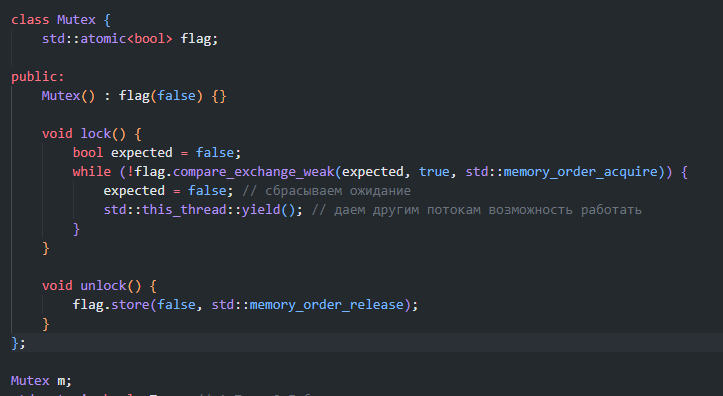
**Лабораторная №7**



Для реализации самодельного мьютекса, я использую класс Mutex, в котором я использую метод compare\_exchange.

С помощью flag(false) flag(true) обозначаем занят ли мьютекс или нет, изначально мьютекс всегда свободен(false).

Через !flag.compare\_exchange\_weak атомарную операцию проверяем текущее значение flag, если оно совпадает с expected, то есть false, то флаг обновляется на true. Так, здесь реализован метод m.lock. В случае если поток занят expected = false,

В void unlock() я использую flag.store, чтобы все действия, которые происходят после захвата мьютекса, видят изменения сделанные до этого в других потоках. Также устанавливаю флаг на false, чтобы освободить мьютекс.

