

První počítačové cvičení

Základy programování (IZP)

Brno University of Technology, Faculty of Information Technology
Božetěchova 1/2, 612 66 Brno - Královo Pole
Petr Veigend, veigend@fit.vut.cz



- Jmenuji se **Petr Veigend, studijní poradce**
- **Můj profil:** <https://www.fit.vut.cz/person/veigend/>
 - Kancelář: A305
 - Konzultační hodiny: po domluvě emailem, případně na kartě
 - Slidy na vizitce
- **Komunikace:**
 - **email – prosím používejte předmět:**
IZP - <předmět emailu>
 - *Discord, ..., (ale hůř se dohledává historie a není přesně jasné kdo se ptá, takže „oficiální věci“ raději mailem ...)*
- **Přestávky?**

- Cvičení bude během semestru **10**
- Bodování cvičení (**celkem 10b**)
 - První dvě **se nebodují**
 - Dalších šest je za **jeden bod**
 - Poslední dvě jsou za **dva body**
- Pro získání zápočtu je nutné získat
 - **Alespoň 6 bodů** ze cvičení
 - **Alespoň 1 bod** z každého projektu
 - a během semestru musíte získat **minimálně 23 bodů**

- **FIT:** <https://www.fit.vut.cz/>
- **EMAIL:** <http://roundcube.fit.vut.cz/>
- **Karta předmětu IZP:**
 - <https://www.fit.vut.cz/study/course/IZP/>
- **Materiály jsou na Moodle IZP (proklik z IS VUT)**

- Nápovědu můžete získat
 - z **internetu**: vygooglit název příkazu + c
 - Výborný zdroj: <http://www.cplusplus.com/>
 - Doporučení AS: <http://devdocs.io/>
 - pomocí **příkazu man** (linux)
 - např. pokud chcete získat informace o funkci **printf**, zadejte:

```
man 3 printf
```

- **Kniha**
 - Herout, P. Učebnice jazyka C.
 - Mohla by ještě být v knihovně

- **Vývojové nástroje**
 - Pokud s programováním začínáte, **nepoužívejte** žádné složité nástroje
 - Můžete pracovat ve Windows, Linux, Mac OS, ...
 - V laboratořích budeme pracovat ve Windows, ale vyzkoušíme i Linux
 - Doporučené vývojové prostředí (IDE): **Visual Studio Code**
- **Jednoduché příklady**

- Pro emulaci Linux na Windows 10 a 11 lze využít WSL
 - Pro práci s VS Code bych to doporučoval, v laboratořích to nevyužijeme, ale může se vám to hodit doma

VISUAL STUDIO CODE

- Do nově vytvořeného textového souboru napište následující zdrojový kód

```
#include<stdio.h>

int main()
{
    printf("Hello world\n");
    return 0;
}
```

- Základní pojmy
 - **Proměnná**
 - Deklarace
 - Inicializace
 - **Konstanta**
 - **Operátor přiřazení**
 - **Příkaz**

- Základní pojmy

- **Proměnná**

- Pojmenované místo v paměti, ve kterém uchováváme data.
 - Má určitý datový typ (`int`) a její hodnota se **může** za běhu programu měnit.
 - Příklad: `int a;`

- **Konstanta**

- Pojmenované místo v paměti, ve kterém uchováváme data.
 - Má určitý datový typ (`int`), její hodnota se **nemůže** za běhu programu měnit.
 - Příklad: `const int b=10;`

- **Přiřazení (=)**

- Základní pojmy

- **Deklarace proměnné**

- Vytvoření proměnné tak, aby ji bylo možno dále použít

```
int i; // proměnná typu int, název i
```

- **Inicializace proměnné**

- Nastavení hodnoty proměnné nebo konstanty

```
int i; // proměnná typu int, název i
```

```
i = 10; // deklarovaná proměnná i má hodnotu 10
```

```
int i = 10; // ekvivalentní (inicializace + deklarace)
```

