продовжуйте розподіл масиву, поки в підмасиві не залишиться менше 20 елементів, і тоді обчисліть суму цих елементів.

1. Напишіть консольний додаток Java, який рекурсивно обробляє вказану директорію, запускаючи для кожної піддиректорії окремий потік, з використанням високорівневого інтерфейсу для доступу до ресурсів:

* знайти в заданій директорії текстові файли, і підрахувати в кожному файлі кількість слів, що починаються із заданої літери. В деякий файл записати назви таких файлів і кількість знайдених слів. Вивести в консоль вміст створеного файлу.