



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ \_\_\_\_\_ «Информатика и системы управления»

КАФЕДРА \_\_\_\_\_ «Теоретическая информатика и компьютерные технологии»

**Лабораторная работа № 12**  
**по курсу «Языки и методы программирования»**  
**«Обработка текстовых файлов»**

Студент группы ИУ9-22Б Тараканов В. Д.

Преподаватель Посевин Д. П.

*Москва 2024*

# 1 Задание

Найти все файлы с расширением «dot» в указанном каталоге, для каждого файла определить, ориентированный или неориентированный граф он описывает, превратить ориентированный граф в неориентированный и сохранить получившееся описание графа в файле с тем же именем, но в текущем каталоге. Работоспособность программы нужно проверить на наборе предварительно созданных DOT-файлов.

# 2 Результаты

## Листинг 1 — Начало кода

```
1  #include <iostream>
2  #include <fstream>
3  #include <filesystem>
4  #include <regex>
5  #include <string>
6
7  namespace fs = std::filesystem;
8
9  bool isDirected(const std::string& content) {
10     std::regex directed_edge("->");
11     return std::regex_search(content, directed_edge);
12 }
13
14 std::string convertToUndirected(const std::string& content) {
15     std::regex directed_edge("->");
16     return std::regex_replace(content, directed_edge, "--");
17 }
18
19 void processDotFile(const fs::path& filePath, const fs::path&
outputPath) {
20     std::ifstream inputFile(filePath);
21     if (!inputFile.is_open()) {
22         std::cerr << "Failed to open file: " << filePath << std::endl;
23         return;
24     }
25
26     std::string content((std::istreambuf_iterator<char>(inputFile)),
27 std::istreambuf_iterator<char>());
28     inputFile.close();
29
30     bool directed = isDirected(content);
31     if (directed) {
32         content = convertToUndirected(content);
33     }
34
35     std::ofstream outputFile(outputPath);
36     if (!outputFile.is_open()) {
37         std::cerr << "Failed to open output file: " << outputPath << std
::endl;
38         return;
39     }
40
41     outputFile << content;
42     outputFile.close();
43 }
44
45
```

## Листинг 2 — продолжение кода из Листинга 1

```
1      void findAndProcessDotFiles(const fs::path& inputDir, const fs::
path& outputDir) {
2          for (const auto& entry : fs::directory_iterator(inputDir)) {
3              if (entry.is_regular_file() && entry.path().extension() == ".
dot") {
4                  fs::path outputPath = outputDir / entry.path().filename();
5                  processDotFile(entry.path(), outputPath);
6              }
7          }
8      }
9
10     int main() {
11         std::string inputDirectory;
12         std::cout << "Enter the input directory: ";
13         std::cin >> inputDirectory;
14
15         fs::path currentPath = fs::current_path();
16         findAndProcessDotFiles(inputDirectory, currentPath);
17
18         std::cout << "Processing completed." << std::endl;
19         return 0;
20     }
21
22
23
24
```