

Задача 1: Да се генерират всички пермутации на дадено множество символи:

- а) чрез рекурсивна функция
- б) чрез стек

И в двата случая функцията да връща вектор, съдържащ пермутациите.

Например за вход "abc" програмата трябва да върне вектор с низовете:

"cba", "cab", "bca", "bac", "acb", "abc" (не задължително в този ред)

в) Да се имплементира клас `PermutationIterator`, позволяващ да се итерира по всички пермутации на зададено множество от символи.

Задача 2:

Да се създаде клас `Taxonomy`, описващ таксономията на животните (корен - царство животни -> тип -> клас -> разред -> семейство -> род -> вид->подвид). Всяко ниво на таксономията да съответства на ниво в дърво. Например върховете на разстояние 3 представляват разред, а видовете са на ниво 6.

1. Да се дефинира подходяща структура `Node` за връх в такова дърво. Всяка категория се идентифицира с латинското си название.
2. Да се имплементира метод `void Taxonomy::addEntity(vector<string> taxonomy)`, който по зададена таксономия добавя нова група или животински вид в дървото.
3. Да се имплементира метод `void Taxonomy::print() const`, който отпечата информацията за видовете с целия път до всеки връх, например ако в таксономията има информация за вида "човек", резултатът трябва да включва: `Animalia->Chordata->Mammalia->Primates->Hominidae->Homo->Homo sapiens`.
4. Да се намери броят листа, чиито имена съдържат поне две думи.
5. Да се имплементира метод `int Taxonomy::countSpeciesInFamily(string family) const`, който намира броят регистрирани видове и подвидове в даденото семейство.
6. Да се имплементира `map<string, int> Taxonomy::speciesPerFamily() const`, който намира по колко вида и подвида има във всяко семейство в дървото.
7. Да се имплементира горният метод чрез стек вместо с рекурсия
8. Да се имплементира рекурсивен метод `vector<string> Taxonomy::listSpecies() const`, който връща списък с всички **видове и подвидове** в таксономията.
9. Да се имплементира горният метод чрез стек вместо с рекурсия