Programování

Datové typy, základní datové struktury, operátory

Opakování - proměnná

- = pojmenované místo v paměti
- Vytváří se na základě deklarace
- Během deklarace určíme její jméno (identifiáktor) a její typ
  - Volitelně můžeme již v průběhu deklarace určit její hodnotu
- Viditelnost proměnné
  - Lokální – viditelná pouze v určité části programu
  - Globální – viditelná v celém programu
- Např.: int suma = 0; double height; char input;

Datové typy – celočíselné datové typy

Datový typ	Rozsah	Velikost
sbyte	-128 až 127	8 bitů
byte	0 až 255	8 bitů
short	-32 768 až 32 767	16 bitů
ushort	0 až 65 535	16 bitů
int	-2 147 483 648 až 2 147 483 647	32 bitů
uint	0 až 4 294 967 295	32 bitů
long	-9 223 372 036 854 775 808 až 9 223 372 036 854 775 807	64 bitů
ulong	0 až 18 446 744 073 709 551 615	64 bitů

Datové typy – desetinná čísla

Datový typ	Rozsah	Přesnost
float	$+-1.5 * 10^{-45}$ až $+-3.4 * 10^{38}$	7 čísel
double	$+-5.0 * 10^{-324}$ až $+-1.7 * 10^{308}$	15-16 čísel

Datové typy - ostatní

Datový typ	Rozsah	Velikost/Přesnost
char	U+0000 až U+ffff	16 bitů
decimal	$+-1.0 * 10^{-28}$ až $+-7.9 * 10^{28}$	28-29 čísel
bool	true nebo false	8 bitů

Základní datové struktury - pole

- = posloupnost proměnných stejného typu (homogenní datový typ)
- V paměti je uložené v nepřetržité řadě
- Při deklaraci určujeme i počet prvků, které pole bude obsahovat
- Jednotlivé prvky jsou indexovány (!!! Indexujeme od 0 !!!)
- První prvek se nachází na indexu 0
- Jednorozměrné pole – běžné pole, řada hodnot
- Vícerozměrné pole – reprezentace matice (dvourozměrné)
- Např.: int[] cisla; int [][] matice;

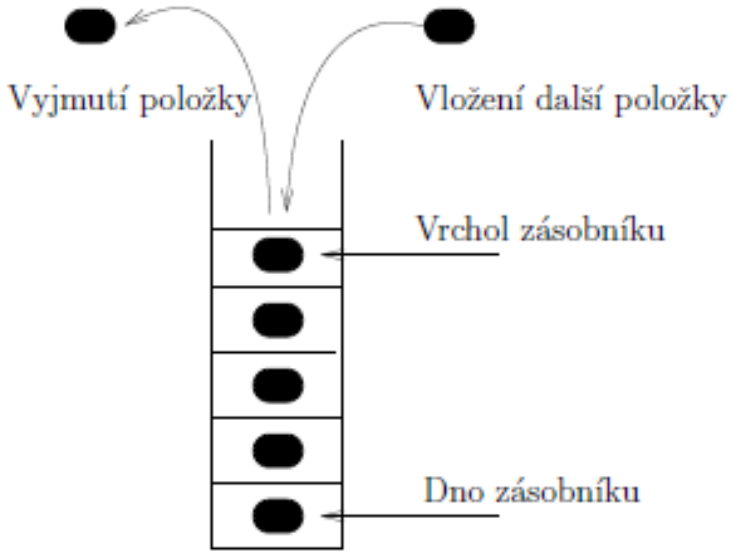
Základní datové struktury – Seznam (List)

- Datová struktura reprezentující posloupnost složek
- Jednotlivé položky jsou řazeny podle určitého klíče, parametru
  - Často se používá index stejně jako u pole
- Oproti poli má dynamickou (proměnitelnou) velikost v paměti
- Složky nemusí za sebou v paměti
  - **Spojovaný seznam** – součástí položky v listu je odkaz na následující položku

