Cvičenie č. 5

Náplň:

- Knižnica STDIO.H
- Premenné
- Údajové typy

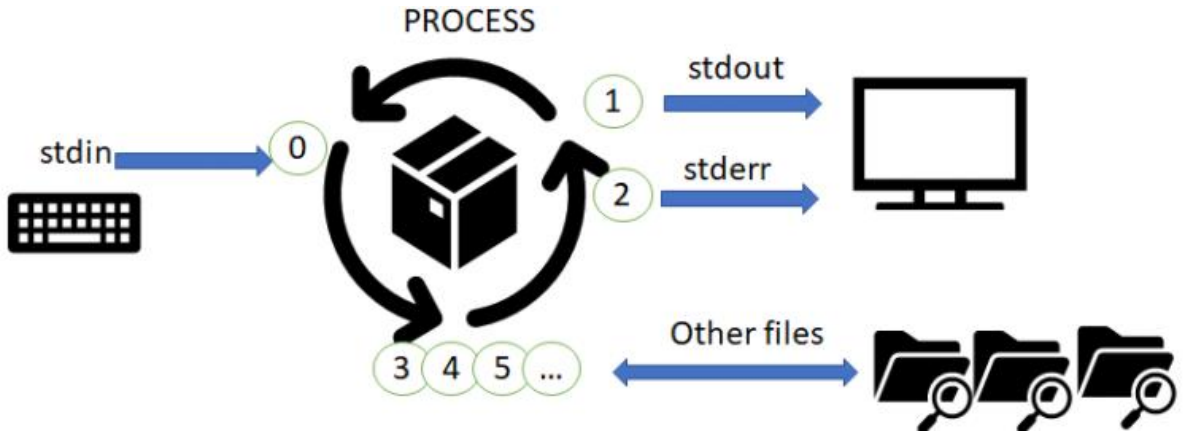
Ing. Ondrej Kováč, PhD.

Knižnica STDIO

Knižnica stdio obsahuje funkcie štandardného vstupu a výstupu, prípadne výstupu pre chybové hlásenie. Skratka pochádza zo slov **Sta**ndard **I**nput **O**utput. Pokiaľ ju chceme použiť, je potrebné v hornej časti .c súboru zaviesť hlavičkový súbor **stdio.h**.

#include <stdio.h>

- Štandardným vstupom môže byť súbor, klávesnica, sériová linka a podobne.
- Štandardným výstupom môže byť súbor, monitor, sériová linka a podobne.



Niektoré funkcie:

Formátovaný text

- int **printf(**const char *format, ...**) – Výpis na štandardný výstup (monitor)**
- int **sprintf(**char *str, const char *format, ...**) – Vygenerovanie formátovaného textu do premennej**
- int **scanf(**const char *format, ...**) – Načítanie formátovaného textu zo štandardného vstupu (klavesnica)**

Jeden znak

- int **getchar(**void**) – Načítanie jedného znaku zo štandardného vstupu (klavesnica)**
- int **putchar(**int char**) – Vypýsanie jedného znaku zo štandardného vstupu (klavesnica)**

Funkcia Scanf()

Funkcia **scanf()** je veľmi užitočná, ale aby sme predišli častým chybám je potrebné mať na pamäti niektoré veci.

Ako ju teda použiť ?

scanf("%<údajový _typ> ", &<názov_premennej>)

Príklad:

scanf("%d ", &vek);

Funkcia Scanf()

Čo teda mám dať za znak %?

scanf(“%<údajový _typ> ”, &<názov_premennej>)

%d – Celé číslo (int)
%f – Reálne číslo (float)
%c – Znak (char)
%s – Reťazec znakov

%iné – samoštúdium ...

Funkcia printf()

Ako ju použiť ?

printf(“Ľubovoľný text %<údajový _typ> iný text ... ”, <názov_premennej>);

Príklad:

printf(“Máš %d rokov, si teda dospelý ”, vek);

Funkcia printf()

Pokiaľ chceme výstup formátovať, je potrebné poznať riadiace znaky, ktoré uvádzame za spätné lomítko **“\”**.

