

```
1 #include <iostream>
2 #include <fstream>
3 #include <filesystem>
4 #include <string>
5 #include "myfile.hpp"
6
7 OutMyFile::OutMyFile(const int n, const std::filesystem::path &path) : N(n),
  file_path(path) {
8     return;
9 }
10
11 void OutMyFile::out(const double * const val) const {
12
13     std::ofstream outfile(file_path);
14     outfile << std::setprecision(std::numeric_limits<double>::max_digits10)
15         << std::showpoint;
16
17     for (int i = 0; i < N; ++i) {
18         outfile << val[i] << std::endl;
19     }
20
21     return;
22 }
23
24 void OutMyFile::out(const double * const xval, const double * const yval)
  const {
25
26     std::ofstream outfile(file_path);
27     outfile << std::setprecision(std::numeric_limits<double>::max_digits10)
28         << std::showpoint;
29
30     for (int i = 0; i < N; ++i) {
31         outfile << xval[i] << " " << yval[i] << std::endl;
32     }
33
34     return;
35 }
36
37 InMyFile::InMyFile(const std::filesystem::path &path) :file_path(path),
  N(getNum()) {
38
39     std::ifstream infile(file_path);
40
41     if (!infile) {
42         std::cout << "cannot open" << std::endl;
43         return;
44     }
45
46     arr = new double[N];
47     for (int i = 0; i < N; ++i) {
48         infile >> arr[i];
49     }
50     infile.close();
51
52     return;
53 }
54
55 InMyFile::~InMyFile() {
56     if (arr != nullptr) {
```

```
57     delete[] arr;
58 }
59 return;
60 }
61
62 inline bool InMyFile::check_index(const int idx) const {
63     if ((idx < 0) || (N <= idx)) return false;
64     return true;
65 }
66
67 int InMyFile::getNum() const{
68
69     std::ifstream infile(file_path);
70
71     int count = 0;
72     std::string line;
73
74     while (std::getline(infile, line)) ++count;
75
76     infile.close();
77
78     return count;
79 }
80
81 double InMyFile::operator() (const int idx) const {
82     if(!check_index(idx)) {
83         std::cout << "error: out of index" << std::endl;
84         return 0;
85     }
86     return arr[idx];
87 }
88
```