

# Modularnost i enkapsulacija

- ❖ Problem: interna implementacija steka (struktura, *stack* i *sp*) je dostupna ostalim delovima programa, pa stoga ti delovi programa mogu:
  - greškom da poremete tu strukturu i dovedu je u nekonzistentno stanje, npr:  
`sp = -5;`
  - da se oslone na informaciju o načinu implementacije (postojanju niza *stack* i indeksa *sp*); ako iz bilo kog razloga imamo potrebu da promenimo tu implementaciju (npr. pređemo na neograničenu, dinamičku strukturu), promene će uticati na sve takve druge delove programa: oni se ili moraju menjati (teško i podložno greškama) ili neće raditi valjano
- ❖ Princip *sakrivanja informacija* (*information hiding*, David Parnas, 1972): svaka logička celina - modul programa, treba da ima jasno izdvojen
  - *interfejs* (*interface*): specifikaciju elemenata (struktura, tipova, operacija...) koje ostali delovi programa mogu da vide i pretpostavki na koje smeju da se oslone, i
  - *implementaciju* (*implementation*): interne delove (strukturu, ponašanje) koje drugi delovi programa ne smeju da vide, niti da se oslanjaju na pretpostavke o njoj
- ❖ Interfejsi treba da budu što opštiji, jednostavniji, kako bi sprege između delova softvera bile jednostavnije, labavije, lakše za kontrolu, a time ti delovi softvera nezavisniji i fleksibilniji
- ❖ *Enkapsulacija* (*encapsulation*) je programska tehnika koja podržava princip sakrivanja informacija (često se ova dva termina poistovećuju i koriste ravnopravno)

# Modularnost i enkapsulacija

❖ Rudiment enkapsulacije u proceduralnom programiranju: razlika između

- deklaracije (potpisa, *signature*) nekog potprograma:

```
int push (unsigned in);
```

- implementacije (potpune definicije tela) tog potprograma:

```
int push (unsigned in) {  
    if (sp==MaxStackSize) return -1; // Exception: stack full  
    stack[sp++] = in;  
    return 0;  
}
```

- ❖ Motivacija je izvorno banalna — nezavisno prevođenje modula: na mestu poziva, za generisanje koda pozivaoca, dovoljno je znati samo potpis potprograma, ne i celo telo; telo je potrebno na mestu gde se generiše kod za telo tog potprograma
- ❖ Opštije, ova ista rudimentarna tehnika predstavlja i podršku enkapsulaciji na nivou potprograma, kao osnovne gradivne jedinice proceduralnog programa
- ❖ Sličan princip potreban je i na krupnijem stepenu granularnosti - modulu