\n – Nový riadok
\t – Tabulátor

Pokiaľ chceme, aby vypísané číslo malo definovanú minimálnu šírku (teda zobrazí definovaný počet znakov), stačí za znak % dať číslo definujúce počet miest napr.

%4d

Pri reálnom čísle, môžeme tiež zadať počet zobrazovaných desatinných miest.

%.4f

Premenné

Premenná je pomenované miesto v pamäti určené pre uloženie hodnoty pre neskoršie použitie. V závislosti od pohľadu na premennú, vieme premenné rozdeliť do viacerých podskupín. Pre nás je zatiaľ dôležité premenné rozlišovať hlavne podľa **Viditeľnosti.**

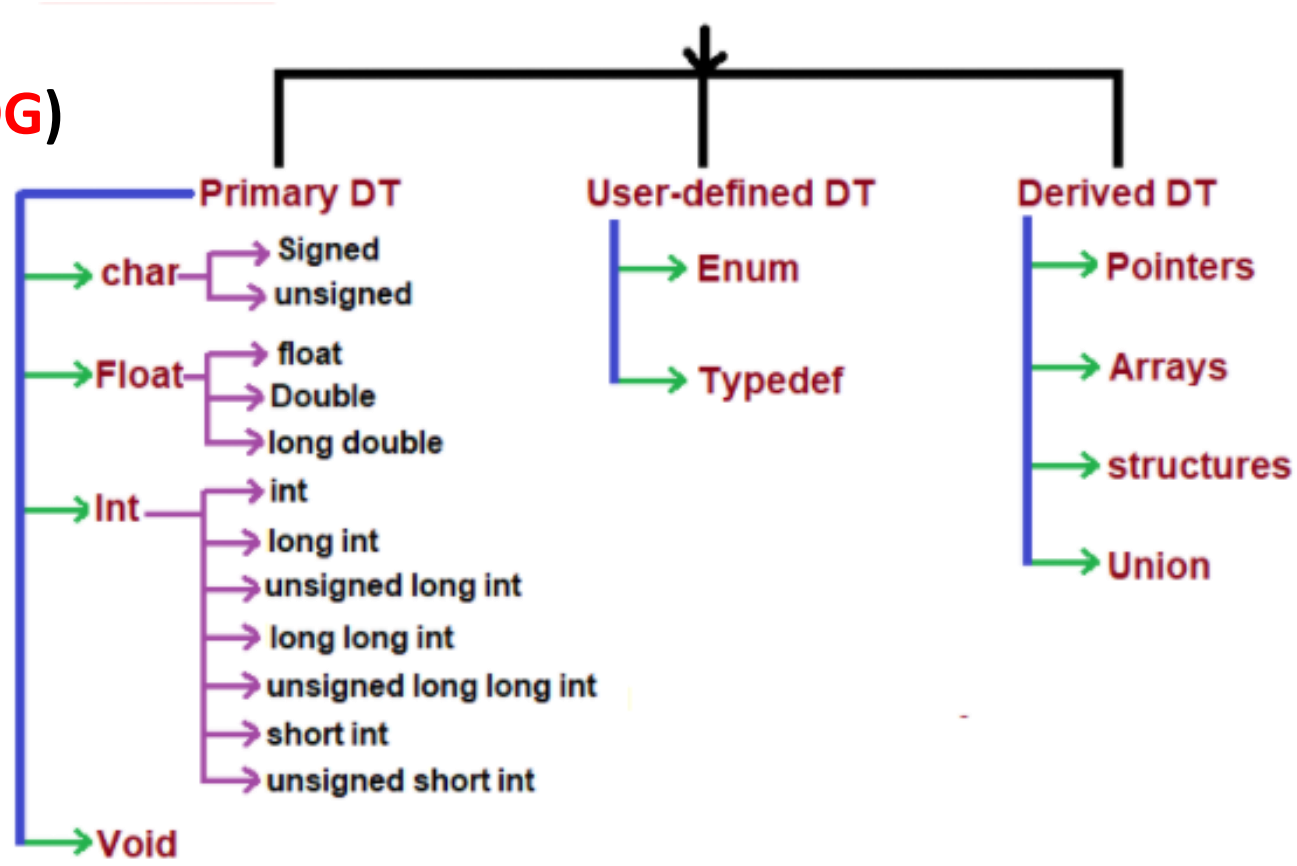
- **Globálna** – Každá funkcia programu k nej má prístup – **Nebezpečné -> Zákaz**
- **Lokálna –** Iba funkcia, v ktorej je deklarovaná k nej môže pristupovať

V jazyku C je každá premenná premennou nejakého **údajového typu**.

