**Задание 1**

**Класс Main**

Создает 2 потока с разницей во времени. В экземпляры класса PingPong создает, передавая в конструктор слово которое будет печатать поток (Ping или Pong) и личный номер потока (1 или 2).Создает Экземпляр класса Sharedresource;

Далее метод printWord

захватывает с помощью sinchronized общий ресурс (sharedresource) проверяет флаг sharedresource.isRead() (флаг отвечает прочитан ли рессурс см. подробнее в разделе Sharedresource) если sharedresource.isRead() возвращает false значит слово не выведено в консоль и он выводит в консоль результат работы метода **sharedresource**.getSayword()(то есть текущее слово которое было записанно в sayword) далее помечаем что слово прочитали

Если sharedresource.isRead() возвращает true тогда ни чего не выводим

Далее отпускаем общий ресурс

**Класс PingPong**

Класс реализует интерфейс Runnable

WORD- поле final слово которое пишем в общий ресурс (это либо Ping либо Pong) получаем через конструктор во время создания объекта в Main классе

PINGPONGINT- поле int является личным номер потока

в методе run

Захватывает с помощью sinchronized общий ресу рс(sharedresource) проверяет флаг sharedresource.isRead() если слово не прочитано main значит ни чего не пишем и отпускает монитор так же в этом же условии if проверяет кто последний записывал слово если метод sharedresource.getNumberLastThread() выдает его же личный номер. Значит он писал туда последним и должен ждать пока запишет в общий рессурс другой поток,вызывая метод wait перед этим пробуждая остальные потоки с помощью notify .

Если слово прочитано и sharedresource.getNumberLastThread() не равняется его личному номеру значит:

Записываем словоДалее помечаем что слово не прочитано и записываем свой личный номер с помощью метода sharedresource.getNumberLastThread() и отпускаем монитор.

**Класс SharedResource**

имеет поле sayword -хранит слово которое записывают 2 потока и читает Main

read -показывает выведено ли слово на консоль

numberLastThread- показывает кто последний записал слово в консоль

поля private поэтому использовал сеттеры и геттеры

**Задание 1а**

Использовал в место sinchronized ReentrantLock из пакета concurrent

Создал объект данного класса в main и передаю в конструкторе в классы PingPong. Те с помощью объекта ReentrantLock блокируют ресурсы