

Петље одобравања улаза

Програмерски израз “смеће уђе, смеће изађе” (garbage in, garbage out) (GIGO), се односи на чињеницу да рачунар не може да разликује доброг од лошег податка.

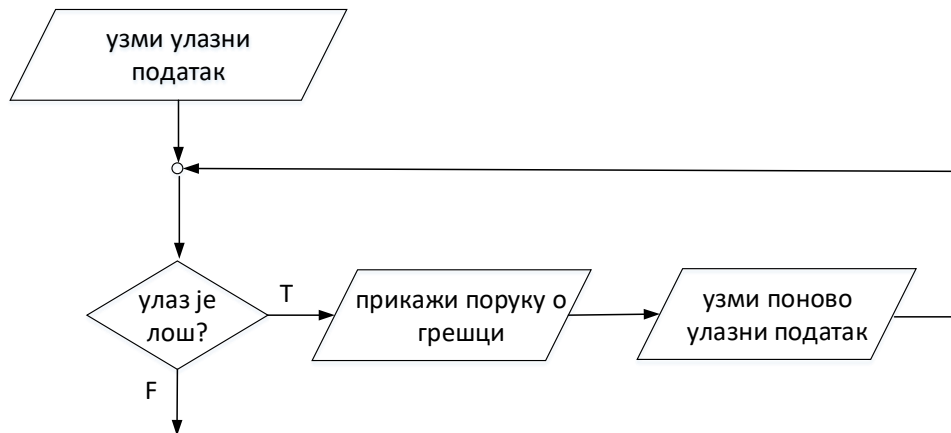
Ако корисник унесе лош податак, рачунар ће га обрадити и на излазу се добија лош резултат.

Овде се мисли чак и на случајне грешке које корисник направи погрешним укуцавањем са тастатуре.

Зато је неопходно дизајнирати програм тако да се лош податак никако не може ни почети обрађивати, а то значи да је неопходно проверити све улазне податке пре обраде.

Ако је улазни податак лош, треба га одбацити и тражити од корисника да унесе добар податак.

Овај процес се назива одобравање улаза (input validation) а понекад и уређивач грешака (error handler).



Прву узимање улазног податка се назива примарно читање (priming read) и тај податак се први проверава у петљи.

Нестоване петље

Нестована (угнеждена) петља је петља унутар друге петље.

**Пример 1)** Неки могући резултати у фудбалу

```

for domacin in range(2):
    for gost in range(2):
        print(domacin, ':', gost)
  
```

```

0 : 0
0 : 1
1 : 0
1 : 1
  
```

Види се да постоје две петље.

Прва петља управља бројем голова за домаћина а друга за госте.

Када промењива domacin добије вредност 0, gost ће извршити обе итерације са вредностима 0 и 1.

Када промењива domacin добије вредност 1, gost ће извршити обе итерације са вредностима 0 и 1.

Петљ гост је унутрашња петља а домаћин је спољашња петља.

Каже се да унутрашња петља брже итеририра него спољна.

Множењем бројева итерациоје сваке од петљи, добија се укупан број обављених итерација у коду.

**Пример 2)** Цртање правоугаоника карактерима

```
for red in range(8):  
    for kolona in range(6):  
        print('*', end='')  
    print()
```

```
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****
```

**Пример 3)** Цртање косе линије карактерима

```
broj_koraka = 6  
for a in range(broj_koraka):  
    for b in range(a):  
        print(' ', end='')  
        print('#')
```

```
#  
 #  
  #  
   #  
    #  
     #
```