

# Obrada izuzetaka

- ❖ Šta je uopšte izuzetak? Šta je izuzetna situacija, a šta nije? Kada koristiti izuzetke, a kada prosto ispitivati stanje i rezultate?
- ❖ Primer: procedura koja učitava znakove iz nekog ulaznog znakovnog toka (npr. fajla) i obrađuje ih. Da li je nailazak na kraj fajla izuzetak?
- ❖ Ako se ne tretira kao izuzetak:  

```
ifstream f("input.txt");  
while (!f.eof()) {  
    char c = f.getc();  
    // ...  
}
```

*ifstream* je bibliotečna klasa za ulazne tekstualne fajlove kao znakovne tokove
- ❖ Ako se tretira kao izuzetak:  

```
ifstream f("input.txt");  
while (true) {  
    try {  
        char c = f.getc();  
        // ...  
    } catch (...) {  
        break;  
    }  
}
```
- ❖ Naravno da je ovo drugo potpuno besmisleno, jer kraj fajla nije neregularna, nego sasvim očekivana i ispravna situacija. Osim toga, kod koji bi tako tretirao kraj fajla bio bi potpuno nepregledan i nelogičan
- ❖ Međutim, šta ako procedura treba da učitava složenije strukture, recimo zapise koji imaju odgovarajući format i strukturu; šta ako se tada nađe na kraj fajla u sledećim slučajevima: a) tačno nakon završetka celog zapisa; b) pre završetka zapisa?
- ❖ Sledeći primer: procedura pretražuje neku kolekciju elemenata i traži zadati element: a) koji može, a ne mora da bude u njoj; b) koji bi morao da bude u njoj, jer je to regularno stanje programa? Kako se tretira situacija ako ta procedura ne nađe traženi element?

# Obrada izuzetaka

- ❖ Paradigma programiranja i projektovanja softvera pod nazivom *programiranje po ugovoru* (*programming by contract/design by contract*, Bertrand Meyer, 1986, programski jezik *Eiffel*) pretpostavlja preciznu, formalnu specifikaciju interfejsa softverskih komponenata (npr. potprograma ili klase) za koje se definišu logički uslovi (kao logički izrazi) sledećih vrsta:
  - *preduslovi* (*precondition*): logički uslovi koji moraju da budu zadovoljeni prilikom poziva potprograma da bi taj potprogram mogao da obavi svoj zadatak; ove uslove mora da zadovolji pozivalac pri pozivu potprograma; na primer, uslovi u pogledu stanja globalnih promenljivih ili argumenata poziva potprograma (npr. indeks mora biti unutar granica niza)
  - *postuslovi* (*postcondition*): logički uslovi koji moraju da budu zadovoljeni na kraju izvršavanja potprograma; ove uslove mora da zadovolji pozvani potprogram; na primer, uslovi u pogledu stanja globalnih podataka ili validnosti povratne vrednosti funkcije
  - *invarijante* (*invariant*): logički uslovi vezani za neku komponentu (npr. instancu apstraktnog tipa podataka, objekat klase, modul, strukturu podataka) koji moraju da budu zadovoljeni uvek, osim u tranzijentnim periodima kada se ta komponenta prevodi iz jednog validnog stanja (kada invarijanta važi) u drugo validno stanje (kada invarijanta ponovo važi); na primer, metode klase koje prevode objekat iz jednog regularnog u drugo regularno stanje, tokom koje invarijanta privremeno ne mora da važi
- ❖ Prema tome, preduslovi su obaveze koje mora da izvrši pozivalac prema pozvanom potprogramu, da bi taj potprogram mogao da izvrši svoju obavezu; postuslovi su obaveze koje potprogram mora da ispuni ukoliko je pozivalac ispunio svoje obaveze; zato oni predstavljaju *ugovor* (*contract*) između dve strane