

Oznam 1

Vo štvrtok 19.10. je študijné voľno.

Študenti, ktorí majú cvičenia vo štvrtok, sa tento týždeň môžu zúčastniť cvičenia:

Streda 18.10. od 17:00 do 19:00 v cpu

Pre študentov, ktorí majú cvičenie vo štvrtok, ale nie je účasť na cvičení tento týždeň povinná.

Ak máte cvičenia vo štvrtok a chceli by ste sa zúčastniť niektorého cvičenia tento týždeň, ale cvičenie v stredu o 17:00 vám nevyhovuje, napíšte email prednášajúcemu (tomas.fabsic@stuba.sk).

Oznam 2

Prvý test bude na cvičeniach v týždni 8.

PROG1: Prednáška 5

Rekurzia

Domaca uloha

1. Vyriesit ulohy z Cvicenia 5.
2. Precitajte v knihe:
 - casti 5.8 - 5.14
 - celu kapitolu 6

Rekurzia - uvod

Uz sme videli, ze funkcie mozu volat ine funkcie.

Napr. v skripte

```
def parne(n):  
    for i in range(n):  
        print((i+1)*2)  
  
def parne_interaktivna_verzia():  
    print('Zadajte kolko parnych cisel chcete vypisat.')  
    vstup=input()  
    n=int(vstup)  
    parne(n)  
  
parne_interaktivna_verzia()
```

vola funkcia “*parne_interaktivna_verzia*“ funkciu “*parne*“.

Rekurzia - uvod

Uz sme videli, ze funkcie mozu volat ine funkcie.

Ale funkcie mozu volat aj same seba!

Takyto jav nazyvame rekurzia!

Rekurzia je velmi casto pouzivany nastroj pri navrhu algoritmov!

Ako vyzerá rekurzívna funkcia?

Priklad 1:

```
def odpocitavanie(n):  
    if n==0:                # ukoncujuca podmienka  
        print('Start!')  
    else:  
        print(n)  
        odpocitavanie(n-1) # rekurzivne volanie
```

Ako vyzerá rekurzívna funkcia?

Priklad 1:

```
def odpocitavanie(n):  
    if n==0:                # ukoncujuca podmienka  
        print('Start!')  
    else:  
        print(n)  
        odpocitavanie(n-1) # rekurzivne volanie
```

Rekurzívna funkcia vždy obsahuje:

- **rekurzívne volanie**
a
- **ukoncujuću podmienku** (base case)

Ako vyzerá rekurzívna funkcia?

Priklad 1:

```
def odpocitavanie(n):  
    if n==0:                # ukoncujuca podmienka  
        print('Start!')  
    else:  
        print(n)  
        odpocitavanie(n-1) # rekurzivne volanie
```

Vyskusajte si funkciu "odpocitavanie" spustit.

Ako vyzera beh rekurzivnej funkcie?

```
def odpocet(n):  
    print('Zacina sa vykonavat funkcia odpocet s argumentom n=',n, sep='')  
    if n==0:                # ukoncujuca podmienka  
        print('Start!')  
    else:  
        print(n)  
        odpocet(n-1) # rekurzivne volanie  
    print('Konci sa vykonavanie funkcie odpocet s argumentom n=',n, sep='')
```

Vyskusajte si funkciu "odpocet" spustit s argumentom n=3.

Ako vyzerá beh rekurzívnej funkcie?

Beh funkcie odpocet vyzerá nasledovne:

Zacína beh funkcie odpocet s $n=3$.

 Zacína beh funkcie odpocet s $n=2$.

 Zacína beh funkcie odpocet s $n=1$.

 Zacína beh funkcie odpocet s $n=0$.

 Končí beh funkcie odpocet s $n=0$.

 Končí beh funkcie odpocet s $n=1$.

 Končí beh funkcie odpocet s $n=2$.

 Končí beh funkcie odpocet s $n=3$.

