

Základy programovania

25/26 ZS

Prednášky, garant: prof. Gabriel Juhás
Cvičenia: Milan Mladoniczky - milan.mladoniczky@paneurouni.com



Tento semester preberieme

- Úvod do jazyka C
- Funkcie, Premenné, dátové typy, I/O a základné aritmetické operácie.
- Podmienky, vetvenia, cykly
- Polia, základné použitie
- Súbory (stdio.h)
- Rekurzia
- Znaky a reťazce (ctype.h, string.h)
- Dynamická alokácia (stdlib.h)
- Štruktúry
- Softvér: životný cyklus programu a projektu



Ako bude prebiehať výučba

1. **Prednáška** – na prednáške odznie príslušné učivo. Pôjde o rôzne programátorské koncepty a ich použitie.
2. **Cvičenie** – na cvičeniach budete samostatne riešiť úlohy, ktorých cieľom je precvičiť koncepty, ktoré odzneli na prednáške.
3. **Samoštúdium** – ak nestihnute vyriešiť všetky úlohy z cvičení, odporúčam venovať sa im vo voľnom čase a na ďalšom cvičení sa opýtať na to, čo ste nevedeli/nerozumeli.



Podmienky absolvovania

- **40 bodov počas semestra**
 - 2 zadania na vypracovanie po 20 bodov
 - na vypracovanie bude 7 dní
 - 1. zadanie na cvičení v 5. týždni semestra
 - 2. zadanie na cvičení v 10. týždni semestra
- **60 bodov skúška** - praktická, zadanie na programovanie
- **Pre úspešné absolvovanie predmetu je potrebné:**
 - **odovzdať obe zadania**
 - **získať 20 bodov zo zadaní počas semestra**
- Výsledná známka bude určená na základe platnej klasifikačnej stupnice PEVŠ



Materiály

- Prednášky budú nahrávané a záznam z nich zdieľaný na verejne dostupnom YouTube kanály - <https://www.youtube.com/@INTERESGroup>
- Materiály použité v prednáškach a na cvičeniach budú zverejnené na stránke predmetu <https://zapr.interes.group>

Programovať sa môžete naučiť iba tak,
že

BUDETE VEĽA PROGRAMOVAŤ !

Treba si to odsedieť za počítačom a
preriešiť
veľa príkladov !



Čomu sa budeme venovať

Budeme sa učiť ako písať počítačové programy.

Ak chceme, aby počítač niečo vykonal, musíme mu krok po kroku povedať, čo má robiť.
(programovanie je tak trochu ako písanie receptov na varenie :)

***Počítač má jednu dobrú vlastnosť:
Vždy urobí presne to, čo mu povieťe.
Ale má aj jednu zlú vlastnosť:
Vždy urobí presne to, čo mu povieťe.***



Cieľom predmetu je **naučiť sa základné programátorské koncepty**.

Pre ich aplikáciu sme vybrali jazyk C.

Avšak cieľom predmetu **nie je** naučiť Vás C či iný konkrétny programátorský jazyk.



Nástroje na vývoj

- V rámci predmetu je budeme používať textový editor podľa voľby a kompilátor **GCC** (prípadne **MinGW** pre Windows).
- Implementačné úlohy budú v jazyku C v štandarde C17
- Odporúčané vývojové prostredia:
 - **CLion** - Licencia od JetBrains pre študentov zadarmo
 - **VS Code** - potrebné nainštalovať rozšírenie/extension pre C/C++
 - Online kompilátor <https://onecompiler.com/>



Jazyk C

- **Kompilovaný jazyk**
 - Program sa najprv musí skompilovať do tzv. strojového kódu, t.j. inštrukciám, ktorým rozumie procesor.
- C je pre všeobecné použitie, čiže v ňom môžete teoreticky naprogramovať čokoľvek
 - množstvo vecí čo používať fičí na C, prípadne C++ (napríklad váš OS)
- Publikovaný v 1972 Denisom Ritchie (C, Unix, Turinogova cena a mnoho ďalších, □ 2011)
- Budeme používať **štandard C17**
 - je aj C23 (ale nie je ešte úplne všade podporované)
- **Imperatívny, procedurálny, štruktúrovaný, staticky typový**
- z C vychádza skoro každý jazyk čo dnes používame: C++, C#, Shell, Go, Java, JavaScript, PHP, Python, Swift, Ruby, Rust a mnoho ďalších...

```
#include <stdio.h>
```

```
int main() {  
    printf("Hello World");  
    return 0;  
}
```



Program

- **Program je postupnosť inštrukcií** (príkazov), ktorá špecifikuje, ako sa má vykonať nejaký výpočet. Takto zapísanú postupnosť príkazov nazývame aj zdrojový kód.
- Výpočet môže byť matematický (napr riešenie sústavy rovníc).
- Výpočet môže byť implementácia nejakého algoritmu (hľadanie reťazca v texte, šifrovanie dát, triedenie hodnôt od minima po maximum, generovanie textu pomocou umelej inteligencie, ...)
- Výpočet je v podstate vykonanie ľubovoľnej postupnosti príkazov.



