

Напишите функцию `getFirstResult`, которая принимает контекст и запускает конкурентный поиск, возвращая первый доступный результат из `replicas`. Возвращать ошибку контекста, если контекст завершился раньше, чем стал доступен какой-то результат из реплики.

Напишите функцию `getResults`, которая запускает конкурентный поиск для каждого набора реплик из `replicaKinds`, используя `getFirstResult`, и возвращает результат для каждого набора реплик.

```
package main

import (
    "context"
    "fmt"
    "math/rand"
    "time"
)

type result struct {
    msg string
    err error
}
type search func() *result
type replicas []search

func fakeSearch(kind string) search {
    return func() *result {
        time.Sleep(time.Duration(rand.Intn(100)) * time.Millisecond)
        return &result{
            msg: fmt.Sprintf("%q result", kind),
        }
    }
}

func getFirstResult(ctx context.Context, replicas replicas) *result {
    // напишите ваш код здесь
}

func getResults(ctx context.Context, replicaKinds []replicas) []*result {
    // напишите ваш код здесь
}

func main() {
    ctx, _ := context.WithTimeout(context.Background(), 50*time.Millisecond)
    replicaKinds := []replicas{
        replicas{fakeSearch("web1"), fakeSearch("web2")},
        replicas{fakeSearch("image1"), fakeSearch("image2")},
        replicas{fakeSearch("video1"), fakeSearch("video2")},
    }
    for _, res := range getResults(ctx, replicaKinds) {
        fmt.Println(res.msg, res.err)
    }
}
```