

**Задача.** Написать класс для управления доступом параллельных потоков к кольцевой очереди.

**Требования к реализации:**

1. Интерфейс класса для управления доступом параллельных потоков к кольцевой очереди:

```
class SyncQueue
{
public:
    // конструктор
    SyncQueue(int nSize); // nSize - размер очереди
    // деструктор
    ~ SyncQueue();
    // функции для доступа к очереди
    void Insert(int nElement); // добавить элемент в хвост очереди
    int Remove(); // удалить головной элемент очереди
};
```

2. Кольцевая очередь реализуется массивом, элементы которого имеют тип `int`. Размер кольцевой очереди задается в конструкторе.

3. Если поток вызывает метод `Insert`, а кольцевая очередь оказывается полной, то метод `Insert` должен переводить этот поток в состояние ожидания до удаления из кольцевой очереди, хотя бы одного элемента другим потоком.

4. Если поток вызывает метод `Remove`, а кольцевая очередь оказывается пустой, то метод `Remove` должен переводить этот поток в состояние ожидания до записи в кольцевую очередь, хотя бы одного нового элемента другим потоком.

**Дополнительные требования:**

Для тестирования класса `SyncQueue` написать программу для консольного процесса, который состоит из потока `main` и нескольких потоков `consumer` и `producer`.

Поток `main` должен выполнять следующие действия:

- создать объект кольцевой очереди, размер очереди вводится пользователем с клавиатуры;
- ввести с клавиатуры количество потоков `producer` и количество потоков `consumer`, которые он должен запустить;
- запросить для каждого из потоков `producer` и `consumer` количество целых чисел, которые эти потоки должны соответственно произвести и потребить;
- создать требуемое количество потоков `producer`, каждому потоку передать его порядковый номер и количество целых чисел, которые он должен произвести; создать требуемое количество потоков `consumer`, каждому потоку передать количество целых чисел, которые он должен потребить;
- подать сигнал на начало работы потоков `producer` и `consumer`;
- завершить свою работу после окончания работы всех потоков `producer` и `consumer`.

Поток `producer` должен выполнять следующие действия:

- ждать сигнал на начало работы;
- циклически выполнять следующие действия (количество циклов задается в параметре):
  - добавить в кольцевую очередь целое число, равное своему порядковому номеру;
  - вывести на консоль сообщение: "Произведено число: N ", где N - номер числа, помещенного в очередь.
  - поспать 7 мс.

Поток `consumer` должен выполнять следующие действия:

- ждать сигнал на начало работы;
- циклически извлекать из кольцевой очереди целые числа с интервалом в 7 мс (количество циклов задается в параметре);
- при извлечении числа из кольцевой очереди, выводить на консоль сообщение: "Употреблено число N ", где N - номер числа, извлеченного из очереди. **Факультативное**

**задание:** Обработать тупик, который возникает при несогласованности количества

произведенных и потребленных целых чисел.