



# Klobúky ako predjedlo

(čo to má s programovaním pochopíte dnes)



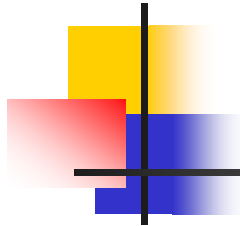
- 3 biele a 2 čierne
- A, B, C si navzájom vidia farby klobúkov
- nesmú komunikovať, ale (aj tak) sú inteligentní 😊
- vyhrávajú, ak **všetci** uhádnu farbu svojho klobúka
- resp. ak sa jeden pomýli, prehrali všetci.

Hint: A,B,C sú spoluhráči, preto predpokladaj,  
že sú chytrí a myslí aj za nich

Hint: úloha nie je o šťastí=hádaní správneho riešenia

## Algoritmus

1. Vidíš dve zelené hovoriš biela (výchlo)
  2. Vidíš jednu biela a jednu zelenú, čakáš či niekto niečo povie (biela) ak nie, si biela
  3. Vidíš dve biely a nikto nič nehovorí si biela ďalší povie biela (lebo vie, že ten, ktorý povedal, že je biela vidí dve biely) tretí človek vie, že keď druhý človek odpovedal výchlo je zelená ak pomaly je biela.
- 
1. Vidíš dve zelené kričíš (hred) biela. hred = po 1 sekunde.
  2. Vidíš jednu bielu a jednu zelenú, čakáš či niekto niečo povie, ak nie, si biela  
ak ano, si zelená
  3. Vidíš dve biely a nikto nič dlho nehovorí, si biela. dlho = 30 sekúnd  
ak hovorí, si zelená



# Do 10 sekúnd

---



ak vidím dva čierne, *určite mám biely*, a preto sa  
hneď ozvem, že "**mám biely**".

ak sa niekto do 10s ozval, že má biely, *musí vidieť  
dva čierne, preto ja mám čierny*,  
tak hneď kričím "**mám čierny**".

inak čakám 10s, *nikto neozval, že „mám biely”,  
preto určite nie sú v hre 2 čierne, ale najviac  
jeden čierny !!!*



# 10 až 20 sekúnd

---



*v hre je najviac jeden čierny*

ak teda vidím čierny, *ja musím mať biely*, tak sa ozvem  
hneď, že mám "**mám biely**".

inak, ak sa ozvú dvaja (do 10 s), že biely, *ja mám čierny*,  
tak kričím "**mám čierny**".



inak, *nevidím čierny a nikto sa neozval*, čakám ďalších 10s,



po 20 sekundách

---



*v hre nie je žiaden čierny*

*keďže sa nikto neozval, tak nie je žiaden čierny, tak kričím "**mám biely**" a ostatní tiež*



# Celý algoritmus

(bez vysvetlenia, už pre cvičenú opicu)

---

Hneď:

ak vidím dva čierne, hneď ozvem, že "**mám biely**".

ak sa niekto do 10 s ozval, tak kričím "**mám čierny**".

inak čakám 10s.

Po 10 sek:

ak vidím čierny, tak hneď kričím "**mám biely**".

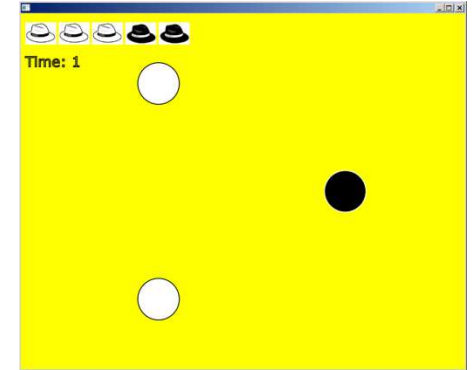
inak, ak sa ozvú dvaja do 10 s, tak kričím "**mám čierny**".

inak čakám ďalších 10s,

Po 20 sek:

