Obrada izuzetaka

- Kako se mogu tretirati izuzeci u slučaju da na programskom jeziku nemamo posebnu podršku za njihovu obradu?
  Svaki potprogram (koji može signalizirati izuzetak) može da vrati status svog izvršavanja kao povratnu vrednost:

```
enum DeviceStatus {ok, deviceFaulty, communicationFailed};

DeviceStatus readTemperature (Temperature* value);
DeviceStatus readPressure (Pressure* value);
DeviceStatus readHumidity (Humidity* value);
…

void readMeteo () {
  Temperature temp;
  Pressure press;
  Humidity hum;

  DeviceStatus status = readTemperature(&temp);
  if (status==ok) {

    status = readPressure(&press);
    if (status==ok) {

      status = readHumidity(&hum);
      if (status==ok) {

        // Finally, do the job with temp, press, and hum

      } else {

        // Handle readHumidity exception

      }

    } else {

      // Handle readPressure exception

    }

  } else {
    // Handle readTemperature exception
  }
}
```

Obrada izuzetaka

- Problemi:
  - kod postaje nepregledan čim postoji nekoliko koraka koji mogu da rezultuju izuzetkom (duboko i nepregledno ugnežđivanje naredbi *if-then-else*)
  - funkcije koje vraćaju status ne mogu da vraćaju svoje "prirodne" rezultate, koji se zato moraju prenositi kao izlazni argumenti po referenci/preko pokazivača
  - pozivalac funkcija koje vraćaju kod greške može greškom ili namerno da ignoriše te greške: da ih ne proverava i ne obrađuje, ili da ih ne propagira svom pozivaocu
    - ispitivanje povratnih statusa, tj. postojanja grešaka, troši vreme i resurse najčešće bespotrebno
- Uzrok problema: obrada izuzetaka učešljana je u regularan tok, isprepletana je sa osnovnim tokom, pa ga čini nepreglednim i težim za programiranje
- Bolji pristup:
  - regularni tok programira se kao da je sve u redu, tj. da nema izuzetaka i nije opterećen njihovom obradom,
  - s tim da određeni koraci regularnog toka mogu da *bace* (*throw*) izuzetak
  - obrada izuzetaka se piše kao *obrađivač izuzetka* (*exception handler*) pridružen bloku sa regularnim tokom, koji *hvata* izuzetak (*catch*), ali nije izmešana sa njim
- Mehanizam obrade izuzetaka nije direktno vezan za objektno orijentisanu paradigmu, ali je podržan u svim današnjim popularnim OO programskim jezicima na vrlo sličan način, po istom principu