

```
#include <iostream>

#include <map>

#include <vector>

#include <fstream>

#include <string>


using namespace std;


const string FILEPATH = "Mitarbeiter.ini";

int Fehlercode = 0;

int menu = 0;


fstream datei;

string fileContent;


// Vector für die Eigenschaften
vector<string> eigenschaften;


// Mitarbeiter <String,String>
map<string, string> mitarbeiter;


// Mitarbeiter in Mitarbeiterliste
vector<map<string, string>> mitarbeiterliste;


vector<string> openFile(string Path) {
    // Datei Öffnen
    datei.open(Path, fstream::in);
    while (!datei.eof()) {
        // Inhalt der Datei in Vector Eigenschaften speichern
        while (getline(datei, fileContent, ',')) {
            eigenschaften.push_back(fileContent);
        }
    }
}
```

```

    }
}
// Datei Schließen
datei.close();
return eigenschaften;
}

```

```

void showMitarbeiter() {
    // Ausgabe der Mitarbeiterliste
    for (map<string, string> p : mitarbeiterliste) {
        for (string key : eigenschaften) {
            cout << p[key] << " | ";
        }
        cout << "\n";
    }
}

```

```

void addMitarbeiter() {
    // Eigenschaften löschen und neu Einlesen
    eigenschaften.clear();
    openFile(FILEPATH);

    // Map (Mitarbeiter) per Benutzereingabe befüllen
    for (int i = 0; i < eigenschaften.size(); i = i + 1) {
        string wert;
        cout << "Bitte " << eigenschaften[i] << " eingeben: ";
        cin >> wert;
        cin.sync();
        mitarbeiter.insert(pair<string, string> { eigenschaften[i], wert });
    }
}

```

```

// Mitarbeiter der Mitarbeiterliste hinzufügen

```

```

    mitarbeiterliste.push_back(mitarbeiter);

    // Mitarbeiter Map Clear für neuen Mitarbeiter
    mitarbeiter.clear();
}

// MAIN MENU
void showMenu() {
    cout << "Mitarbeiterverwaltung \n";

    cout << "1 | Mitarbeiter Liste" << endl;
    cout << "2 | Mitarbeiter Hinzufuegen" << endl;
    cin >> menu;
    cin.sync();

    cout << "\n";
    // Menu
    switch (menu) {
    case 1:
        showMitarbeiter();
        showMenu();
        break;
    case 2:
        addMitarbeiter();
        showMenu();
        break;
    default:
        showMenu();
    }
}

// MAIN PROGRAMM

```

```
int main() {  
  
    showMenu();  
  
    return Fehlercode;  
}
```