

Programátorské finty

Co všechno jde, i když třeba těžko.

© Leonard Walletzký, ESF MU, 2003

Kontrola uživatelského vstupu

- Je třeba zajistit vložení správných parametrů do programu a funkce
- Kontrola u programu
 - cyklus *while*
 - program potřebuje správnou hodnotu, aby mohl pracovat dál
- Kontrola u funkce
 - struktura *if*
 - funkci stačí, pokud zjistí, že vstup je neplatný

Příklad

- Modifikujte funkci *soucet* tak, aby správně reagovala při chybném vstupu
- **Co je chybný vstup?**
 - zadání prázdného vektoru
 - funkce *length(v)* vrátí 0
- **Co vrátíme v případě chybného vstupu?**
 - nejlépe hodnotu, která se dá jednoduše testovat v programu a jinak smysl nemá
 - nejčastěji prázdný vektor []

Modifikace funkce

```
function s=soucet(v)
vel=length(v);
if vel==0
    s=[ ];
else
    s=0;
    for i=1:vel
        s=s+v(i);
    end
end
end
```

Program – kontrola vstupu

- Modifikujte **program** na výpočet součtu prvků vektoru, který zadáme pomocí *input*.
- Tvar kontrolního cyklu:
while ~<podmínka platnosti vstupu>
 <proměnná>=*input*(<text výzvy k opravě>);
end
- Stanovení podmínky platnosti
 - $\text{length}(v) > 0$

Řešení

```
v=input('Zadej vektor');  
while ~(length(v)>0)  
    v=input ('Spatne. Zadej opravdovy vektor:');  
end  
vel=length(v);  
s=0;  
for i=1:vel  
    s=s+v(i);  
end  
disp(s)
```

Příklady

- Napište funkci, která zjistí, zda vektor obsahuje zadané číslo (obsahuje=1, neobsahuje 0)
- Funkce vrátí pravdivostní hodnotu!
- Funkce bude mít dva vstupní parametry (proměnné)
 - a – číslo, které chci testovat
 - v – vektor, ve kterém se má nacházet

Řešení

```
function c=obsahuje(a,v)
vel=length(v);
c=0;
for i=1:vel
    if a==v(i)
        c=1;
    end
end
end
```


Modifikace funkce

- Modifikujte funkci *obsahuje* tak, aby cyklus probíhal do chvíle než:

- najdu číslo *a*
- dorazím na konec vektoru

- Místo cyklu `for` použijte cyklus `while`

```
i=1;
while v(i)~=a&i<length(v)
    i=i+1;
end
if a==v(i)
    c=1;
else
    c=0;
end
```

Příklady

- Napište funkci *nahoda*, které na vstupu zadáme číslo n a ona vrátí vektor n neopakujících se náhodných čísel z intervalu $(0, 100)$
- Modifikujte funkci *obsahuje* tak, aby správně reagovala, zadáme-li jí na vstupu prázdný vektor

Příklady

- Modifikujte funkci *nahoda* tak, abychom jí na vstupu zadávali nejen počet náhodných čísel, ale také horní a dolní hranici intervalu, ze kterého mají být
- Napište program na hru Sazka, který hráči umožní:
 - definovat interval čísel pro hru
 - vsadit možná čísla
 - vygeneruje „tah“ a vyhodnotí úspěšnost hráče

Příklady

- Napište program, ve kterém budete hádat číslo, které si „myslí“ počítač.
- Napište funkci, která vygeneruje zadaný počet tipů Sazky a vrátí je jako vektor
- Napište program, který vám umožní vsadit si Sazku a řekne vám, kolikrát jste se trefili.

Příklady

- Modifikujte předcházející hry pro více hráčů - jejich počet není předem znám
- Napište program na hru „kámen-nůžky-papír“ při zachování jejích pravidel.

Příklady

- Napište hru pro dva hráče s následujícími pravidly:
 - počítač vygeneruje matici, obsahující náhodně prvky 1,2
 - hráči se střídají v tazích a snaží se uhodnout, které číslo je na daném místě.
 - pokud uhodnou, získávají bod
 - počítač zobrazuje pouze odkrytá pole

Příklady

- Modifikujte předchozí příklad tak, aby hráči zadávali i:
 - rozměr matice (čtvercová)
 - počet možných prvků

Příklady

- Napište program, který vygeneruje náhodně 10 lineárních rovnic $ax+b=c$, nakreslí jejich graf a napíše jejich řešení.
- Napište program, kterému zadáte parametry paraboly a přímky. Program zobrazí jejich graf a určí, zda mají průsečík. Pokud ano, vypočte jeho souřadnice