Údajové typy v jazyku C

V jazyku C rozlišujeme tri kategórie údajových typov:

- **Primárne, resp. základné (ZAP)**
- **Definované používateľom (1. r. LS – PROG)**
- **Odvodené (1. r. LS – PROG)**



Základný údajový typ

Sú to základné aritmetické typy, ktoré je možné ďalej rozdeliť na

- **Celočíselné (Integers, characters)**
- **Reálne - neceločíselné (Floating-Point)**

Type	Storage size	Value range
char	1 byte	-128 to 127 or 0 to 255 or a-z A-Z
unsigned char	1 byte	0 to 255 or a-z A-Z
signed char	1 byte	-128 to 127
int	2 or 4 bytes	-32,768 to 32,767 or -2,147,483,648 to 2,147,483,647
unsigned int	2 or 4 bytes	0 to 65,535 or 0 to 4,294,967,295
short	2 bytes	-32,768 to 32,767
unsigned short	2 bytes	0 to 65,535
long	8 bytes	-9223372036854775808 to 9223372036854775807
unsigned long	8 bytes	0 to 18446744073709551615

Základný údajový typ

Základné aritmetické typy, je možné ďalej rozdeliť na **Celočíselné** (Integers, characters) **a Reálne - neceločíselné** (Floating-Point)

Tab. 1: Celočíselné údajové typy

Type	Storage size	Value range
char	1 byte	-128 to 127 or 0 to 255 or a-z A-Z
unsigned char	1 byte	0 to 255 or a-z A-Z
signed char	1 byte	-128 to 127
int	2 or 4 bytes	-32,768 to 32,767 or -2,147,483,648 to 2,147,483,647
unsigned int	2 or 4 bytes	0 to 65,535 or 0 to 4,294,967,295
short	2 bytes	-32,768 to 32,767
unsigned short	2 bytes	0 to 65,535
long	8 bytes	-9223372036854775808 to 9223372036854775807
unsigned long	8 bytes	0 to 18446744073709551615

Základný údajový typ

Základné aritmetické typy, je možné ďalej rozdeliť na **Celočíselné** (Integers, characters) **a Reálne - neceločíselné** (Floating-Point)

Tab. 2: Reálne údajové typy

Type	Storage size	Value range	Precision
float	4 byte	1.2E-38 to 3.4E+38	6 decimal places
double	8 byte	2.3E-308 to 1.7E+308	15 decimal places
long double	10 byte	3.4E-4932 to 1.1E+4932	19 decimal places

Iné údajové typy

V jazyku C existujú ešte ďalšie údajové typy

- **Void –** Typ hovoriaci o tom, že nie je dostupná žiadna hodnota (tento už poznáme z volania funkcie bez návratovej hodnoty)
- **Enumeračný –** Typ, ktorého premenná nadobúda iba presne definované hodnoty
- **Pokročilý údajový typ –** Smerník, Pole, Štruktúra, Union ...

Viac sa dozviete v lete na predmete Programovanie v jazyku C (PROG).

Samostatná práca

Program **Eur2SKK.c**

Spomínate si ešte na koruny? Dnes už málokto z mladej generácie rozmýšľa v korunách. Starší ľudia si na eurá ešte nezvykli, a aby mali predstavu o hodnote kupovanej veci, potrebujú urobiť prepočet konverzným kurzom:

1€ = 30,126SKK

Napíšte funkciu, ktorá zo štandardného vstupu od používateľa načíta sumu v Eurách a následne ju prevedie na hodnotu Slovenských korún.

Algoritmus tohto programu môže vyzerať nasledovne:

- **Výzva na zadanie sumy**
- **Uloženie zadanej hodnoty do premennej** (celočíselná alebo reálna?)
- **Volanie funkcie pre prepočet**
- **Vypísanie hodnoty v SKK**
- **Koniec**