



Bay! ИИ готовит к ЕГЭ по информатике

[Попробовать](#)

Урок Спринт 2.5

## Особенности использования

Мьютексы предоставляют удобный механизм синхронизации, однако надо учитывать ряд особенностей при работе с ними.

### Копирование мьютекса

Мьютексы не следует копировать. Когда вы копируете мьютекс, вы получаете два мьютекса: оригинал и копию, и блокировка оригинала никак не связана с блокировкой копий. Поэтому, если вы объявите метод `Get` следующим образом, это приведёт к нежелательным последствиям:

```
func (c Cache) Get(ID string) (Data, bool)
```

### Двойная блокировка

Мьютекс не отслеживает, какая горютина его заблокировала. Поэтому, двойная блокировка мьютекса из одной и той же горютины приведёт к такой ситуации, как блокировка этой горютины. Это распространённая проблема с несколькими функциями, которые могут вызывать друг друга, а также блокировать один и тот же мьютекс:

```
var m sync.Mutex

func first() {
    m.Lock()
    defer m.Unlock()
    second() // lock
}

func second() {
    m.Lock() // здесь будет вызов m.Lock() второй раз
    defer m.Unlock()
    // ... далее основное тело функции
}
```

В примере выше при вызове функции `second()` мы получим заблокированную горютину, потому что

к этому моменту уже был вызов `m.Lock()`.

## Использование из нескольких горутин

Так как мьютекс не отслеживает, какая горутина его заблокировала, может получиться так, что блокирует мьютекс одна горутина, а снимает блокировку — другая. Такое использование мьютекса является недопустимым и приводит к нежелательному поведению.

### 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»