Лабораторная работа №5

Вариант 1

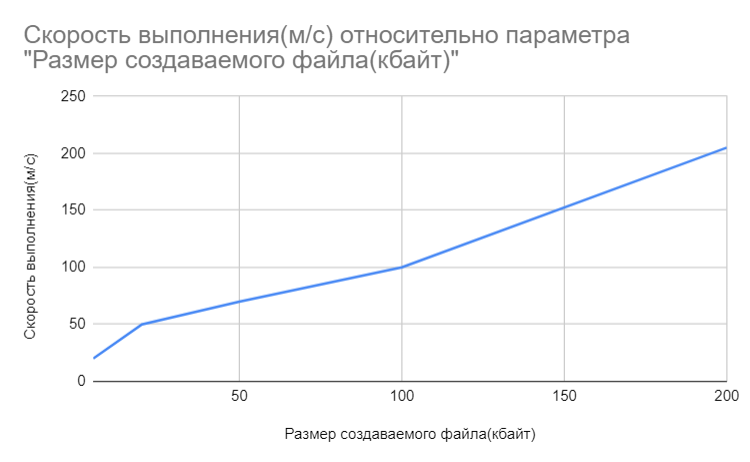
Выполнили ст. гр. 221703:

Воложинец А.А.

Оскирко Д.А.

*#include* <iostream>  
*#include* <vector>  
*#include* <fstream>  
*#include* <sstream>  
  
*class* File {  
*public*:  
 std::string name;  
 std::string content;  
  
 File(*const* std::string& n, *const* std::string& c = "") : name(n), content(c) {}  
};  
  
*class* FileSystem {  
*private*:  
 std::vector<File> files;  
  
*public*:  
 *void* createFile(*const* std::string& name, *const* std::string& content = "") {  
 files.emplace\_back(name, content);  
 saveToFile(name, content);  
 std::cout << "File '" << name << "' created.\n";  
 }  
  
 *void* deleteFile(*const* std::string& name) {  
 *auto* it = std::remove\_if(files.begin(), files.end(),  
 [name](*const* File& file) { *return* file.name == name; });  
 *if* (it != files.end()) {  
 files.erase(it, files.end());  
 std::cout << "File '" << name << "' deleted.\n";  
 } *else* {  
 std::cout << "File '" << name << "' not found.\n";  
 }  
 }  
  
 *void* copyFile(*const* std::string& source, *const* std::string& destination) {  
 *auto* it = std::find\_if(files.begin(), files.end(),  
 [source](*const* File& file) { *return* file.name == source; });  
  
 *if* (it != files.end()) {  
 files.push\_back(File(destination, it->content));  
 std::cout << "File '" << source << "' copied to '" << destination << "'.\n";  
 } *else* {  
 std::cout << "File '" << source << "' not found.\n";  
 }  
 }  
  
 *void* moveFile(*const* std::string& source, *const* std::string& destination) {  
 *auto* it = std::find\_if(files.begin(), files.end(),  
 [source](*const* File& file) { *return* file.name == source; });  
  
 *if* (it != files.end()) {  
 it->name = destination;  
 std::cout << "File '" << source << "' moved to '" << destination << "'.\n";  
 } *else* {  
 std::cout << "File '" << source << "' not found.\n";  
 }  
 }  
  
 *void* writeFile(*const* std::string& name, *const* std::string& content) {  
 *auto* it = std::find\_if(files.begin(), files.end(),  
 [name](*const* File& file) { *return* file.name == name; });  
  
 *if* (it != files.end()) {  
 it->content = content;  
 std::cout << "Content written to file '" << name << "'.\n";  
 } *else* {  
 std::cout << "File '" << name << "' not found.\n";  
 }  
 }  
  
 *void* readFile(*const* std::string& name) {  
 *auto* it = std::find\_if(files.begin(), files.end(),  
 [name](*const* File& file) { *return* file.name == name; });  
  
 *if* (it != files.end()) {  
 std::cout << "Content of file '" << name << "':\n";  
 std::cout << it->content << "\n";  
 } *else* {  
 std::cout << "File '" << name << "' not found.\n";  
 }  
 }  
  
 *void* dumpFileSystem() {  
 std::cout << "File System Dump:\n";  
 *for* (*const auto*& file : files) {  
 std::cout << "File: " << file.name << "\n";  
 std::cout << "Content: " << file.content << "\n";  
 std::cout << "-----------------\n";  
 }  
 }  
  
*private*:  
 *void* saveToFile(*const* std::string& name, *const* std::string& content) {  
 std::ofstream file(name);  
 *if* (file.is\_open()) {  
 file << content;  
 file.close();  
 } *else* {  
 std::cerr << "Unable to open file '" << name << "' for writing.\n";  
 }  
 }  
};  
  
*int* main() {  
 FileSystem fileSystem;  
  
 fileSystem.createFile("file1.txt", "Hello, World!");  
 fileSystem.copyFile("file1.txt", "file2.txt");  
 fileSystem.moveFile("file1.txt", "file3.txt");  
 fileSystem.createFile("file2.txt", "It's me!");  
 fileSystem.writeFile("file2.txt", "New content for file2");  
 fileSystem.readFile("file1.txt");  
 fileSystem.dumpFileSystem();  
  
 *return* 0;  
}

**Графики**:

****