Vsimnite si: Beh funkcie s argumentom 3 sa ukončí, až keď sa ukončí beh funkcie s argumentom 2. Beh funkcie s argumentom 2 sa ukončí, až keď sa ukončí beh funkcie s argumentom 1. Beh funkcie s argumentom 1 sa ukončí, až keď sa ukončí beh funkcie s argumentom 0.

Hovoríme, že beh funkcie s argumentom 0 je **vnorený** do behu funkcie s argumentom 1, beh funkcie s argumentom 1 je **vnorený** do behu funkcie s argumentom 2 a beh funkcie s argumentom 2 je **vnorený** do behu funkcie s argumentom 3.

Nekonecna rekurzia

!!! Pozor na nekonecnu rekurziu !!!

Priklad 1 (chybna verzia):

```
def odpocitavanie(n):  
    print(n)  
    odpocitavanie(n-1)
```

Vyskusajte si spustit takto definovanu funkciu "odpocitavanie"!

Porovnajte tuto funkciu s funkciou na piatom slajde! (V chybnej verzii **chyba ukoncovacia podmienka!**)

Rekurzia – priklad 2

Priklad 2:

```
def factorial(n):  
    if n==1:  
        return 1  
    else:  
        rec=factorial(n-1)  
        result=n*rec  
        return result
```