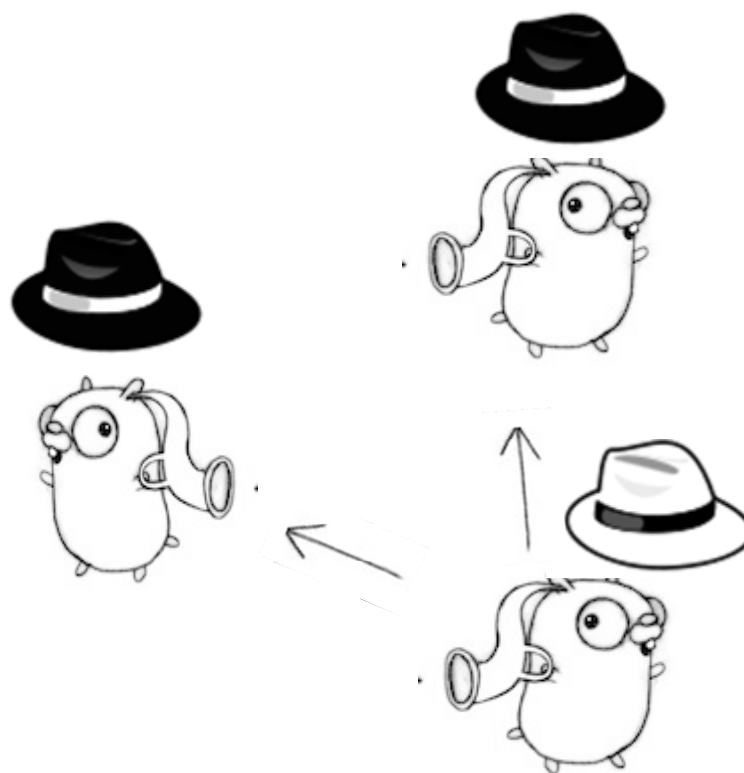
kričím "**mám biely**"

# Na zamyslenie



- je podstatné, či kričím *mám biely/mám čierny*, nestačí len **už viem** ?!
- Dalo by sa to pre 3 biele, 3 čierne, 3 ľudia ?
- Dalo by sa to pre 2 biele, 1 čierne, 2 ľudia ?
- Dalo by sa to pre N biele, (N-1) čierne, N ľudia ? (napr. 6,5,6)
- Dalo by sa to pre >N biele, (N-1) čierne, N ľudia ? (napr. 8,5,6)
- Dalo by sa to pre N biele, <(N-1) čierne, N ľudia ? (napr. 6,4,6)

# Komunikácia–každý s každým







# Správa, kanály, agenti

---

```
type Message struct {
    who int    // od koho, odosielateľ
    what int } // čo, obsah správy
func makeChannels(n int) []chan Message {
    chArray := make([]chan Message, n)
    for i:= 0; i < n; i++ { // kanál, na ktorom počúva i-ty agent
        chArray[i] = make(chan Message)
    }
    return chArray
}
func main() {
    chArray := makeChannels(numb)
    for a:= 0; a<numb; a++ {
        runAgent(a, chArray)
    } }
```



# Agenti napriamo

```
func runAgent(agent int, channels []chan Message) {
    go func() {          // ID agenta, kanály na všetkých agentov
        i := 1           // iniciálny stav agenta
        for {            // loop forever
            timeout := time.After(...)
            select {
                case msg := <- channels[agent]: // agent počúva len svoj
                    fmt.Printf("agentovi %d: prišla správa:%s", agent, msg)
                case <-timeout: // prešiel timeout, vyrobíme správu msg
                    msg := Message{who:agent, what:i++} //zmeníme svoj stav
                    for index, ch := range channels { // povedz každému
                        if index != agent {           // okrem seba
                            go func(ch chan Message) { // !!!!!!!!!!!!!!!!!!!!!
                                cha <- msg              // správu msg
                            }(ch)
                        }
                    }
                }
            }
        }
    }
}
```

[modelBezDispechera.go](#)



# Agenti napriamo

```
func runAgent(agent int, channels []chan Message) {
    go func() {          // ID agenta, kanaly na vsetkych agentov
        i := 1           // iniciálny stav agenta
        for {            // loop forever
            timeout := time.After(...)
            select {
                case msg := <- channels[agent]: // agent počúva len svoj
                    fmt.Printf("agentovi %d: prišla správa:%s", agent, msg)
                case <-timeout: // prešiel timeout, vyrobíme správu msg
                    msg := Message{who:agent, what:i++} //zmeníme svoj stav
                    for index, ch := range channels { // povedz každému
                        if index != agent {           // okrem seba
                            go func() { // ?????? ZLE ??????
                                ch <- msg           // správu msg
                            }()
                        }
                    }
                }
            }
        }
    }()
```

