WYDZIAŁ
INFORMATYKI
I TELEKOMUNIKACJI

drzewa

Veronika Tronchuk

I rok Informatyki

Nr albumu: 30019

Zadanie 1

Drzewo jest strukturą rekurencyjną, bo wygląda tak samo w każdej części. Każdy węzeł może mieć dzieci, a te dzieci też mogą mieć swoje dzieci i tworzyć kolejne poddrzewa. Dzięki temu całe drzewo jest zbudowane z małych drzew i można je opisywać i przetwarzać w ten sam sposób.

Zadanie 2

```
ceo
  manager 1
    employee A
  anmager 2
    employee B
```

```
struct Node {
    string data;
    vector<shared_ptr<Node>> children;

    Node(const string& d) : data(d) {}
};
```

Struktura węzła: przechowuje tekst (data) i listę dzieci jako wskaźniki smart (shared_ptr). Konstruktor od razu przypisuje nazwę węzła.

```
void addChild(shared_ptr<Node> parent, shared_ptr<Node> child) {
    parent->children.push_back(child);
}
```

Funkcja dodaje jeden węzeł potomny do listy dzieci wskazanego węzła rodzica.

```
void printTree(const shared_ptr<Node>& node, int depth = 0) {
    for (int i = 0; i < depth; ++i) cout << "  ";
    cout << node->data << "\n";
    for (const shared_ptr<Node>& child : node->children) {
        printTree(child, depth + 1);
    }
}
```

Funkcja rekurencyjna do wyświetlania drzewa:
- Wypisuje wcięcia według poziomu węzła (depth).
- Wypisuje nazwę węzła.
- Dla każdego dziecka wywołuje samą siebie z większym poziomem.

```
auto root = make_shared<Node>("ceo");
auto manager1 = make_shared<Node>("manager 1");
auto manager2 = make_shared<Node>("anmager 2");
auto employee1 = make_shared<Node>("employee A");
auto employee2 = make_shared<Node>("employee B");
```

W main tworzony jest korzeń drzewa i kilka węzłów dla menedżerów i pracowników. Używany jest make_shared do automatycznego zarządzania pamięcią.

```
addChild(root, manager1);
addChild(root, manager2);
addChild(manager1, employee1);
addChild(manager2, employee2);
```

Węzły są łączone w drzewo: do korzenia dodaje się menedżerów, do menedżerów — pracowników.

```
printTree(root);
```

Wyświetlenie całej struktury drzewa w konsoli. cin.get(); zatrzymuje zamknięcie programu, aby można było zobaczyć wynik. return 0; kończy działanie.

Zadanie 3

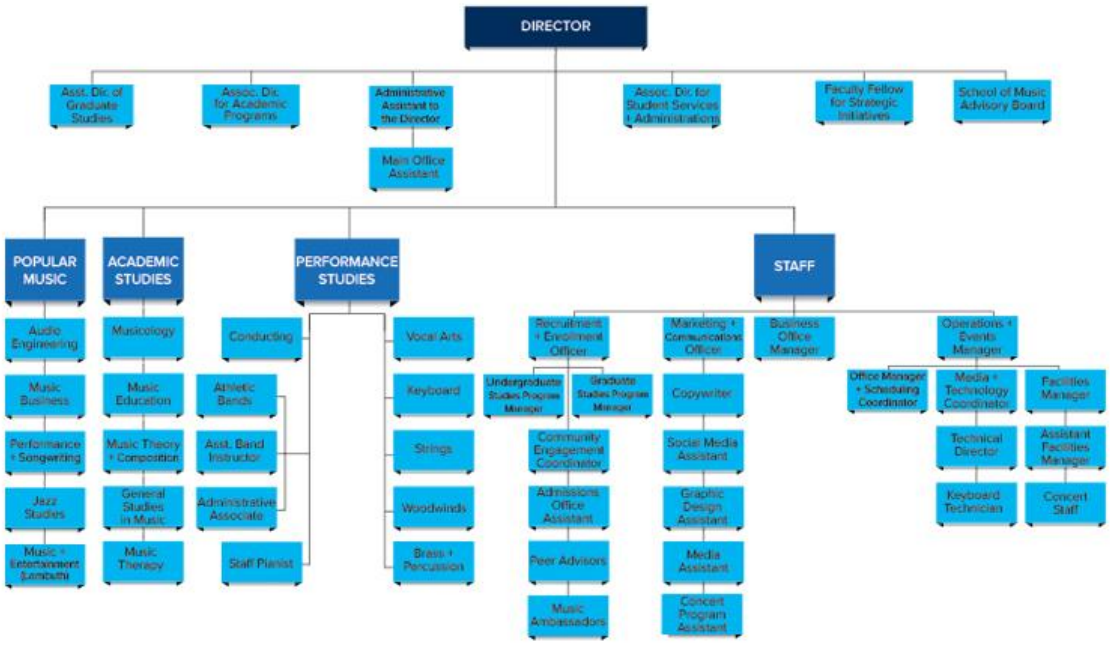
W ramach tego zadania znalazłam/znalazłem rzeczywisty przykład schematu organizacyjnego szkoły muzycznej. Wybrany został Faculty/Staff Handbook Uniwersytetu w Memphis – School of Music.

Schemat przedstawia strukturę zarządzania: dyrektor szkoły, dyrektorzy ds. programowych, koordynatorzy działów, wykładowcy i pracownicy administracyjni. Struktura jasno obrazuje hierarchię, przypisane role oraz podział obowiązków.

Źródło:
https://www.memphis.edu/music/about/facultyresources/facultystaffhandbook_1.php

THE UNIVERSITY OF MEMPHIS.