Program

- Každý programovací jazyk obsahuje nějakú množinu inštrukcií, ktoré je možné použiť na programovanie.
- Každá takáto inštrukcia niečo prikazuje počítaču – čo má robiť.
- Rôzne programovacie jazyky (Python, C, Java, ...) majú rôzne inštrukcie (príkazy), avšak vo všeobecnosti každý programovací jazyk obsahuje nasledovné typy príkazov (inštrukcií)



Príkazy (inštrukcie programu)

- **Vstup** – príkazy, pomocou ktorých je možné načítať nejaké dáta do programu od používateľa (napr. z klávesnice, z disku, atď.)
- **Výstup** – príkazy, pomocou ktorých je možné zobrazíť dáta z programu (napr. na obrazovku, vytlačiť, uložiť na disk, poslať po sieti, atď.)
- **Aritmetické inštrukcie** (s dátami v programe je možné vykonávať matematické operácie súčet, súčin, rozdiel, podiel, atď.)
- **Volania iných funkcií** / podprogramov
- **Podmienky** (vetvenie) – výpočet môže pokračovať rôznymi cestami podľa toho, či platí nejaká podmienka
- **Opakovanie** (iterácia) – časti programu sa môžu opakovane vykonávať.



Výstup, či ako niečo vypísať na obrazovku

```
int printf ( const char * format, ... );
```

