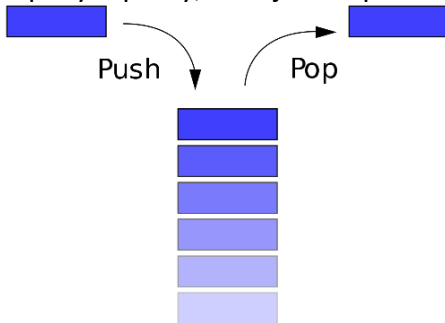


Стек

Стек као тип података

У рачунарству, стек је апстрактни тип података који се користи као колекција елемената.



Користи се и други назив за стек, LIFO (last in first out) са којим се описује редослед коришћења елемената на стеку.

Операције на стеку су: push (додаје елемент на стек), pop (скида елемент са стека), peek (откривање елемента са врха стека без његовог скидања са стека), length (добивања тренутног броја елемената на стеку).

Назив стек се изводи из аналогије са групом физичких објеката који се гомилају једна на друге.

При томе, могуће је смештање новог елемента само на врх стека; исто тако, могуће је само скидање тренутног елемента са врха стека.

То значи, да би се дошло до неког елемента раније смештеног у стеку, може се урадити само скидањем свих елемената са стека који су после њега смештени на стек.

У хардверу рачунара, стек се смешта у меморију која има ограничени капацитет, па се може десити да стек нема више простора за примање нових елемената; овакав случај се назива stack overflow.

Рад са стеком у Пајтону

У Пајтону се користе листе као уграђене структура података, заједно са својим методама, да би симулирале операције којима се дефинише рад са стеком.

0233 Симулација стека употребом листа

```
def main():
    test1()
    test2()

#funkcija kreira stek i inicijalizuje ga kao prazan
def nov_stek():
    stek_prazan = []
    return stek_prazan

def duzina_steka(s):
    return len(s)

#stek je prazan ako je njegova duzina 0
def jeste_prazan(s):
```

```

    return len(s) == 0

#funkcija dodaje element u stek
def dodaj_element(s, element):
    s.append(element)

#funkcija sklanja poslednje uneti element sa steka
def skini_element(s):
    if (jeste_prazan(s) != True):
        return str(s.pop())
    else:
        return "Stek je prazan."

def test1():
    stek = nov_stek()
    print(skini_element(stek))
    proba1 = duzina_steka(stek)
    print("Duzina steka na pocetku je", proba1)
    dodaj_element(stek, 10)
    dodaj_element(stek, 20)
    proba2 = duzina_steka(stek)
    print("Duzina steka posle dva dodata elementa je", proba2)
    skini_element(stek)
    proba3 = duzina_steka(stek)
    print("Duzina steka posle skidanja jednog elementa je", proba3)
    print()
    if proba1 == 0 and proba2 == 2 and proba3 == 1:
        print("1.testiranje koda steka uspesno!!")
    else:
        print("1.testiranje koda steka NEUSPESNO!!")
    print()

def test2():
    stek = [100, 200, 300]
    proba1 = duzina_steka(stek)
    print("Duzina steka na pocetku je", proba1)
    skini_element(stek)
    proba2 = duzina_steka(stek)
    print("Duzina steka posle skidanja prvog elementa je", proba2)
    skini_element(stek)
    skini_element(stek)
    proba3 = duzina_steka(stek)
    print("Duzina steka posle skidanja treceg elementa je", proba3)
    print()
    if proba1 == 3 and proba2 == 2 and proba3 == 0:
        print("2.testiranje koda steka uspesno!!")

```

```
else:  
    print("2.testiranje koda steka NEUSPESNO!!")  
print()  
  
main()
```