Rudi E. Scheidt School of Music



```
Main Office Assistant
Assoc. Dir. for Student Services Administrations
Faculty Fellow for Strategic Initiatives
School of Music Advisory Board
Popular Music
  Audio Engineering
  Music Business
  Performance & Songwriting
  Jazz Studies
  Music + Entertainment
Academic Studies
  Musicology
  Music Education
  Music Theory/Composition
  General Studies in Music
  Music Therapy
Performance Studies
  Conducting
  Athletic Bands
  Asst. Band Instructor
  Administrative Associate
  Staff Pianist
Staff
  Business Office Manager
  Operations + Events Manager
  Marketing Communications Officer
  Recruitment + Enrollment Officer
  Technical Director
  Facilities Manager
```

Schemat w programie odwzorowuje główne działy i stanowiska zgodnie z oryginalną strukturą. Struktura jest zgodna

```
 auto director = make_shared<Node>("Director");
```

Tworzy węzeł główny drzewa — dyrektor szkoły muzycznej.

```
auto asstGrad = make_shared<Node>("Asst. Dir. of Graduate Studies");
auto assocAcad = make_shared<Node>("Assoc. Dir. for Academic Programs");
auto adminAsst = make_shared<Node>("Administrative Assistant to the Director");
auto mainOffice = make_shared<Node>("Main Office Assistant");
auto assocStudent = make_shared<Node>("Assoc. Dir. for Student Services Administrations");
auto fellow = make_shared<Node>("Faculty Fellow for Strategic Initiatives");
auto advisoryBoard = make_shared<Node>("School of Music Advisory Board");
```

Tworzy węzły podrzędne: zastępcy dyrektora, asystenci, rada doradcza

```
addChild(director, asstGrad);
addChild(director, assocAcad);
addChild(director, adminAsst);
addChild(director, mainOffice);
addChild(director, assocStudent);
addChild(director, fellow);
addChild(director, advisoryBoard);
```

Dodaje powyższe węzły jako dzieci dyrektora

```
auto popularMusic = make_shared<Node>("Popular Music");
auto academicStudies = make_shared<Node>("Academic Studies");
auto performanceStudies = make_shared<Node>("Performance Studies");
auto staff = make_shared<Node>("Staff");
```

Tworzy główne działy: muzyka popularna, studia akademickie, studia wykonawcze oraz dział personelu.

```
addChild(director, popularMusic);
addChild(director, academicStudies);
addChild(director, performanceStudies);
addChild(director, staff);
```

Dodaje te działy do dyrektora jako kolejne gałęzie.

```
addChild(popularMusic, make_shared<Node>("Audio Engineering"));
addChild(popularMusic, make_shared<Node>("Music Business"));
addChild(popularMusic, make_shared<Node>("Performance & Songwriting"));
addChild(popularMusic, make_shared<Node>("Jazz Studies"));
addChild(popularMusic, make_shared<Node>("Music + Entertainment"));
```

Dodaje poddziały w sekcji Popular Music (np. realizacja dźwięku, biznes muzyczny, songwriting, jazz).

```
addChild(academicStudies, make_shared<Node>("Musicology"));
addChild(academicStudies, make_shared<Node>("Music Education"));
addChild(academicStudies, make_shared<Node>("Music Theory/Composition"));
addChild(academicStudies, make_shared<Node>("General Studies in Music"));
addChild(academicStudies, make_shared<Node>("Music Therapy"));
```

Dodaje poddziały w sekcji Academic Studies (muzykologia, edukacja muzyczna, teoria muzyki, muzykoterapia).

```
addChild(performanceStudies, make_shared<Node>("Conducting"));
addChild(performanceStudies, make_shared<Node>("Athletic Bands"));
addChild(performanceStudies, make_shared<Node>("Asst. Band Instructor"));
addChild(performanceStudies, make_shared<Node>("Administrative Associate"));
addChild(performanceStudies, make_shared<Node>("Staff Pianist"));
```

Dodaje poddziały w sekcji Performance Studies (dyrygentura, zespoły sportowe, pianista zespołowy).

```
addChild(staff, make_shared<Node>("Business Office Manager"));
addChild(staff, make_shared<Node>("Operations + Events Manager"));
addChild(staff, make_shared<Node>("Marketing Communications Officer"));
addChild(staff, make_shared<Node>("Recruitment + Enrollment Officer"));
addChild(staff, make_shared<Node>("Technical Director"));
addChild(staff, make_shared<Node>("Facilities Manager"));
```

Dodaje główne stanowiska w sekcji Staff (kierownik biura, kierownik ds. wydarzeń, marketing, rekrutacja, dyrektor techniczny).

```
printTree(director);
```

Wywołuje funkcję printTree, aby wyświetlić całe drzewo w konsoli.

Wnioski

Wykonane zadanie pokazało, jak można krok po kroku odwzorować strukturę organizacyjną prawdziwej szkoły muzycznej i przedstawić ją w czytelny sposób. Dzięki temu łatwo zobaczyć, jakie są główne działy, kto do kogo należy i jakie stanowiska występują w różnych częściach schematu. Struktura jest przejrzysta i można ją łatwo rozbudować o kolejne elementy, jeśli zajdzie taka potrzeba. Najwięcej trudności sprawiło dokładne ustawienie wszystkich ról tak, żeby zgadzały się z oryginalnym schematem, ponieważ przy dużej liczbie działów łatwo coś pominąć lub pomieszać kolejność. Mimo to zadanie pozwoliło lepiej zrozumieć, jak działają hierarchie w dużych organizacjach i jak ważne jest zachowanie porządku przy planowaniu takiej struktury.