- **Příkaz**

- Definuje činnost, kterou program vykoná (např. výpis textu na obrazovku)

- `printf()` je tzv. knihovní funkce, nachází se v souboru (knihovně funkcí) `stdio.h`

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

- **Výpis konstantního řetězce**

```
printf("text"); // pozor na "
```

- Napište program, který:
 - Proveďte deklaraci a inicializaci tří proměnných typu **int**
 - **a, b, c**
 - Do další deklarované a inicializované proměnné vypočtete diskriminant z těchto tří proměnných
 - $b^2 - 4ac$ **b*b - 4*a*c**
 - Vypočtený diskriminant vypíšte na obrazovku

```
// výpočet diskriminantu  
// (např. v int dis)  
printf("Diskriminant: %d", dis);
```

- K načítání vstupů se používá funkce `scanf`
`scanf ("formátovací řetězec", &proměnná)`
- Formátovací řetězec indikuje formát dat, která načítáme např. `%f`, `%d`
- Proměnná, do které budou zadaná data uložena
 - Znak `&`: chceme adresu, na kterou ukládáme
 - Pozor na řetězce/pole
- Příklad:

```
int celeCislo; // kam ukládáme
scanf ("%d", &celeCislo); // načtení
printf ("%d\n", celeCislo); // výpis
```

- Základní pojmy
 - **podmíněný příkaz**

```
if (podmínka)
{
    // podmínka splněna
}
else {
    // podmínka nesplněna
}
```

- V podmínkách budeme potřebovat porovnávat hodnoty (**relační a logické operátory**)

- **Aritmetické:** unární, binární
- **Relační:** ==, !=, >, <, >=, <=
- **Logické:** && (AND), || (OR)

- **Aritmetické:** unární, binární
- **Relační:** ==, !=, >, <, >=, <=
- **Logické:** && (AND), || (OR)

A	B	AND	OR
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

- Operátor % vrací zbytek po celočíselném dělení
 - Operace **modulo**, např.: $5\%5 = 0$, $6\%5 = 1$
- Napište program, který určí, zda bylo načtené číslo sudé.

- Načtení čísla

```
int celeCislo; // kam ukládáme
scanf("%d",&celeCislo); // načtení
printf("%d\n",celeCislo); // výpis
```

- Výpis:

```
printf("Diskriminant: %d", d);
```

- Podmínka

```
if (podmínka) { příkazyT } else {příkazyF}
```

- Napište program, který určí, zda je číslo x v intervalu $\langle a, b \rangle$.
- Varianta 1: pevně zadaný interval
- Varianta 2: uživatelem zadaný interval

- Najděte maximum z čísel 3, 10, 18
- Varianta 1: porovnejte všechny dvojice mezi sebou
- Varianta 2: uchovávejte si aktuálně největší číslo v pomocné proměnné

- Číslo roku je dělitelné 4 a
 - číslo roku není dělitelný 100 → **rok je přestupný,**
 - číslo roku je dělitelný 100 a
 - číslo roku je dělitelný 400 → **rok je přestupný,**
 - číslo roku není dělitelný 400 → **rok není přestupný,**
- Číslo roku není dělitelné 4 → **rok není přestupný.**

- Zkuste stejnou složenou podmínku napsat pomocí logických operátorů `&&`, `||`, `()`

- Přidejte kontrolu `rok >= 1582` (dříve se přestupné roky nepočítaly), pokud selže, vypište chybu a return 1;

- Příklady jsou vyřešeny na Moodle

Děkuji za pozornost

- Napište program, který:
 - Vypíše na obrazovku **číslo 38** (%d)
 - Deklarujte **proměnnou typu int** (může se jmenovat libovolně)
 - Do proměnné uložte hodnotu 38
 - K deklarované proměnné **přičte hodnotu 4**
 - **Opět vypíše** novou hodnotu na obrazovku (%d)

```
int i = 10;  
printf("cislo: %d", i);
```