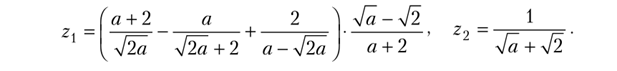
# Теоретическая часть

* Понятие о доступе к общему ресурсу из потока исполнения
* Синхронизация потоков при помощи блока синхронизации
* Синхронизация потоков при помощи метода синхронизации
* Синхронизация потоков при помощи интерфейса **Lock**, класса **ReentrantLocker**
* Взаимодействие потоков при помощи синхронизации: паттерн Производитель – Потребитель
* Взаимодействие потоков при помощи методов: wait(), notify(), notifyAll():
* паттерн Производитель – Потребитель
* Взаимодействие потоков при помощи объектов класса **ReentrantLocker**
* Взаимодействие потоков при помощи семафора
* Использование методов класса **Condition**: await(), signal(), signalAll() для взаимодействия потоков
* Семафоры, мьютексы для взаимодействия потоков, класс **Semaphore**

# Практическая часть

Напишите консольное Java приложение для решения следующих задач.

**Задача 1.** Реализация паттерна Производитель – Потребитель с использованием класса **ReentrantLocker**:

1. Производитель генерирует n вещественных случайных чисел (от 10 до 50) с диапазоном значений от -10 до 20.
2. Потребитель вычисляет z1 и z2 по выражению:  
     
   полученное значение и вычисленные результаты (или сообщение «ошибка вычислений») выводить в элемент управления окна.

Реализуйте три пары «Потребитель» – «Производитель», запускайте все три пары одновременно.

**Задача 2.** Реализация паттерна Производитель – Потребитель, для синхронизации использовать **одноканальный семафор**:

1. Есть текстовый файл, содержащий вещественные числа, записанные по одному в строке. В файле может быть от 12 до 18 чисел.
2. Один поток записывает в файл данные и ожидает, пока второй поток их прочитает и обработает. Поток извещает другой поток о завершении записи каждого числа. Поток завершает работу, когда количество чисел в файле станет больше 64
3. Второй поток ожидает, пока данные будут записаны в файл, читает файл и выводит числа в две строки – в первую строку в исходном порядке следования, во вторую строку нечетные (нечетная целая часть) в конце строки. Поток извещает другой поток о завершении чтения. Поток завершает работу, когда количество прочитанных и обработанных чисел станет больше 64.

# Дополнительно

Запись занятия можно скачать [**по этой ссылке**](https://cloud.mail.ru/public/hcvL/VkGYUETqK). Материалы занятия в этом же архиве.