Obrada izuzetaka

- *try-catch* konstrukt može da uokviri celo telo funkcije, uključujući i inicijalizatore podobjekta osnovne klase i članova klase za konstruktore te klase (u kojima se ovo tipično i koristi); svaki *catch* za ovakav *try* treba da se završi bacanjem izuzetka; ukoliko toga nema, isti izuzetak koji se obrađuje se implicitno baca na kraju *catch* bloka (kao sa *throw;*)

```
template <typename T>
class huge_vector : public vector<T> {
  …
}

template <typename T>
huge_vector<T>::huge_vector<T> () try : vector<T>(0x100000000) {
}
catch (bad_alloc& e) {
  cerr<<"Cannot allocate a huge vector: “<< e.what() << “\n";
}
```

Pokušaj alokacije niza ove veličine; može baciti izuzetak ako alokacija ne uspe

Ovde je implicitan *thow;*

- Ako se ne pronađe odgovarajući hvatač za bačeni izuzetak sve do dna steka poziva za tekuću nit, izvršava se bibliotečna funkcija *terminate* koja podrazumevano prekida izvršavanje programa (ali se to može promeniti)
- Ako se funkcija označi kao *noexcept*, onda ona implicitno hvata sve izuzetke koji eventualno nisu obrađeni unutar nje i poziva funkciju *terminate*; takva funkcija tako nikada ne baca (ne prosleđuje) izuzetke:

```
void transaction () noexcept {
  …
}
```

Ova funkcija ne baca izuzetke, čak i ako ih ne obrađuje

isto ovo se može postići i eksplicitno:

```
void transaction () noexcept {
  try {
    …
  }
  catch (...) {
    std::terminate();
  }
}
```

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();
  // …
}
```

- 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 naiđe 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 sutuacija ako ta procedura ne nađe traženi element?