



Ву! ИИ готовит к ЕГЭ по информатике

[Попробовать](#)

## Урок Спринт 2.5

## RW mutex

В некоторых случаях, как в примере с кешем, можно выделить два типа критических секций: для чтения и для записи. Мы можем предположить, что чтение объекта из мапы несколькими горутинami не нарушит её целостность, и неплохо бы блокировать чтение только в момент записи другими горутинami. Для этого можно использовать `RMutex`:

```
type Cache struct {  
    m map[string]*Data  
    dr DataRetriever  
    mu sync.RWMutex  
}
```

Давайте посмотрим, как изменится код `Get`:

```
func (c *Cache) Get(ID string) (Data, bool) {  
    c.mu.RLock()  
    // теперь в эту секцию могут зайти несколько горутин  
    data, exists := c.m[ID]  
    c.mu.RUnlock()  
    // .... далее без изменений  
}
```

Теперь, код для получения объекта из мапы может выполняться из нескольких горутин, а когда нужно внести изменения, будет вызываться `c.mu.Lock()`, который не позволит читать данные из мапы до снятия блокировки `c.mu.Unlock()`.

[Справка](#)

Исключительное право на учебную программу и все сопутствующие ей учебные материалы, доступные в рамках сервиса, принадлежат АНО ДПО «Образовательные технологии Яндекса». Воспроизведение, копирование, распространение и иное использование программы и материалов допустимо только с предварительного письменного согласия АНО ДПО «Образовательные технологии Яндекса».

[Пользовательское соглашение.](#)

© 2018 – 2024 ООО «Яндекс»