

# B4B33RPH - závěrečná přednáška

Tomáš Svoboda, [svobodat@fel.cvut.cz](mailto:svobodat@fel.cvut.cz)

katedra kybernetiky, centrum strojového vnímání

8. ledna 2019

01

# Plán přednášky

- ▶ Obecně k předmětu, zpětná vazba, ...
- ▶ Vězňovo dilema (Otakar Jašek, Tomáš Svoboda)
- ▶ Reversi (Robert Pěnička)
- ▶ Spam (Tomáš Svoboda, Petr Pošík)
- ▶ Otázky, odpovědi, co dál ...

## Zpětná vazba ...

- ▶ je pro nás, tým RPH, velmi důležitá pro další rozvoj předmětu. Prosím vyplňte [anketu k předmětu](#) (je na webu cvičení).
- ▶ existuje i školní [všeobecná studentská anketa ČVUT](#) (<https://anketa.cvut.cz/>), kde je méně otázek, vyjadřujete se ke všem předmětům i k jednotlivým učitelům. Prosím najděte si čas a vyplňte i tuto. <http://www.fel.cvut.cz/cz/aktuality/anketa-zima-slosovani.html>
- ▶ ... reakce podepřené argumenty mají mnohem větší dopad než emocionální výkřiky ...



# Bodování, zápočty, známky, ...

- ▶ na posledním cvičení ještě drobná písemka
- ▶ odevzdávání týmové úlohy
- ▶ zápočty/znamky pouze elektronicky ...
- ▶ zapíšeme co nejdříve, na počátku zkouškového
- ▶ hlašte svým cvičícím, pokud ještě nemáte vše v pořádku
- ▶ lze i během ledna/února, pozor na konec zkouškového



# Věžňovo dilema – základní turnaje

		hráč B			
		C		D	
hráč A	C	4	4	1	6
	D	6	1	2	2

$$P(\text{flip}) = 0.02$$

		hráč B			
		C		D	
hráč A	C	10	10	1	70
	D	70	1	2	2

$$P(\text{flip}) = 0.00$$

		hráč B			
		C		D	
hráč A	C	4	4	4	8
	D	8	4	4	4

$$P(\text{flip}) = 0.02$$

01

# Věžňovo dilema – základní turnaje

		hráč B			
		C		D	
hráč A	C	4	4	1	6
	D	6	1	2	2

$P(\text{flip}) = 0.02$

		hráč B			
		C		D	
hráč A	C	10	10	1	70
	D	70	1	2	2

$P(\text{flip}) = 0.00$

		hráč B			
		C		D	
hráč A	C	4	4	4	8
	D	8	4	4	4

$P(\text{flip}) = 0.02$

01

# Věžňovo dilema – základní turnaje

		hráč B			
		C		D	
hráč A	C	4	4	1	6
	D	6	1	2	2

$P(\text{flip}) = 0.02$

		hráč B			
		C		D	
hráč A	C	10	10	1	70
	D	70	1	2	2

$P(\text{flip}) = 0.00$

		hráč B			
		C		D	
hráč A	C	4	4	4	8
	D	8	4	4	4

$P(\text{flip}) = 0.02$

01

brezimi4	[8, 2, 2]	12
mareddan	[13, 7, 3]	23
tomanka4	[1, 41, 29]	71
bashkkse	[22, 38, 20]	80
tollavoj	[18, 3, 60]	81
mentzleo	[30, 20, 33]	83
lukesj11	[23, 4, 61]	88
hlavava1	[33, 49, 13]	95
hrazdond	[27, 59, 25]	111
kubinmi2	[33, 36, 46]	115
pokorfr4	[123, 6, 5]	134
gebausim	[29, 87, 19]	135
levinjan	[118, 9, 8]	135
hanakdan	[70, 56, 9]	135
vokalro1	[35, 27, 76]	138
lukesma8	[71, 56, 15]	142
kasltoma	[122, 1, 26]	149



# Úspěšné strategie

```
# plays according to given payoff_matrix;  
# reacts to opposing player's moves
```

Hodnocení cvičícího: Velice komplexní hráč s analýzou protihráče a volbou strategie podle ní. Je to super ...

```
# Decent player, ready for unusual matrix
```

Celkově na cca 60 řádcích. Polovina na analýzu Payoff matice, základní rozhodnutí, pak během move analýza historie.

01

# Úspěšné strategie

```
# plays according to given payoff_matrix;  
# reacts to opposing player's moves
```

Hodnocení cvičícího: Velice komplexní hráč s analýzou protihráče a volbou strategie podle ní. Je to super ...

```
# Decent player, ready for unusual matrix
```

Celkově na cca 60 řádcích. Polovina na analýzu Payoff matice, základní rozhodnutí, pak během move analýza historie.



