## **Informatika 3**



## Deklarácia

Deklarácia premenných

<ukladacia\_trieda> <modifikátor> <deklarátor> pr1, pr2, ... prN;

- Ukladacia trieda:
  - auto
  - extern
  - static
  - register
  - volatile



## Automatická premenná

#### auto

- deklarovaná v zloženom príkaze (funkcii)
- lokálne v zloženom príkaze
- pri začiatku vykonávania zloženého príkazu premenná vzniká, pri skončení zaniká
- je alokovaná v zásobníku rýchla alokácia
- kľúčové slovo auto je nepovinné lokálne premenné sú implicitne auto

auto int i=0;



# Externé premenné

#### extern

- deklarácia premennej bez alokácie pamäťového miesta
- deklarované mimo akejkoľvek funkcie
- globálne premenné, dostupné v celom programe
- kľúčové slovo extern použité iba pre deklaráciu premennej, ktorá bola definovaná v inom zdrojovom súbore
- Vo všetkých zdrojových súboroch musí byť externá premenná dekladovaná slovom extern, ale v jednom musí byť definovaná bez kľúčového slova extern
- najčastejšie používané deklarácie globálnych premenných v hlavičkových súboroch



# Statická premenná

- static
  - platia pre ňu pravidlá ako pre globálnu premennú
    - vzniká pri spustení programu
    - · inicializuje sa na nulu alebo na hodnotu na zadanú pri deklarácií
- static lokálne premenné
  - deklarované v zloženom príkaze alebo vo funkcii
  - identifikátor premennej je platný v rámci zloženého príkazu alebo funkcie
  - po vykonaní zloženého príkazu/funkcie nie je premenná zrušená
  - zostáva jej hodnota z predchádzajúceho vykonávania int pocitadlo() {
    static int pocet\_volani = 1; // !!! inicializácia // iba pri spustení programu return pocet\_volani++; }
- static globálne premenné
  - identifikátor premennej je platný iba v rámci zdrojového súboru
  - je opakom ku extern premenná sa nedostane do tabuľky krížových referencií



# Registrové premenné

## register

- pre premenné ku ktorým potrebujeme rýchly prístup
- prekladač sa pokúsi uložiť premennú do registra procesora
- nie je zaručené, že premenná bude v registri (obmedzené množstvo registrov)
- nemožno na premennú použiť operátor adresy



### volatile

- pre premenné, ktoré môžu meniť svoju hodnotu mimo program (napr. cez prerušenie)
- každá operácia s premennou je vykonávaná priamo v pamäti
- operácie sú pomalšie



### Kľúčové slová v deklarácii funkcie

Deklarácia funkcie

```
<typ> <meno>(parametre);
```

- Pred deklaráciou funkcie môže byť:
  - extern
  - static
  - interrupt
  - inline



#### extern

- deklarácia funkcie hlavička
- kľúčové slovo extern pre funkcie je nepovinné, hlavičky funkcií sú implicitne typu extern
- identifikátor funkcie sa dostane do tabuľky krížových referencií – implicitné chovanie hlavičiek funkcií

### static

 je opakom ku extern – identifikátor funkcie sa nedostane do tabuľky krížových referencií



## extern, static pre premenné a funkcie

- Pre globálne premenné a funkcie sa extern a static používa pri linkovaní viacerých zdrojových súborov
- static funkcia/globálna premenná platná iba v danom zdrojovom súbore
- extern funkcia/globálna premenná platná v celom programe



## interrupt

- funkcie, ktoré sú väčšinou volané cez prerušenie
- pred spustením funkcie sa uchová aktuálny stav registrov
- po ukončení funkcie je obnovený stav registrov
- volanie funkcie je pomalšie



### inline

- pre jednoduché funkcie
- funkcia nie je volaná ako štandardná funkcia ale je kompilovaná priamo do volania
- ak sa používa vo viacerých zdrojových súboroch, musí byť definovaná (aj s telom) v hlavičkovom súbore
- nie je možné získať smerník na funkciu



## Veľké projekty

- Rozdeliť program do viacerých zdrojových súborov
- Používať hlavičkové súbory
- V hlavičkovom súbore:
  - deklarácia štruktúr
  - deklarácia globálnych premenných s kľúčovým slovom extern
  - hlavičky funkcií
  - symbolické konštanty
- Prekladač automaticky zistí závislosti zdrojových súborov na hlavičkových súboroch
- Pri zmene hlavičkového súboru sa prekladajú len súbory, ktoré sú na ňom závislé
- #include aj v súbore, kde sa funkcia definuje
- linkovanie tabuľka krížových referencií

