



# Dátové štruktúry a algoritmy

## 24/25 LS

Prednášky, garant: prof. Gabriel Juhás  
Cvičenia: Milan Mladoniczky - [milan.mladoniczky@paneurouni.com](mailto:milan.mladoniczky@paneurouni.com)



# Tento semester preberieme

- Úvod do jazyka C++
- Triedy, abstraktné dátové typy
- Dátové štruktúry - zásobník, zoznam, fronta, strom, graf
- Triediace algoritmy
- Vyhľadávacie algoritmy
  - stromy
  - grafy
- Zložitosť a optimalizácia algoritmov
- Bonus: Git, Dokumentácia, Testovanie



# 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 nestihnete 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 (s možnosťou predĺženia)
  - 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 budú sprístupnené pomocou MS Teams v sekcií pre súbory v priečinku "**Materiály/Prednášky**"
- Materiály použité v cvičeniach budú sprístupnené pomocou MS Teams v sekcií pre súbory v priečinku "**Materiály/Cvičenia**"
- Úlohy pre jednotlivé cvičenia budú dostupné ???

Programovať sa môžete naučiť iba tak, že

BUDETE VEĽA PROGRAMOVAŤ !

Treba si to odsedieť za počítačom a

preriešiť veľa príkladov !

—



Cieľom predmetu je **naučiť sa koncepty pre efektívne programovanie**.

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 **G++** (prípadne **MinGW** pre Windows).
- Implementačné úlohy budú v jazyku C++ v štandarde C++17 (C++1z)
- Odporúčané vývojové prostredia:
  - **CLion** - Licencia od JetBrains pre študentov zadarmo
  - **VS Code** - potrebné nainštalovať rozšírenie/extension pre C/C++





# 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++ (napríklad vás OS, či herné enginy)
- Publikovaný v 1998 (prvý výskyt v 1985) Bjarne Stroustrup (C++, profesor na univerzite v Kolumbii, spoluzakladateľ ISO Committee)
- Budeme používať **štandard C++17**
  - posledný je C++23 (nemá však úplnú podporu)
  - teraz sa chystá C++26
- **Imperatívny, štruktúrovaný, staticky typový, objektovo-orientovaný, funkcionálny**
- C++ je široko používaný jazyk naprieč všetkými odvetvami. Na jeho základoch vznikli ďalšie jazyky ako Java, Lua, PHP, Rust a ďalšie.

```
#include <iostream>
```

```
int main() {
```

```
    std::cout << "Hello World!";
```

```
    return 0;
```

```
}
```



# Konštanty

Ak je premenná konštanta jej hodnota sa po inicializácii nemení.

```
const int MAGIC_NUMBER = 42;
```



# Input & Output

- Výstup predstavuje objekt ***cout***.
- Vstup predstavuje objekt ***cin***.
- Pre výpis alebo načítanie dát sú použité operátory bitových posunov ***<<*** a ***>>***.
- Objekty sú dostupné v knižnici ***<iostream>***.

```
#include <iostream>

using namespace std;

int main() {

    string input;

    cin >> input;

    cout << "Hello! You typed: " << input << endl;

    return 0;

}
```



# String

- Reťazec znakov ako dátový typ, resp. trieda ***string***.
- Funkcie pre manipuláciu v knižnici ***<string>***.
- Je možné použiť aj polia znakov ako v C (tzv. C-string).

```
char str[] = "C++";
```

```
char str[4] = "C++";
```

```
string str = "Some string";
```



# Dynamická alokácia

- V C++ fungujú pointre rovnako ako v C.
- Alokácia pamäte prebieha operátormi:
  - **new** - Alokácia novej pamäte daného typu.
  - **delete** - Uvoľnenie pamäte

```
int* point_var = new int{45};  
  
cout << *point_var; // out: 45  
  
delete point_var;  
  
point_var = nullptr;
```



# Dynamická alokácia

- V C++ fungujú pointre rovnako ako v C.
- Alokácia pamäte prebieha operátormi:
  - **new** - Alokácia novej pamäte daného typu.
  - **delete** - Uvoľnenie pamäte

```
float* array = new float[5];

for (int i = 0; i < 5; i++) {
    *(array + i) = i + 1;
}

for (int n : array) {
    cout << n << ", ";
}

delete[] array;

array = nullptr;
```



# Trieda

- Triedy sú vlastné zložené dátové typy, ktoré otvárajú možnosti k Objektovo-orientovanému programovaniu.
- Z triedy je vytvorený **objekt** (runtime reprezentácia triedy).
- Každý objekt je vlastná inštancia triedy.
- Trieda môže obsahovať **atribúty** (data members) a funkcie/**metódy** (member functions).

```
class Student {  
    public:  
        string name;  
        int year;  
        string study_program;  
  
    void enroll(int new_year) {  
  
        year = new_year;  
  
    }  
};
```





# Trieda

- Triedy sú vlastné zložené dátové typy, ktoré otvárajú možnosti k Objektovo-orientovanému programovaniu.
- Z triedy je vytvorený **objekt** (runtime reprezentácia triedy).
- Každý objekt je vlastná inštancia triedy.
- Trieda môže obsahovať **atribúty** (data members) a funkcie/**metódy** (member functions).

```
int main(){  
  
    Student milan;  
    milan.name = "Milan";  
    milan.enroll(2);  
  
    cout << milan.name << " is student  
of " << milan.year << ". year at  
university" << endl;  
  
    return 0;  
}
```

---

**Prejdime k praktickej ukážke ->**



**<https://interes-group.github.io/pevs-DSA-BIAX10031>**