ZPRO 5. cvičení

Co jsme nestihli minule

- Cyklus **for**

- V inicializaci lze deklarovat i více proměnných. Potřebujeme-li jako krok několik výrazů **oddělíme je čárkou**.

- Není možné definovat proměnné více datových typů v inicializaci cyklu for

Příklad: Otočení řetězce

- Napište proceduru:

  void otoc_retezec(char a[], int delka);,

  která hodnoty v poli **a**

Přenos řízení (skoky)

- break;
  - Ukončí cyklus nebo příkaz switch.
- continue;
  - Předčasný přechod k další iteraci
- return;
  - Ukončení funkce (procedury) void.

Nekonečný cyklus

- Cyklus, jehož podmínka je vždy splněna.
- Takový cyklus lze v těle ukončit pouze použitím skoků **break; return; (goto)**

Příklad: Bezpečné čtení nezáporného čísla

- Když se čtení pomocí proudu `cin` nepodaří, například proto, že se pokoušíme číst celé číslo a vstup obsahuje něco jiného, přejde tento proud do chybového stavu a na další požadavky čtení nereaguje.
- „Probudíme“ ho voláním metody `clear()`.
- Testování stavu proudu: instanci proudu lze použít jako podmínku, která je splněna, právě když se poslední operace podařila.

Nepodmíněný skok **goto**

- návěští:
- goto návěští;

Slabé výčtové typy

deklarace_slabého_výčtového_typu:

enum *identifikátor*$_{\text{nep}}$ { *seznam_výčtových_konstant* } **;**

seznam_výčtových_konstant:

výčtová_konstanta

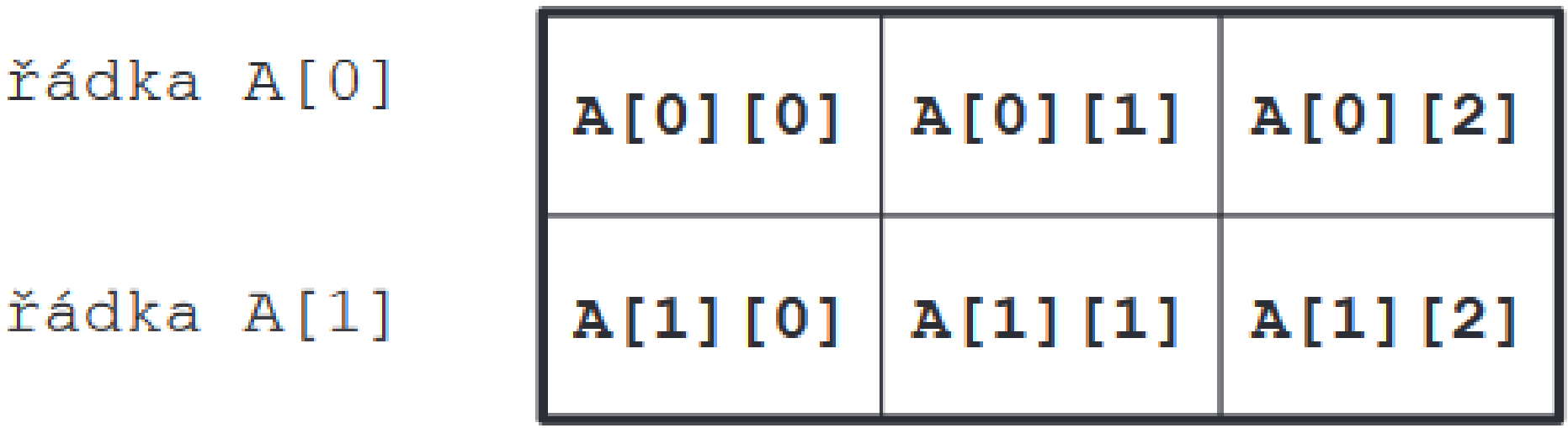
výčtová_konstanta **,** *seznam_výčtových_konstant*

Příklad: Funkce na vypsání světových stran

Vícerozměrné pole

- Deklarace: za identifikátorem tolik specifikací indexů, kolik má deklarované pole rozměrů
- Dvourozměrné pole: Jednorozměrné pole složené z jednorozměrných polí
- Třírozměrné pole: Jednorozměrné pole složené z dvourozměrných polí
- … atd.
- Každý index v samostatných závorkách

```
int A[2][3];
```



Příklad: Sčítání a násobení matic

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj} = \sum_{k=1}^{n} a_{ik}b_{kj},$$

Nulou ukončené znakové řetězce

- Pole typu **char** nebo **wchar_t**

Struktury

- Skupina proměnných (mohou být různého typu) chápaná jako celek
- „Třída bez metod“
- Přístup ke složkám – operátor tečka
- Inicializace – podobně jako pole
- Funkce mohou vracet struktury

deklarace strukturového typu:

struct *identifikátor*$_{\text{nep}}$**{** *deklarace_složek*$_{\text{nep}}$ **};**

Unie

- Skupina proměnných „položených přes sebe“
- Zacházení podobné jako u struktur
- Inicializace - pouze první složka