

18. Algoritmizace - datové typy, deklarace, příklady (datový typ pole - struktura, deklarace, indexy, průchod polem)

Datový typ definuje v programování druh nebo význam hodnot, kterých smí nabývat proměnná (nebo konstanta). Datový typ je určen oborem hodnot a zároveň výpočetními operacemi, které lze s hodnotami tohoto typu provádět (srovnej abstraktní datový typ). Datový typ nemůže být určen pouze oborem hodnot, protože existují i datové typy, lišící se pouze v operacích které je s nimi možné provádět. Součástí programovacího jazyka je definice základních datových typů.

Číselné typy

celočíselné

- **tinyint** – celé číslo, rozsah -128 až +127 (8 bitů)
- **smallint** – rozsah -32 768 až 32767 (16 bitů)
- **mediumint** – rozsah -8 388 608 až 8 388 607 (24 bitů)
- **int** (též **integer**) – rozsah -2 147 483 648 až 2 147 483 647 (32 bitů)
- **bigint** – rozsah -9 223 372 036 854 775 808 až 9 223 372 036 854 775 807 (64 bitů)

s desetinnou čárkou

- **float** (někdy též **real**) – plovoucí desetinná čárka podle [IEEE 754](#), rozsah 32 (u **real** též 48) bitů
- **double** (v [PostgreSQL](#) **double precision**) – jako **float** ale s dvojitou přesností, rozsah (64 bitů)
- **decimal** (někde též **numeric**) – pevná desetinná čárka, vhodné pro ukládání [měny](#) nebo [racionálních čísel](#) s omezeným rozsahem a přesností
- **money** (s variantou **smallmoney**) – datový typ s pevnou desetinnou čárkou, používaný v MSSQL speciálně pro ukládání měny

Booleovské typy

- **bit**, **bool** nebo **boolean** – dokáže uložit pouze jeden [bit](#) informace; je použit jen v některých databázích, např. Microsoft Access a MSSQL. Prakticky odpovídá typu **enum** s hodnotami **false** a **true** (resp. 0 a 1).

Textové typy – především pro použití v databázi MySQL (nám bohatě stačí znalost typu **String**)

- **char** – textový typ pevně stanovené šířky (počtu znaků), které zabírá nezávisle na tom, je-li vyplněna nebo ne (v takovém případě je v datovém souboru databáze doplněna na tuto šířku). Výhodou je rychlejší manipulace a přepisování hodnot, nevýhodou je toto nadbytečné místo.
- **varchar** (zkratka z **variant char** nebo **char varying**) – *náš String, který používáme ve Flashi, v PHP žádnou deklaraci neprovádíme, přiřazením \$jmeno="Petr" je automaticky proměnná jméno nastavena jako String (řetězec), na rozdíl od typu char se ukládané hodnoty nedoplňují na zadanou šířku, ale v datovém souboru se uloží délka a za ní data; délka je tedy pohyblivá (odtud označení variant).*

- **text** (původně **clob**, řidčeji **smalltext**, u starších databází **memo**) – textový typ, který se na rozdíl od typů **char** a **varchar** ukládá do oddělených souborů (závisí na implementaci). Typ **text** je omezen na 65536 bajtů.
- **tinytext** – jako **text** s omezením na 256 bajtů
- **mediumtext** – jako **text** s omezením na 16777216 bajtů
- **longtext** – jako **text** s omezením (podle implementace) na 2 nebo 4 [GB](#)

Deklarace proměnných obecně

Deklarace proměnné je instrukcí, která říká programu, jaké má proměnná jméno a na základě datového typu jí vymezí prostor v paměti. Deklaraci program provede sám (implicitní deklarace) nebo ji provedeme sami zápisem v kódu (explicitní deklarace). Konstanty je také třeba deklarovat.

Pole je datová struktura, která obsahuje více položek stejného datového typu za sebou. K jednotlivým položkám se přistupuje pomocí **indexu**, což je celočíselná proměnná. První prvek pole má index 0, poslední pak $n-1$ v případě, že má pole n prvků.

Pomocí indexu se s polem jednoduše pracuje, můžeme získat prvek pole s daným indexem, jednoduchý je také průchod pole prvek po prvku - zvyšujeme index a přistupujeme k prvkům.

Deklarace pole vypadá následovně:

```
datovýTyp[] názevPole;
```

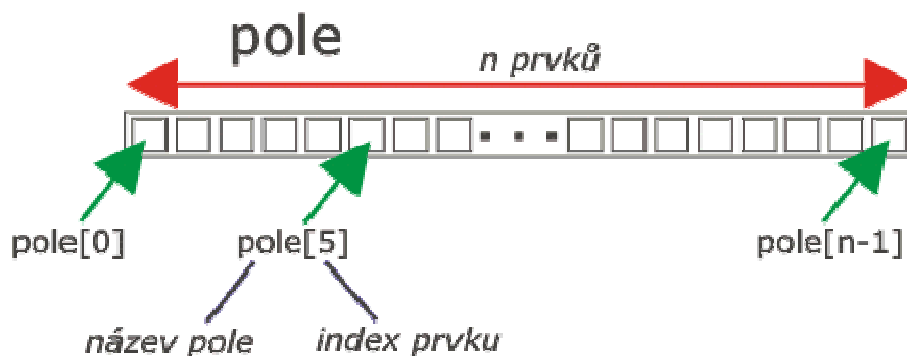
Pokud však požadujeme vytvořit i paměť pro prvky pole, nejen referenci (odkaz), musíme postupovat následovně:

```
datovýTyp[] názevPole = new datovýTyp[početPrvků];
```

K prvku s indexem i přistoupíme následovně:

```
názevPole[i];
```

Přístup k prvkům pole demonstrujeme na následujícím obrázku:



Deklarace proměnné

Bez proměnných se neobejde žádný skript. Ani PHP není výjimkou. Proměnné deklarujeme pomocí znaku dolaru \$ (zapišete ho buď pomocí [PRAVÝ ALT]+[Ú] nebo [SHIFT]+[4/Č]):

```
$dolar = "1$";  
$den="středa";  
$datum =          14;
```

Název proměnné se uvádí hned za značku \$ (bez mezery). Zde jsme tedy zavedli proměnné: dolar, den a datum. Jejich hodnoty jsou uvedené za rovnítkem. Před rovnítkem a za ním nemusí, ale může být mezera (i více). Pokud proměnnou tvoří číslo, se kterým se bude později počítat (sčítat, dělit atd.), musí být zapsáno bez uvozovek, jinak by jej server pokládal za text. Název proměnné nesmí začínat číslem (jinak může čísla obsahovat), toto je chybné \$1promenna = "hodnota". Při zavádění a vyvolávání proměnných také musíte respektovat velikost písmen. Toto nefunguje:

```
$prom = "abc";  
echo $Prom;
```