[modelBezDispechera.go](#)



# Príklad komunikácie 3 agentov

---

1: povedal 1

agentovi 2: prisla sprava:"1: povedal 1"

agentovi 0: prisla sprava:"1: povedal 1"

0: povedal 1

agentovi 2: prisla sprava:"0: povedal 1"

agentovi 1: prisla sprava:"0: povedal 1"

1: povedal 2

agentovi 2: prisla sprava:"1: povedal 2"

0: povedal 2

1: povedal 3

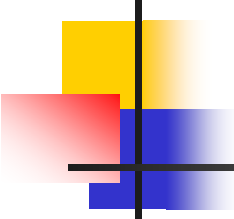
agentovi 0: prisla sprava:"1: povedal 2"

agentovi 0: prisla sprava:"1: povedal 3"

agentovi 1: prisla sprava:"0: povedal 2"

agentovi 2: prisla sprava:"0: povedal 2"

agentovi 2: prisla sprava:"1: povedal 3"



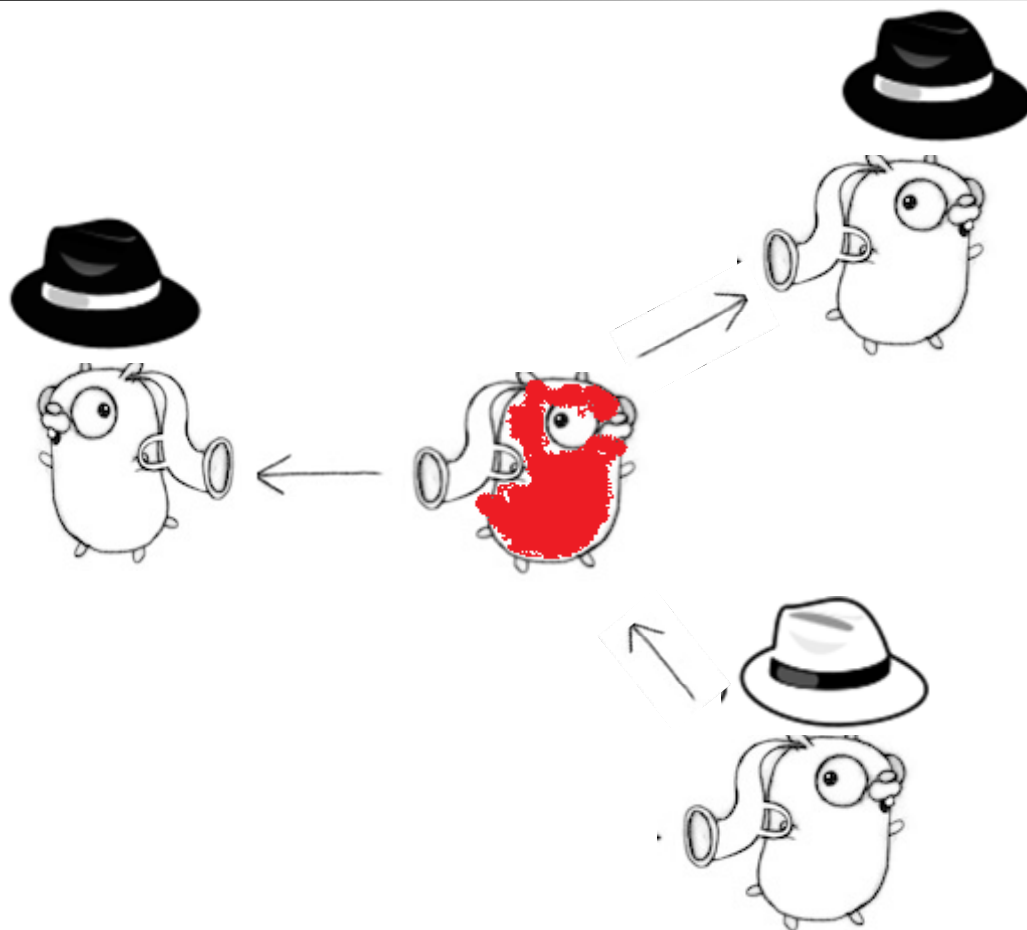
# Aplikácia na klobúky

## (domáca úloha)

```
func vidim(name String) (int, int) {
```

- [0s] A: vidim 1 biele a 1 cierne
- [0s] A: cakam 10 sek
- [0s] C: vidim 1 biele a 1 cierne
- [0s] C: cakam 10 sek
- [0s] B: vidim 2 biele a 0 cierne
- [0s] B: cakam 10 sek
  