- List <T> seznam

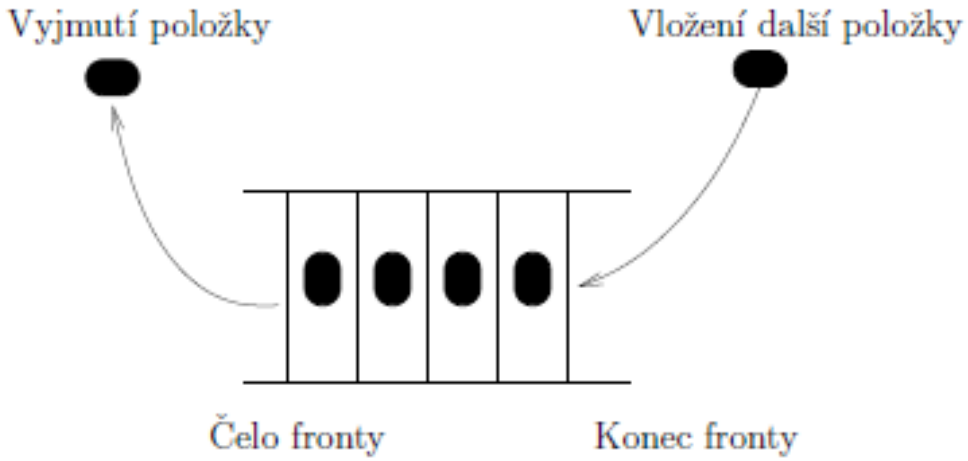
Základní datové struktury – zásobník (LIFO)

- Datová struktura, do které „odkládáme“ položky v průběhu programu
- **LIFO** = Last In First Out
- Data vybíráme v obráceném pořadí, než v jakém je vkládáme
- Vrchol zásobníku = poslední vložená položka



Základní datové struktury – fronta (FIFO)

- Struktura podobná zásobníku s jiným chováním
- FIFO = First In First Out
- Data odebírám v tom pořadí, v jakém jsme je vložili
- Čelo fronty = první položka ve frontě



Operátory

- Aritmetické
  - Operace s čísly
- Relační/porovnávací
  - Operace sloužící k porovnávání hodnot
- Logické
  - Rozhoduji o pravdivosti daných tvrzeních
- Bitové operátory

Aritmetické operátory

Operátor	Název	Účel	Příklad užití
++	inkrement	Zvýšení hodnoty	++x, x++
--	dekrement	Snížení hodnoty	--y, y--
+		Sečtení dvou hodnot	3 + 2
-		Odečtení dvou hodnot	5 - 6
*		Násobení dvou hodnot	7 * 3
/		Celočíselné dělení	8 / 2
%	modulo	Zbytek po dělení	8 % 6

Relační / porovnávací operátory

Operátor	Účel	Příklad užití
<	Vrátí hodnotu True, pokud je první hodnota menší než druhá hodnota	X < Y
<=	Vrátí hodnotu True, pokud je první hodnota menší nebo rovná druhé hodnotě	X <= Y
>	Vrátí hodnotu True, pokud je první hodnota větší než druhá hodnota	X > Y
>=	Vrátí hodnotu True, pokud je první hodnota větší nebo rovná druhé hodnotě	X >= Y
==	Vrátí hodnotu True, pokud je první hodnota rovná druhé hodnotě	Y == X
!=	Vrátí hodnotu True, pokud se první hodnota nerovná druhé hodnotě	X != Y

Logické operátory

Operátor	Název	Účel	Příklad užití
!	Negace	Obrací logickou hodnotu výroku	X!
|	OR	Vyhodnotí logickou podmínku **nebo** mezi dvěma výrazy	X | Y
||	Podmíněný OR	Vyhodnotí logickou podmínku **nebo** mezi dvěma výrazy	X || Y
&	AND	Vyhodnotí logickou podmínku **a zároveň** mezi dvěma výrazy	X & Y
&&	Podmíněný AND	Vyhodnotí logickou podmínku **a zároveň** mezi dvěma výrazy	Y && X
^	Exkluzivní OR / XOR	Vyhodnotí logickou podmínku **buď a nebo** mezi dvěma výrazy	X ^ Y

Bitové operátory

Operátor	Název	Účel
~	Bitový doplněk	Převrácení všech bitových hodnot
<<	Bitový posun vlevo	Posunutí bitové hodnoty vlevo o příslušný počet míst
>>	Bitový posun vpravo	Posunutí bitové hodnoty vpravo o příslušný počet míst
&	Bitový součin	Vytvoří bitový součin mezi dvěma hodnotami
|	Bitový součet	Vytvoří bitový součet mezi dvěma hodnotami
^	Bitový XOR	Vytvoří bitový XOR mezi dvěma hodnotami