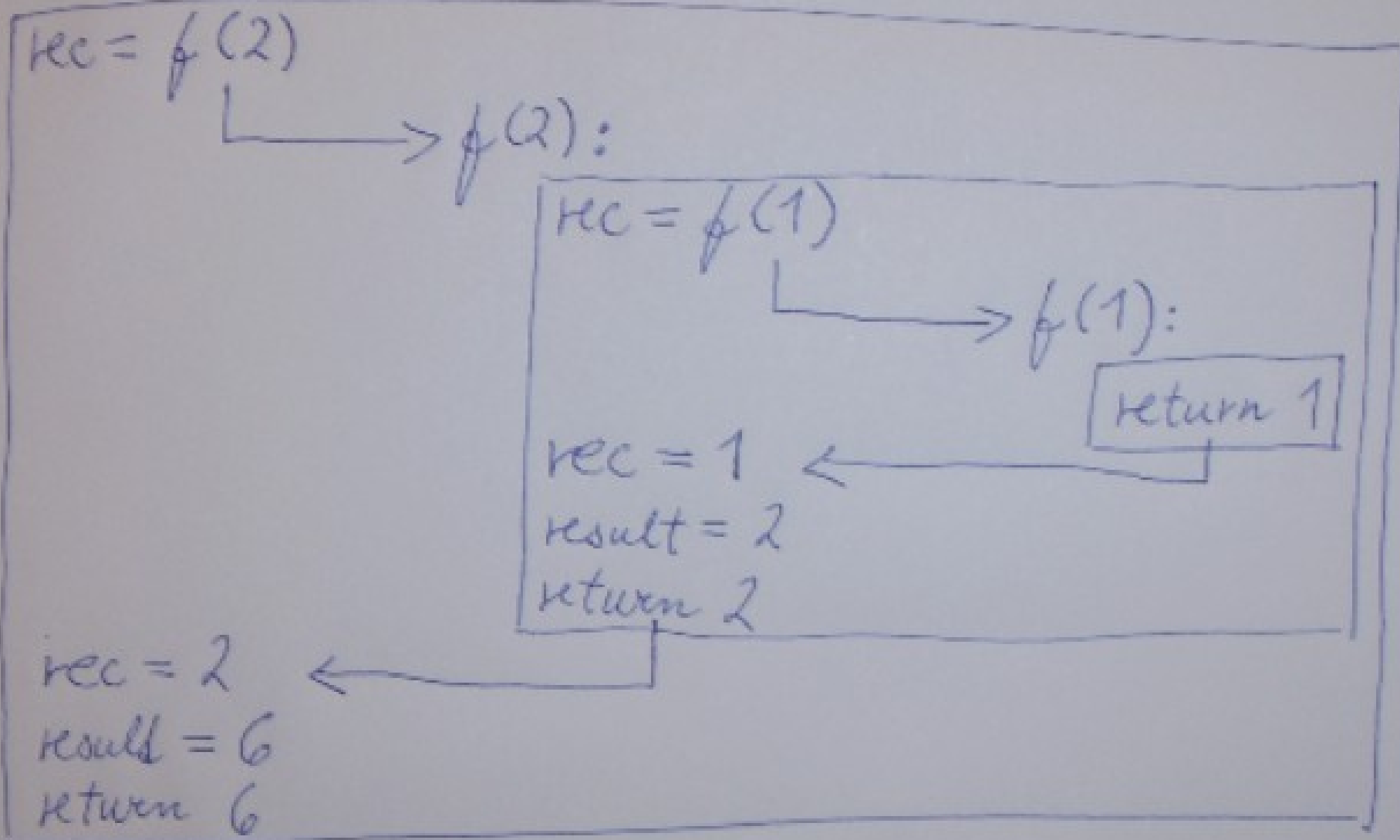
Rekurzia – príklad 2

Funkcia factorial s popisom:

```
def factorial(n):  
    print('Zacina sa vykonavat funkcia factorial s argumentom n=',n,sep='')  
    if n==1:  
        print('Funkcia factorial s argumentom 1 vrati hodnotu 1')  
        return 1  
    else:  
        rec=factorial(n-1)  
        result=n*rec  
        print('Funkcia factorial s argumentom n=',n,end='',sep='')  
        print(' vrati hodnotu ',n,'!',sep='')  
        return result
```

Vyskusajte spustit factorial(3) !

$f(3)$:



Rekurzia – príklad 2

- Podrobný popis behu funkcie factorial najdete na webstránke predmetu v dokumente factorial.pdf.
- Dokument factorial.pdf najdete pri prezentácii z prednasky.

Rekurzia – globalne premenne

- **Pozor!** Nevytvarajte rekurzivne funkcie pomocou globalnych premenných!
- Je to zly zvyk a na skuske bude za taketo riesenia 0 bodov!

Ako tvorit rekurzivne funkcie?

Uloha:

Definujte funkciu F s parametrom n . Funkcia nacita z klavesnice n cisel a vrati sucet nacistanych cisel. Vo funkcii pouzite **rekurziu!**

Postupujeme v 2 krokoch.

Krok 1:

Polozim si otazku:

Ako by som vedel dosiahnut efekt funkcie F pre argument n pomocou volania funkcie F s nejakym nizsim argumentom? (napriklad pomocou $F(n-1)$)

Krok 2:

Polozim si otazku:

Ako sa ma rekurzia zastavit? (t.j. Aku ukoncovaciou podmienku potrebujem?)

Ako tvorit rekurzivne funkcie?

Uloha:

Definujte funkciu F s parametrom n . Funkcia nacita z klavesnice n cisel a vrati sucet nacistanych cisel. Vo funkcii pouzite **rekurziu!**

Krok 1:

Ako by som vedel dosiahnut efekt funkcie F pre argument n pomocou volania funkcie F s nejakym nizsim argumentom? (napriklad pomocou $F(n-1)$)

V nasom pripade:

Efekt funkcie F pre argument n viem dosiahnut tak, ze:

1. zavolam funkciu F s argumentom $(n-1)$
2. hodnotu, ktoru mi $F(n-1)$ vrati, si ulozim (napr. do premennej rec)
3. nacistam jedno cislo z klavesnice
4. cislo pripocitam k premennej rec (sucet cisla a rec oznacim ako suc)
5. vratim suc

Ako tvorit rekurzivne funkcie?

Uloha:

Definujte funkciu F s parametrom n . Funkcia nacita z klavesnice n cisel a vrati sucet nacistanych cisel. Vo funkcii pouzite **rekurziu!**

Krok 2:

Ako sa ma rekurzia zastavit? (t.j. Ako ukoncovaci podmienku potrebujem?)

V nasom pripade:

Pre argument $n=0$, staci, ak funkcia vrati nulu.