

Obrada izuzetaka

- ❖ Izuzetak može biti detektovan u *okruženju* programa
- ❖ Na primer, sledeće izuzetne situacije detektuje hardver (procesor), a obrađuje operativni sistem, podrazumevano tako što gasi program koji je izazavao tu situaciju:
 - nelegalan kod operacije, način adresiranja i slično: ako se radi o programu koji je preveden sa višeg programskog jezika, može da znači “skok” na adresu na kojoj se ne nalazi regularan mašinski kod, već slučajan sadržaj; na primer, skok u potprogram dinamičkim vezivanjem: preko “oštećenog” pokazivača na funkciju, ili poziv virtuelne funkcije oštećenog pokazivača na objekat ili oštećenog objekta
 - neregularan pristup memorijskoj lokaciji: adresiranje memorijske lokacije koja nije dozvoljena, ili pristup toj lokaciji koji nije dozvoljen (npr. upis u lokaciju za koju je dozvoljeno samo čitanje); slično, ovo može biti posledica oštećenih pokazivača na objekte
- ❖ Kako C/C++ ne proverava pokazivače, već prevodilac prosto generiše kod za pristup do objekta ili za poziv funkcije kao u regularnim uslovima, ponašanje programa u slučaju oštećenih pokazivača postaje nedefinisano i pre ili kasnije može da rezultuje ovakvim izuzecima
- ❖ Izuzetak može da detektuje i signalizira i operativni sistem, kada program zahteva neku njegovu uslugu tzv. *sistemskim pozivom*, npr. zahteva ulazno / izlaznu operaciju, alokaciju dela memorije i slično; u ovom slučaju, operativni sistem će vratiti grešku, a implementacija bibliotečnog potprograma u kom je sistemski poziv može da signalizira izuzetak
- ❖ Bibliotečni potprogrami ili konstrukti programskog jezika takođe mogu da signaliziraju izuzetak jer ne mogu da obave svoj zadatak iz određenog razloga
- ❖ Slično, i potprogrami ili moduli samog programa mogu da signaliziraju izuzetak jer iz nekog razloga ne mogu da obave zahtevanu operaciju kako je to predviđeno

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
```

```
    }
```

```
}
```