- [10s] B: cakam dalsich 10 sek
- [11s] A: mam biely !!! true
- [11s] C: mam biely !!! true
- [11s] B:: prisla sprava, ze [11s] A: mam biely !!! true
- [12s] B: mam cierny !!! true
- finito

# Komunikácia s dispečerom





# Dispatcher

čo počujete to prepošle

```
func runDispatcher(channels []chan Message) chan Message {  
    dispch := make(chan Message)  
    // kanál na komunikáciu s dispatcherom  
    go func() {  
        for {  
            msg := <- dispch // ak prišla správa  
            fmt.Println("dispecer sa dozvedel: " + msg.toString())  
            for _,ch := range channels {  
                go func(x chan Message) {  
                    x <- msg  
                }(ch)  
            }  
        }  
    }()  
    return dispch }
```

[modelSDispecherom.go](#)



# Agenti cez dispečera

```
func runAgentCommunicatingWithDispatcher(agent int,
    dispch chan Message, input chan Message) {
    go func() {
        i := 1    // stav agenta
        for {
            timeout := time.After(...) // náhodny delay
            select {
                case msg := <- input: // ak prišla správa agentovi,
                    fmt.Printf("agentovi %d: prisla sprava:%s", agent, msg)
                case <-timeout: // po timeout, vytvoríme správu
                    msg := Message{who:agent, what:i}
                    dispch <- msg // pošleme dispecerovi
                    i++           // agent si zvýši stav
            }
        }
    }()
}
```





# Agenti cez dispečera

```
func runAgentCommunicatingWithDispatcher(agent int,
    dispch chan Message, input chan Message) {
    go func() {
        i := 0 // stav agenta
        for {
            timeout := time.After(...) // náhodny delay
            select {
                case msg := <- input: // ak prišla správa agentovi,
                    fmt.Printf("agentovi %d: prisla sprava:%s", agent, msg)
                case <-timeout: // po timeout, vytvoríme správu
                    msg := Message{who:agent, what:i}
                    go func() { dispch <- msg }() // pošleme dispecerovi
                    i++ // agent si zvýši stav
            }
        }
    }()
}
```



# Príklad komunikácie 3 agentov

---

1: povedal 0

dispecer sa dozvedel: 1: povedal 0

agentovi 2: prisla sprava:"1: povedal 0"

agentovi 0: prisla sprava:"1: povedal 0"

agentovi 1: prisla sprava:"1: povedal 0"

0: povedal 0

dispecer sa dozvedel: 0: povedal 0

agentovi 2: prisla sprava:"0: povedal 0"

agentovi 0: prisla sprava:"0: povedal 0"

agentovi 1: prisla sprava:"0: povedal 0"

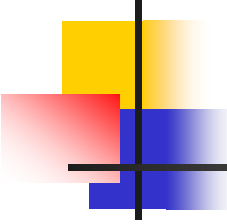
0: povedal 1

dispecer sa dozvedel: 0: povedal 1

agentovi 2: prisla sprava:"0: povedal 1"

agentovi 0: prisla sprava:"0: povedal 1"

agentovi 1: prisla sprava:"0: povedal 1"



# Aplikácia na klobúky

## (domáca úloha)

```
func vidim(name String) (int, int) {
```

- [0s] A: vidim 1 biele a 1 cierne
- [0s] A: cakam 10 sek
- [0s] B: vidim 2 biele a 0 cierne
- [0s] B: cakam 10 sek
- [0s] C: vidim 1 biele a 1 cierne
- [0s] C: cakam 10 sek
- [10s] B: cakam dalsich 10 sek
- [11s] A: mam biely !!! true
- od A prisla sprava, ze [11s] A: mam biely !!! true
- [11s] C: mam biely !!! true
- [11s] B:: prisla sprava, ze [11s] A: mam biely !!! True
- od C prisla sprava, ze [11s] A: mam biely !!! true
- od B prisla sprava, ze [11s] A: mam biely !!! true
- od B prisla sprava, ze [11s] C: mam biely !!! true
- [12s] B: mam cierny !!! True
- finito