## Pečlivá analýza PM

```
# def analyze_matrix(self):
co_co = self.payoff_matrix[0][0][0]
de_de = self.payoff_matrix[1][1][0]
co_de = self.payoff_matrix[0][1][0]
de_co = self.payoff_matrix[1][0][0]
#if opposite moves together are better than any same options t
if (co_de + de_co) > (2 * co_co) and \
    (co_de + de_co) > (2 * de_de):
    if(de_co > co_de):
        return 'OppositeDefect'
    else:
        return 'OppositeCooperate'

if co_co >= de_de and co_co >= de_co:
    return 'CooperateForever' #cooperate is better than everyt
if co_co >= de_de:
    return 'CooperateAndBetray' #standard, both Cooperates are
else:
    return 'Defect' #if both Defects are better than both Coop
```

Dále:

- ▶ vhodně začít
- ▶ zužitkovat případnou znalost posledního tahu
- ▶ tit-for-tat (varianty)

01

## Ještě úspěšnější strategie ...

... ale neříkejte to Mirkovi Dušínovi ;-)

```
1 blackwiz [ 1, 1, 1] 3
2 paclimar [13, 2, 6] 21
```

v bodech zdrcující odstup (např. pro turnaj 2)

```
1 | blackwiz | 4 | 374008.0 | 5376 |
2 | paclimar | 4 | 204643.0 | 5376 |
```

## Ještě úspěšnější strategie ...

... ale neříkejte to Mirkovi Dušínovi ;-)

```
1 blackwiz [ 1, 1, 1] 3
2 paclimar [13, 2, 6] 21
```

v bodech zdrcující odstup (např. pro turnaj 2)

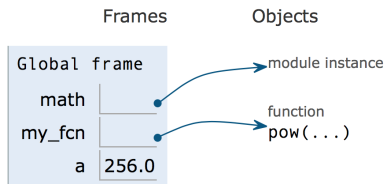
```
1 | blackwiz | 4 | 374008.0 | 5376 |
2 | paclimar | 4 | 204643.0 | 5376 |
```

# Přejmenování funkce

Write code in Python 3.3

(drag lower right corner to resize code editor)

```
1 import math
2
3 my_fcn = math.pow
4 → a = my_fcn(2,8)
```

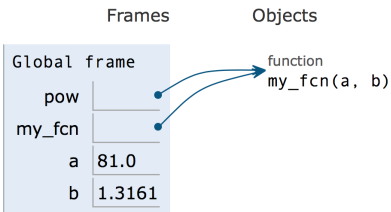


# Ukradení funkce

Write code in Python 3.3

(drag lower right corner to resize code editor)

```
1 from math import pow
2
3 def my_fcn(a,b):
4     return a**(1/b)
5
6 a = pow(3,4)
7 pow = my_fcn
8 → b = pow(3,4)
```



Ale pozor, pomalu přecházíme na temnou stranu Síly.

01

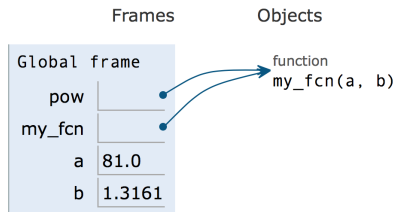


# Ukradení funkce

Write code in Python 3.3

(drag lower right corner to resize code editor)

```
1 from math import pow
2
3 def my_fcn(a,b):
4     return a**(1/b)
5
6 a = pow(3,4)
7 pow = my_fcn
8 → b = pow(3,4)
```



Ale pozor, pomalu přecházíme na **temnou stranu Síly**.

# Jak najít soupeřovu funkci?

Introspekce - průzkum běžícího kódu, <https://docs.python.org/3.4/reference/datamodel.html>

`object.__dict__`

# Jak najít soupeřovu funkci?

Introspekce - průzkum běžícího kódu, <https://docs.python.org/3.4/reference/datamodel.html>

`object.__dict__`

# Jak najít soupeřovu funkci?

Introspekce - průzkum běžícího kódu, <https://docs.python.org/3.4/reference/datamodel.html>

`object.__dict__`

# Brzy se dozvíte více ...

- ▶ B0B01LGR - Logika a grafy
- ▶ B4B33ALG - Algoritmizace
- ▶ B4B36ZUI - Základy umělé inteligence
- ▶ B4B33RPZ - Rozpoznávání a strojové učení
- ▶ BE4M33SSU - Statistical Machine Learning
- ▶ ...

01