

Prevođenje

- ❖ Prevodilac učitava znak po znak iz ulaznog fajla sa izvornim kodom programa; iako tekst programa ljudi doživljavaju dvodimenzionalno (u ravni), pri čemu im te dve dimenzije pomažu u čitanju i razumevanju (prelom redova, proredi i uvlačenje, odnosno “nazublјivanje” koda), prevodilac kod tumači isključivo sekvencijalno
- ❖ Prevodilac najpre učitane znakove grupiše u veće celine, tzv. *leksičke elemente* ili *lekseme* (*lexical element*, *lexem*), ili *žetone* (*token*), u skladu sa pravilima jezika; ova faza prevođenja naziva se *leksička analiza* (*lexical analysis*); na primer, u sledećem delu koda, različitim bojama označene su različite lekseme:

```
if (i++ +j>=0 && i<this->size())
```

- ❖ U daljem postupku prevodilac tretira lekseme kao integralne celine, odnosno kao elemente od kojih su izgrađeni krupniji jezički iskazi, tj. rečenice
- ❖ Prevodilac tokom prevođenja prepoznaje te veće jezičke celine (rečenice) na osnovu *gramatike* (*grammar*) jezika; ova faza prevođenja naziva se *parsiranje* (*parsing*); u slučaju prestupa nekog pravila gramatike, prevodilac prijavljuje grešku u prevođenju
- ❖ Za prepoznate rečenice i elemente u njima, prevodilac proverava ostala pravila jezika, tzv. *semantička pravila* (*semantic rules*), i opet prijavljuje greške u slučaju prestupa
- ❖ Konačno, za one elemente rečenica za koje je to definisano semantikom jezika, prevodilac generiše sadržaj u prevedenom objektnom fajlu u kome se principijelno nalazi:
 - binarni mašinski kod za mašinske (procesorske) instrukcije
 - alociran prostor za određene kategorije objekata sa tzv. *statičkim trajanjem skladištenja* (*static storage duration*)

Prevođenje

- ❖ Kada naiđe na novu deklaraciju, prevodilac dodaje deklarisanu identifikator u strukturu podataka koju izgrađuje tokom prevođenja i koja se tradicionalno naziva *tabela simbola* (*symbol table*); u ovoj strukturi prevodilac čuva informacije o svakom deklarisanom identifikatoru: o tome kojoj jezičkoj kategoriji pripada (tip, objekat, funkcija itd.), kog je tipa, kao i sva ostala svojstva deklarisanog entiteta definisana pravilima jezika
- ❖ Kada naiđe na neki upotrebljen identifikator, prevodilac:
 - proverava da li je taj identifikator deklarisan i da li je dostupan, po pravilima jezika; ako nije, prijavljuje grešku;
 - proverava da li je identifikator upotrebljen u skladu sa pravilima jezika i ako nije, prijavljuje grešku; na primer, ne može se vršiti operacija $f++$ ako je f funkcija, ili operacija $a()$ ako je a objekat tipa *int* i slično;
 - ako je to definisano semantikom jezika, zna kako da generiše kod za upotrebu tog identifikatora u odgovarajućem kontekstu