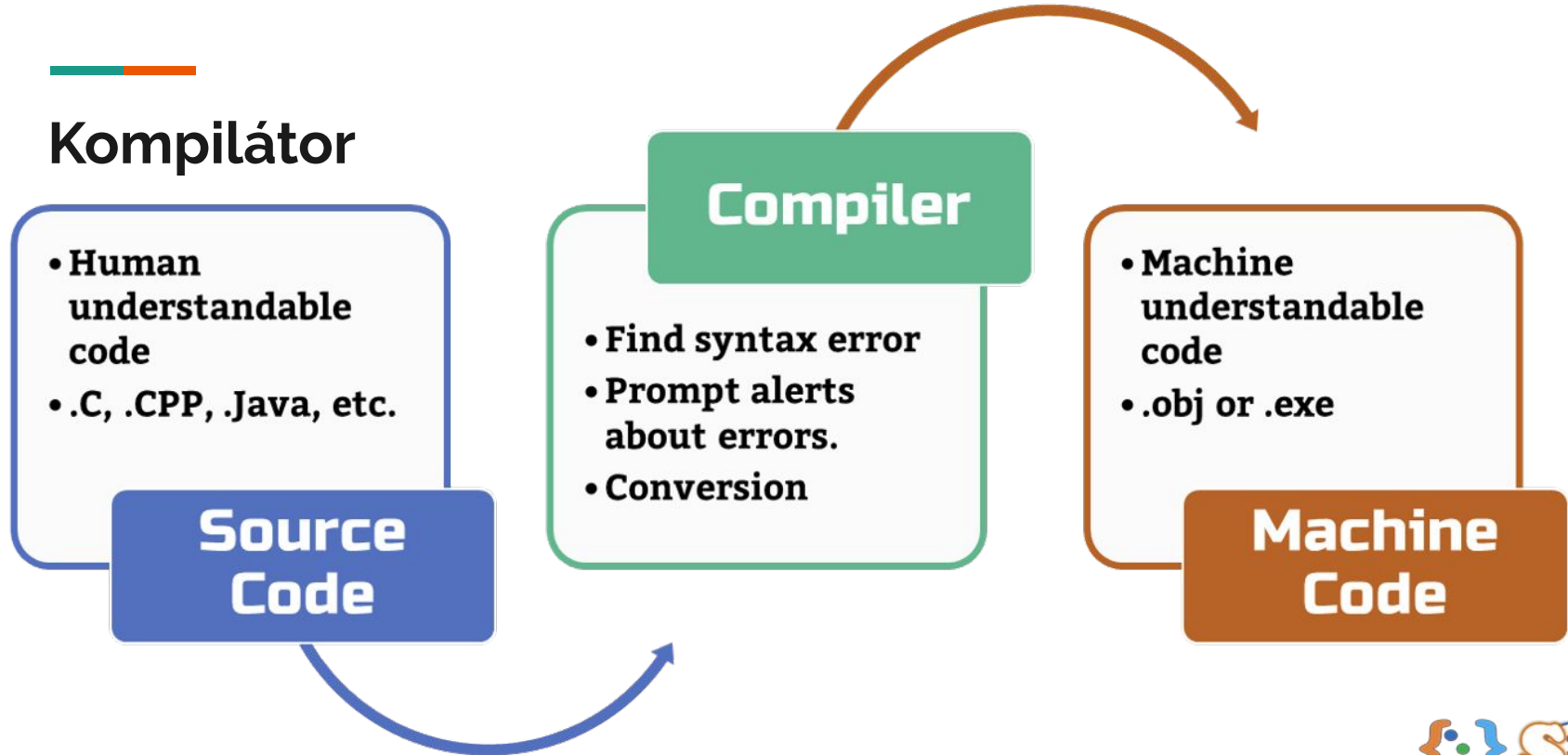
```
printf("Hello World");
```



Formátovanie výpisu

- V rámci reťazca je možné uviesť znak % pre možnosti formátovania
 - **%d** - výpis čísla v desiatkovej sústave
 - **%x** - výpis čísla v hexadecimálnej sústave
 - **%s** - výpis iné reťazca znakov
 - **%c** - výpis jedného znaku

Kompilátor



Prejdime k praktickej ukážke ->



Aritmetické operátory

- Symbol, ktorý predstavuje danú operáciu, nazývame **operátor**
- Hodnoty, s ktorými sa operácia vykonáva, nazývame **operandy**
- Napríklad v zápise: **1 + 2**
 - **+** je operátor (sčítania)
 - Hodnoty **1 a 2** sú operandy



Aritmetické operátory

- + pre súčet,
- - pre rozdiel
- * pre súčin,
- / pre podiel
- % pre zvyšok po delení
- () zátvorky pre ozátvorkovanie výrazu
- a ďalšie



Hodnoty a ich typy

- V uvedených príkladoch sme pri použití (volaní) funkcie ***printf()*** do zátvoriek napísali hodnotu, aká sa má vypísať na obrazovku.
- Hodnota je vo všeobecnosti nejaký údaj, s ktorým môže program pracovať. V našich príkladoch spočívala „práca“ s týmito hodnotami v tom, že sa vypísali na obrazovku pomocou `printf()`.
- V uvedených príkladoch sme mali 3 typy hodnôt:
 - celé číslo (v C int od slova integer)
 - číslo s pohyblivou čiarkou (v podstate desatinné číslo, v C float od slova floating-point)
 - reťazce znakov (postupnosť znakov, v C reťazce ako typ neexistujú ale o tom neskôr).



Úskalie programovacích jazykov

- Programovacie jazyky vyžadujú, aby v nich napísané programy dodržiavali všetky stanovené pravidlá (tzv. syntax).
- Ak to nedodržíte, program sa nebude dať korektne spustiť.
- Našťastie, C kompilátor vás o tom vždy informuje, čiže:

Čítajte chybové výpisy!!!



int premenné =

- Počítač samozrejme dokáže realizovať rôzne výpočty na dátach, t.j. pracovať s rôznymi hodnotami a zmeniť ich.
- Pre jednoduchšiu prácu s dátami v pamäti počítača používame premenné.
- Premenná je v podstate miesto v pamäti počítača, ktoré má svoje meno a nejakú hodnotu.
- Keď chceme s touto hodnotou pracovať, tak k nej pristúpime pomocou mena príslušnej premennej.
- V jazyku C je potrebné pre použitie premennej ju najprv deklarovať:

```
typ_premennej meno_premennej = hodnota
```



// Komentáre

- Pri programovaní je často užitočné vkladať si do kódu tzv. komentáre
- Komentár je slovný popis toho, čo vykonáva nejaká časť kódu.
- C pri spracovaní kódu vie, že komentár je len pre informáciu, a teda ho pri vykonávaní programu ignoruje!
- Aby toto však C vedel, je nutné komentár označiť!
- Komentáre slúžia len pre programátorov, aby lepšie popisovali, čo ktorá časť kódu robí!
- Komentár je označená znakmi `//` na začiatku komentára



Vstup, či ako získať dáta

- V rámci programov na precvičenie budeme často používať príkaz na zadanie vstupu od používateľa.

```
int scanf ( const char * format, ... );
```

```
int a;
```

```
scanf ("%d", &a) ;
```

Podíme na cvičenia —>